# Kalenderdarstellung in Computersystemen ohne Festhalten an der klassischen KalenderblattMetapher

Michael Treml, Mat.-Nr. 1025496

Bachelor-Arbeit aus Medieninformatik und Visual Computing (E $033\ 532)$ 

# Sommersemester 2013

Betreuer: Univ.Prof. DI Dr. Thomas Grechenig Technische Universität Wien Fakultät für Informatik Institut für rechnergestützte Automation Research Group for Industrial Software (INSO)

# Inhaltsverzeichnis

1	Kurzfassung	4
2	Motivation	5
3	Problemstellung	6
4	Ziele4.1 Zielgruppe	8 9 10 12
5	Methodik	14
6	Klassische Kalenderansichten  6.1 Referenzkalender: Lightning 6.2 Persistente Elemente in allen Ansichten 6.3 Tageskalender 6.4 Wochenkalender 6.5 Monatskalender 6.6 Mehrere Wochen 6.7 Termindetails	14 15 15 19 22 24 25 27
7	Alternative Ansätze 7.1 Semantic Zooming	29 31 32 36
8	Prinzip eines Tagbalkenkalenders	38
9	Low-Fidelity-Prototyp         9.1       Hauptfenster         9.1.1       Navigation         9.1.2       Termindarstellung         9.1.3       Anpassung der angezeigten Intervalle         9.1.4       Menüleiste         9.2       Terminfenster         9.3       Kalenderverwaltung         9.4       Einstellungen	40 41 43 45 46 47 50 51
10	Cognitive Walkthrough 10.1 Zuordnung von Wochen- und Monatstagen ablesen 10.2 Terminübersicht anzeigen	<b>54</b> 55 57 59
11	High-Fidelity-Prototyp	60

12	Usa	bility-	$\Gamma\!\mathrm{est}$	62
	12.1	Ziele .		62
	12.2	Metho	de	63
		12.2.1	Testpersonen	64
		12.2.2	Testszenarien	65
			Testmaterial und Einrichtung	68
			Design und Messungen	70
			Pilottest	72
	12.3		nisse	73
	12.0	_	Quantitative Analyse	73
			Beobachtungen in Lightning	
			Beobachtungen in TaBaKa	79
13	Ank	nüpfp	unkte für aufbauende Arbeiten	82
14	Fazi	t		83
15	Lite	ratur		85
16	Anh	_		86
	16.1	Beispie	elangabe für den Usability-Test	86
		16.1.1	Allgemeines	87
		16.1.2	Demographische Daten	87
		16.1.3	Vorkenntnisse	88
		16.1.4	Vorgehensweise	89
		16.1.5	Kalender 1: Lightning (Thunderbird-Plugin)	90
			Kalender 2: TaBaKa	94
			Vergleich der beiden Kalender	98
	16.2		ergebnisse des Usability-Tests	
			Erwartete Antworten	99
			Testperson 1 (Pilottest)	100
			Testperson 2	
			Testperson 3	
			Testperson 4	
			Testperson 5	
			Testperson 6	
			Testperson 7	
	16.3		ne im Usability-Test	
			ext des High-Fidelity-Prototypen	
	10.1	•	HTML	
		16.4.2		
			Initialisierungsdaten	
			Hilfsfunktionen zur Formatierung	
			Hauptfunktionen	
			Markierungen für Früh- und Spättermine	
			Navigationsleiste	
			Markierung des aktuellen Tages	
			Zienen von Eiementen	155 156
		1111411	Lakaberen der Ausichi.	1.00

# 1 Kurzfassung

Heutige Kalendersoftware bietet zumeist drei verschiedene Ansichten auf den selben Datenbestand: Tagesansicht, Wochenansicht und Monatsansicht. Dass diese Ansichten sich etabliert haben, ist naheliegend, wenn man bedenkt, dass diese Kalenderdarstellungen auch schon vor dem Computerzeitalter existiert hatten und daher schon aus älteren Medien vertraut waren.

Doch mit diesen althergebrachten Darstellungen ergeben sich auch Probleme. Insbesondere ist es in Bezug auf Usability nur selten eine gute Idee physische Objekte unreflektiert in den digitalen Raum zu übernehmen. Auf Papier bestand noch die physische Notwendigkeit Kalenderdaten auf Blätter aufzuteilen; Ein Computer dagegen hat grundsätzlich die Möglichkeit Zeit besser als das darzustellen, was sie tatsächlich ist – nämlich etwas, das kontinuierlich fortläuft.

In der nachfolgenden Arbeit wurde analysiert, was an klassischen Kalenderdarstellungen von Nachteil ist und welche Alternativen es gibt.

Aufbauend auf diesen Beobachtungen wurde ein Prototyp – genannt Ta-BaKa (steht für »Tagbalkenkalender«) – erstellt, der die besten Eigenschaften aus allen analysierten Darstellungen in sich vereinen soll. Dazu wurde im ersten Schritt ein Low-Fidelity-Prototyp erstellt, der den gesamten geplanten Funktionsumfang abdeckt. Anschließend wurde in HTML und JavaScript ein funktionell stärker eingeschränkter High-Fidelity-Prototyp umgesetzt, in dem die gesamte Navigationsfunktionalität des Kalenders implementiert wurde.

Dieser Prototyp wurde abschließend einem Usability-Test unterzogen, in dem er direkt mit einem klassischen Kalender verglichen wurde.

Die Testpersonen benötigten ein wenig, um sich an TaBaKa zu gewöhnen, konnten aber letztendlich die aufwändigste Aufgabe des Tests, in der es darum ging abweichende Termine innerhalb von drei Monaten ausfindig zu machen, deutlich effizienter lösen als in dem Vergleichskalender. Die Personen äußerten sich außerdem sehr positiv darüber, dass man in TaBaKa den Intervall der angezeigten Tage frei ändern kann.

Wie zu erwarten war, sind Benutzer aber im Allgemeinen trotzdem nur schwer für etwas Neues zu begeistern, wenn die Alternative so bekannt und etabliert ist, wie das bei Kalenderdarstellungen der Fall ist. Die direkten Vergleichsbeurteilungen der Testpersonen fielen daher in den meisten Punkten zugunsten der klassischen Darstellung aus, obwohl auch die Handhabung des Vergleichskalenders nur selten ohne Komplikationen verlaufen war.

Da zumindest manche Personen dem Konzept aber doch relativ offen gegenüberstanden, ist eine Weiterentwicklung von TaBaKa sicher sinnvoll. Die Ergebnisse des Usability-Tests liefern dazu gute Ansätze, wo Verbesserungen notwendig sind. Insbesondere konnte in den Tests nachgewiesen werden, dass Markierungen, die eigentlich auf Termine außerhalb des sichtbaren Bereichs hinweisen sollten, übersehen werden und daher deutlicher hervorgehoben werden müssen.

# 2 Motivation

Als Student, der nebenbei auch arbeitet, ist gute Zeitplanung für mich sehr wichtig und bestehende Kalender geben mir nicht den Überblick, den ich gerne hätte. In der Regel benötige ich detaillierte Informationen über den aktuellen Tag, aber auch Grundinformationen über die nächsten paar Tage. In klassischen Kalendern sind diese Informationen aber ungünstig über mehrere verschiedene Ansichten verteilt und in vielen Fällen – zum Beispiel am Ende einer Woche in der Wochenansicht – erhalte ich mehr Informationen über vergangene Tage, die mich in der Regel nicht mehr tangieren, als über bevorstehende Ereignisse.

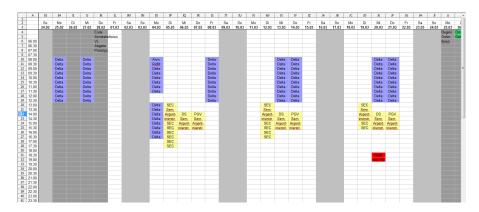


Abbildung 1: Händisch geführter Ersatzkalender

Aus diesem Grund hatte ich mir schon im ersten Semester einen rudimentären Ersatzkalender (siehe Abbildung 1) angelegt, indem ich ein Tabellenkalkulationsblatt so formatiert habe wie ich mir einen übersichtlicheren Kalender vorgestellt hatte. Das Grundprinzip dieses einfachen Prototypen hat sich für mich persönlich bewährt und ich führe diesen Kalender auch jetzt im sechsten Semester noch weiter, allerdings muss ich dabei auf viel Komfort verzichten, den mir ein richtiger Software-Kalender bieten würde. Für jeden Termin muss ich selbst die Formatierung der entsprechenden Zellen festlegen, Serientermine kann ich nur durch händisches Kopieren eintragen und für detaillierte Informationen ist meistens nicht ausreichend Platz.

Neben meinem eigenen Planungsbedarf beobachte ich auch immer wieder bei meiner Arbeit in einem Büro abseits der Informatik- und Technik-Branche, dass bestimmte Eigenschaften von elektronischen Kalendern auch bei anderen Leuten zu Komplikationen und Problemen führen.

Zwischen den drei klassischen Ansichten – Tag, Woche und Monat – wird in der Regel nur sehr selten gewechselt. Stattdessen neigen die meisten Mitarbeiter dazu nur in einer einzigen Ansicht zu arbeiten und stoßen dabei an die Grenzen ebendieser Darstellung. So wurde ich zum Beispiel schon von jemandem gebeten bei Einträgen in seinen Kalender immer die Startzeit auch in den Betreff zu schreiben, weil diese in seiner gewohnten Darstellungsform, der Wochenansicht, sonst nur schwer ablesbar ist, was darauf zurückzuführen ist, dass die Uhrzeiten hier ganz links angeschrieben sind, der aktuelle Tag aber außer montags immer weiter rechts liegt. Bei einer Umstellung auf die Monatsansicht würde

die Uhrzeit des Termins zwar automatisch vor dem Titel angezeigt werden, allerdings ist dieser Unterschied nicht intuitiv verständlich und diese Ansicht hat wieder andere Nachteile – zum Beispiel können an einem dicht verplanten Tag aus Platzgründen nicht alle Termine angezeigt werden, weshalb die betroffene Person diese Ansicht wiederum ablehnt.

Irritierend kann durch diese Praxis weiter sein, dass im Büro mehrere Leute über einen Server auf die selben Kalender zugreifen, sich diese aber jeweils in ihrer eigenen Lieblingsansicht zeigen lassen. So benutze ich in der Regel die Monatsansicht, weil ich im Büro aufgrund meiner geringfügigen Beschäftigung nur selten spezielle Termine habe und eher einen groben zeitlichen Überblick benötige. Das angesprochene separate Anschreiben von Uhrzeiten im Betreff führt in der Monatsansicht allerdings dazu, dass die Startzeit dann doppelt dort steht und der ohnehin knappe Platz für den Termintitel noch geringer wird.

Neben der allgemeinen Darstellung als Hauptkritikpunkt gibt es auch weitere Dinge, die vieles erleichtern würden, aber nach wie vor nicht zur Standardausstattung von Kalendern gehören. So gibt es zwar in den meisten Programmen eine vorgefertigte Kategorie für Geburtstage, aber nur in wenigen Fällen auch die durchaus naheliegende Möglichkeit sich bei jedem Termin gleich das Alter anzeigen zu lassen – obwohl das nur vergleichsweise geringen Mehraufwand bei der Implementierung bedeuten würde.

Ein weiteres Beispiel wäre die direkte Unterstützung der Eingabe und Anzeige von Anreisezeiten. Im Büro kommt es gelegentlich vor, dass Anreisezeiten übersehen werden, weil diese nicht im Kalender blockiert sind. In den meisten Kalendern müsste man dazu entweder einen zusätzlichen Hilfstermin anlegen oder den ursprünglichen Termin künstlich verlängern, wodurch die tatsächliche Startzeit schwieriger abzulesen wird.

Meine eigenen Bedürfnisse in der Lehrveranstaltungs- und Arbeitszeitplanung sowie meine Beobachtungen im Büro haben mich so letztendlich dazu motiviert nach einer Alternative zu klassischer Kalendersoftware zu suchen.

# 3 Problemstellung

Die in der Motivation genannten Probleme lassen sich großteils darauf zurückführen, dass sich das Interface der am weitesten verbreiteten Kalenderprogramme nach wie vor an der Metapher des Kalenderblattes orientiert.

Im Vorgänger von Kalendersoftware, dem klassischen Papierkalender, ist die Aufteilung auf einzelne Blätter aufgrund physischer Beschränkungen mehr oder weniger unumgänglich. Je nachdem, wie viele Tage auf ein einzelnes Blatt gedruckt werden und wie groß dieses Blatt ist, bleibt unterschiedlich viel Platz für Zusatzinformationen. So gibt es Jahreskalendarien, an denen man lediglich die Wochentage bestimmter Tage im Jahr ablesen kann, Terminplaner, in denen man detaillierte Daten zu jedem Tag niederschreiben kann und mehrere Abstufungen dazwischen, in denen jeweils ein anderes Verhältnis zwischen weitläufiger Übersicht und Detailinformationen herrscht. Da bei einer Aufteilung auf mehrere Seiten irgendwo eine Trennung erfolgen muss, ist es naheliegend Seiten bei Terminplanern in logisch abgeschlossene Zeitintervalle zu teilen. Daraus resultiert wahlweise ein neues Blatt für jeden Monat, jede Woche oder jeden Tag.

Da man am effizientesten nur einen einzigen Papierkalender für die persönliche Terminplanung nutzt, um keinen Mehrfachaufwand zu haben, entscheidet

man sich hier natürlich für jene Variante, die den eigenen Bedürfnissen – insbesondere der Anzahl der Termine und deren notwendigen Details – am meisten entgegenkommt. Eine Studie von 1985 zeigte zwar auf, dass im Büroumfeld durchschnittlich zwei Kalender genutzt wurden, für gewöhnlich diente einer davon aber nur als Ergänzung für einen gröberen Überblick. [6]

Kalendersoftware greift dieses Prinzip auf und stellt den Benutzern die gleichen Ansichten zur Verfügung, die sie bereits in ihrer Papierform kennen. Neu ist demgegenüber, dass man hier grundsätzlich zwischen den drei Ansichten Monat, Woche und Tag wechseln kann, um je nach Bedarf unterschiedlich viele Informationen zu erhalten.

Das Problem an dieser Kalenderblattmetapher ist aber, dass dadurch bestimmte Potentiale der elektronischen Darstellung ungenutzt bleiben. Die klassischen Darstellungsformen haben auf Papier ihre Berechtigung, weil sie sich nach den speziellen Anforderungen des Mediums richten, Computer haben aber andere Anforderungen. Eine Unterteilung in Seiten ist hier nicht länger zwingend erforderlich und weil ein Computer mit dem aktuellen Tag das wichtigste Element des Kontextes kennt, kann er die Darstellung des Kalenders kontextabhängig anpassen, sodass immer die wichtigsten Informationen im Zentrum der Aufmerksamkeit stehen. Das kann ein großer Vorteil gegenüber der zwangsweise statischen Darstellung auf Papier sein.

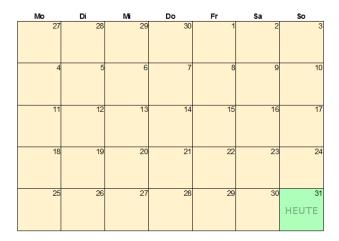


Abbildung 2: Im Extremfall liegen am Monatsletzten in der Monatsansicht alle Felder außer einem in der Vergangenheit, welche für die alltägliche Kalendernutzung nur selten von Bedeutung ist. Die relevante Information beschränkt sich hier auf rund drei Prozent der verfügbaren Fläche.

Dass solche Kalendersoftware sich zu sehr an ihrem Papiervorbild orientiert, wird nicht nur von mir kritisiert. Marco Arment, ehemaliger Lead-Developer bei der erfolgreichen Internet-Plattform Tumblr, zeigt zum Beispiel in seinem Blog anschaulich anhand einer eingefärbten Monatsübersicht, dass am Ende des Monats mit Abstand der meiste Platz für Informationen vergangener Tage beansprucht wird – also für Informationen, die man mit hoher Wahrscheinlichkeit gar nicht mehr benötigt. [8] Und auch wenn man dieses Verhältnis in Zahlen ausdrückt, ist das Ergebnis ernüchternd. Eine klassische Monatsansicht zeigt

fünf Wochen zu zu je sieben Tagen an – das ergibt in Summe 35 gleich große Felder. Geht man davon aus, dass vergangene Tage im alltäglichen Gebrauch uninteressant sind, dann ist am Monatsletzten im Extremfall nur ein einziges dieser Felder von Bedeutung – das entspricht einem Anteil von weniger als drei Prozent der verfügbaren Fläche und ist damit in hohem Maße ineffizient.

Natürlich ist dieses Verhältnis in anderen klassischen Ansichten besser. Im Optimalfall nutzt man seinen Kalender in der Tagesansicht, womit der gesamte angezeigte Bereich relevant ist, hat dann aber andererseits wieder das Problem, dass der Kontext verloren geht, denn auch wenn der aktuellste Tag der wichtigste ist, benötigt man für Planungsaufgaben zumindest auch einen Überblick der nachfolgenden Tage. Ein Problem ist hier, dass man nicht eine einheitliche Ansicht nach eigenen Bedürfnissen frei skalieren kann, sondern drei fixe Darstellungsformen vorgesetzt bekommt, die ganz unterschiedlichen Paradigmen folgen. Vor allem die tabellarische Anordnung in in der Monatsübersicht hat eine ganz andere Struktur als die nebeneinander angereihten Tage in einer Wochenübersicht. Insofern ist es nicht verwunderlich, wenn sich Benutzer auf eine einzelne Ansicht spezialisieren, auch wenn sie dann mit deren Einschränkungen leben müssen.

Insgesamt ließen sich also einige Verbesserungen für Kalenderdarstellungen finden, wenn man sich nicht zu sehr auf die althergebrachten Lösungen versteift.

# 4 Ziele

Basierend auf einer Analyse bestehender Kalenderdarstellungen und deren jeweiliger Vor- und Nachteile soll aus dieser Arbeit ein Programmartefakt hervorgehen, das ein alternatives Konzept für die Kalenderdarstellung ohne die übliche Aufteilung in Tag-, Wochen- und Monatsansicht umsetzt und dazu geeignet ist später zu einem voll funktionsfähigen Kalender weiterentwickelt zu werden.

Ziel dieser Neuentwicklung soll primär eine verbesserte Handhabung alltäglicher Anwendungsfälle sein, indem die klassische Kalenderblattmetapher aufgebrochen wird. Komplexere Aufgaben wie Mehrbenutzerverwaltung oder Suchanfragen sollen nur in frühen Entwicklungsphasen im Konzept aufgegriffen werden, um den allgemeinen Kontext einer voll einsatzfähigen Kalenderanwendung zu diskutieren und um in wesentliche Design-Entscheidungen einzufließen, werden später aber nicht mehr im Detail verfolgt, da sie den Rahmen dieser Arbeit sprengen würden.

Während der Entwicklung des Artefakts soll eruiert werden, für welche alltäglichen Standard-Aufgaben – ein Beispiel wäre das Finden freier Terminzeiten – es tatsächlich besser geeignet ist als klassische Kalenderdarstellungen und wie dieser alternative Kalender von realen Benutzern wahrgenommen wird.

Basierend auf den Ergebnissen bestehender Arbeiten zu neuen Kalenderdarstellungen ist abzusehen, dass es trotz nachweisbarer Verbesserungen zu Akzeptanzproblemen kommen kann, weil die Handhabung klassischer Kalender gut eintrainiert ist, wogegen ein neuer Ansatz erst einen Lernprozess beim Benutzer erfordert. So wurde etwa bei der Entwicklung eines Kalender-Interfaces namens FishCal [3] festgestellt, dass vorgegebene Tasks damit eher vollständig und in manchen Fällen auch schneller abgeschlossen werden als mit einem herkömmlichen Kalender, während die Testbenutzer auch einige klare Vorteile, vor allem einen besseren Wechsel zwischen Ansichten, einräumten – trotzdem gaben die

Benutzer an für den täglichen Gebrauch einen herkömmlichen Kalender vorzuziehen und gaben diesem auch bessere Bewertungen in allgemeinen Kategorien wie »Attractiveness« und »Behavior«.

Anmerken sollte man zu diesem Beispiel aber, dass die Ziele von FishCal etwas anders definiert waren als die Ziele dieser Arbeit. Insbesondere lag der Schwerpunkt stärker auf komplexen Tasks und langfristiger Planung, zum Beispiel Suchabfragen über mehrere Monate. Das Artefakt, das aus der vorliegenden Arbeit hervorgehen soll, hat dagegen das primäre Ziel alltägliche Standardaufgaben, zum Beispiel das Finden einer freien Terminzeit in den kommenden sieben Tagen, zu erleichtern. Dieser Unterschied könnte dazu führen, dass Benutzer eher auch die Vorzüge für den täglichen Gebrauch erkennen.

# 4.1 Zielgruppe

Das neu entwickelte Artefakt soll sich an Personen richten, die mit elektronischen Kalendern konfrontiert sind oder diese grundsätzlich effektiv nutzen könnten. Neben Büroangestellten oder Selbständigen kann diese Definition auch auf Schüler und Studenten aufgrund von Stundenplänen und Prüfungsterminen oder auf Personen mit vielen privaten Terminen zutreffen.

Der Schwerpunkt liegt dabei vorwiegend auf Personen, die laufend Details zum aktuellen Tag und eine Übersicht der nächsten Tage bis Wochen benötigen, da sie eine entsprechende Anzahl relevanter Termine haben. Wer dagegen einen Kalender nur benötigt, um sich, um ein Beipsiel zu nennen, durchschnittlich ein Mal im Monat durch eine automatische Meldung an einen Geburtstag erinnern zu lassen, hat keine wesentlichen Vorteile von einer veränderten Darstellung, weil der Kalender in diesem Fall nur als Service im Hintergrund agiert.

Die Benutzer müssen grundsätzlich offen für Neues und bereit zum Umdenken sein, weil das entwickelte Programm in Konkurrenz zu Kalender-Darstellungsformen steht, die durch ihre Papier-Vorgänger seit Generationen als selbstverständlich gelten. Aus diesem Grund sollen bevorzugt – aber nicht ausschließlich – jüngere Menschen angesprochen werden, weil diese sich vermutlich noch leichter umgewöhnen können als Personen, die schon seit Jahrzehnten mit herkömmlichen Darstellungen arbeiten.

Da das Konzept klassischer Kalender aufgebrochen werden soll und diese sich ohnehin an ihren Papiervorgängern orientieren, ist es nicht erforderlich, dass Benutzer schon Erfahrung in der Bedienung von Kalenderprogrammen haben. Es wird davon ausgegangen, dass zumindest in Europa und vorerst vor allem den deutschsprachigen Raum als primärem Zielgebiet jeder Volljährige mit dem grundsätzlichen Konzept eines Kalenders vertraut ist.

Eine allgemeine Computerkompetenz wird dagegen vorausgesetzt. Basierend auf eigenen Erfahrungen in Studium und Büro soll das Artefakt vorerst vor allem für den Einsatz auf Desktop-Rechnern und Laptops konzipiert werden, wodurch es vorteilhaft ist, wenn auch die Benutzer durch Beruf oder Ausbildung mit diesem Medium vertraut sind. Das grundlegende Konzept, das im Rahmen dieser Arbeit zustande kommt, soll grundsätzlich auch auf kleinere tragbare Geräte wie Smartphones anwendbar sein, diese Portierung wird aber vorerst noch nicht im Detail beachtet, um das Grundkonzept erst einmal mit angemessenem Aufwand auf einem Einzelmedium zu testen.

# 4.2 Anwendungsfälle

Um bestehende Kalender analysieren und selbst einen Prototypen entwickeln zu können, ist es im ersten Schritt notwendig festzulegen, was ein Kalender leisten soll. Aus diesem Grund werden nachfolgend Anwendungsfälle und zugehörige Anforderungen aufgelistet, die von bestehenden Kalendern abgedeckt werden oder für eine Neuentwicklung erstrebenswert wären. Um in nachfolgenden Entwicklungsphasen Prioritäten setzen zu können, werden die einzelnen Anwendungsfälle absteigend nach ihrer Wichtigkeit für die Ziele dieser Arbeit sortiert. Weit hinten gelistete Punkte, die nur geringen Einfluss auf die allgemeine Kalenderdarstellung haben, werden im später implementierten Artefakt vorerst ausgelassen.

#### 1. Zuordnung von Wochen- und Monatstagen ablesen

Die Basisfunktion jedes Kalenders besteht darin ablesen zu können, welcher Wochentag einem bestimmten Tag im Monat zugewiesen ist – und umgekehrt. Bei manchen Kalendertypen wie Jahreskalendarien in Scheckkartenform ist das mangels Platz für eigene Notizen die einzige unterstützte Funktion. Neben Wochen- und Monatstagen ist oft auch die Kalenderwoche von Bedeutung.

# 2. Terminübersicht anzeigen

Kalender sollen einen Überblick geben, welche Termine man an bestimmten Tagen hat. In dieser Übersicht sollten zumindest die essentiellen Fragen nach dem Was und dem Wann beantwortet werden, also eine Terminzeit und ein Titel angegeben sein. Wenn in der Übersicht schon mehrere Daten angezeigt werden, erspart es sich der Benutzer die Details gesondert aufzurufen – bei zu vielen Daten kann allerdings auch die Übersicht verloren gehen.

Um verschiedene Terminarten – bei Studenten beispielsweise Vorlesungen, Übungen und Prüfungen – schneller unterscheiden zu können, ist eine optische Differenzierung zwischen unterschiedlichen Kategorien sinnvoll.

Da ein Kalender sehr viele Informationen enthält, ist es für die Planung wichtig, dass die essentiellen Daten, vor allem der aktuelle Tag, besondere Beachtung in der Darstellung finden, damit sie in der großen Menge vorhandener Informationen nicht untergehen.

#### 3. Termindetails anzeigen

Wenn in der Übersicht der Termine aus Platz- oder Übersichtsgründen noch nicht alle Details zu einem Termin angezeigt werden, muss es eine gesonderte Möglichkeit geben diese abzurufen.

#### 4. Termin eintragen

Es muss zumindest eine Möglichkeit bestehen neue Termine zu dem Kalender hinzuzufügen. Da es sich hierbei um eine häufig genutzte Operation handelt, bei der in der Praxis oft nur eine Bezeichnung und eine Zeit angegeben werden, sollen auch nur diese beiden Daten verpflichtend angegeben werden müssen.

#### 5. Termin bearbeiten

Bestehende Termine müssen nachträglich verändert werden können. Im Idealfall stehen dazu auch Drag-and-Drop-Funktionalitäten zur Verfügung, um Zeiten zu verändern oder Termine zu kopieren.

#### 6. Termin löschen

Bestehende Termine müssen wieder aus dem Kalender entfernt werden können. Im Idealfall lässt sich das auch über die Entfernen-Taste der Tastatur bewerkstelligen, um ein bekanntes Verhaltensmuster anwenden zu können.

# 7. Kategorien verwalten

Je nach eigenen Umständen hat man Bedarf für unterschiedliche Terminkategorien. Studenten benötigen Vorlesungen und Prüfungen, Büroangestellte dagegen Meetings und Kundentermine. Da eine fixe Vorgabe an Kategorien nie alle möglichen Bedürfnisse zufriedenstellend abdecken kann, müssen die Kategorien anpassbar sein.

#### 8. Synchronisieren von Kalendern

Wenn man einen Kalender von mehreren Geräten aus benutzt, ist eine Synchronisation erforderlich, um immer die aktuellsten Daten zur Verfügung zu haben. Im Optimalfall läuft die regelmäßige Synchronisation im Hintergrund ab, sodass der Benutzer sich nicht weiter darum kümmern muss, aber zumindest für die ursprüngliche Einrichtung müssen entsprechende sichtbare Komponenten vorhanden sein.

# 9. Importieren von Terminen

Das Importieren von Terminen ermöglicht einen leichteren Wechsel von anderen Kalendersystemen und die Aufnahme großer Terminmengen aus externen Quellen.

Besonders wichtig ist bei dem ursprünglichen Einrichten des Kalenders der Import regionaler Feiertage, da diese für fast jeden Kalenderbenutzer von Bedeutung sind. Im Optimalfall sollte dafür eine eigene Funktion zur Verfügung stehen, sodass man diese Termine nicht händisch aus einer externen Datei importieren muss.

# 10. Benutzerrechte vergeben

Vor allem im Büro-Umfeld ist es sehr wichtig auch Zugriff auf die Kalender der Kollegen zu haben, um besser miteinander kooperieren zu können. Aus diesem Grund soll es möglich sein nach Bedarf Lese- und Schreibrechte an andere Personen zu vergeben.

#### 11. Suchabfrage ausführen

Um in einem gut gefüllten Kalender spezifische Termine wiederzufinden, kann die Integration einer Suchfunktion hilfreich sein.

#### 12. Exportieren von Terminen

Äquivalent zum Import sollte auch eine Exportfunktion für Termine vorhanden sein, um diese in anderen Systemen nutzen und mit anderen Personen teilen zu können.

#### 4.3 Termindetails

Im letzten Kapitel wurden mehrfach Termindetails angesprochen ohne diese genauer zu definieren. Da es sich hierbei um Daten handelt, die nicht zwangsweise Anwendungsfälle repräsentieren, werden mögliche Detailinformationen, die zu einzelnen Terminen angegeben werden könnten, nachfolgend in einer separaten Liste angeführt. Auch diese Liste ist absteigend nach der Priorität für diese Arbeit sortiert.

#### 1. Zeitrahmen (Wann?)

Zu jedem Termin muss ein Start und ein Ende definiert sein. Die Zeitpunkte können entweder konkrete Uhrzeiten wie im Fall eines Zahnarzttermines oder allgemein Tage wie im Fall eines Geburtstages sein.

#### 2. Titel (Was?)

Der Titel soll in wenigen Worten angeben, worum es sich bei dem entsprechenden Termin handelt. Das ist die zentrale Information, die in der Terminübersicht angezeigt werden muss, um wirklich einen Überblick zu vermitteln.

#### 3. Ort (Wo?)

Der Ort gibt an, wo ein Termin stattfindet. Wenn sich ausreichend Platz findet, ist es sinnvoll diesen schon in der Terminübersicht aufzurufen, sodass man für diese essentielle Information nicht extra die Termindetails aufrufen muss und auch auf einen Blick erkennt, wann man von einem Ort zu einem anderen muss..

#### 4. Kategorie

Eine zugewiesene Kategorie gibt an, dass ein Termin von einer bestimmten Art ist. Das könnten etwa Prüfungen, Arzttermine oder Meetings sein. Die zugewiesene Kategorie sollte die Darstellung eines Termins in der Terminübersicht beeinflussen, sodass man bei einer großen Anzahl unterschiedlicher Termine schneller einen Überblick bekommt, um welche Art von Terminen es sich handelt.

#### 5. Beschreibung

Ein Beschreibungstext kann zusätzliche Informationen zu einem Termin enthalten, die so speziell sind, dass sie durch vorgegebene Termineigenschaften nicht abgedeckt werden. Das können zum Beispiel Informationen zu den Themen in einem Meeting sein.

#### 6. Anfahrt

Oft muss zwischen Terminen auch eine Reisezeit eingeplant werden. Es wäre günstig diese auch gesondert als Anreise zu vermerken, weil dieser Termin dann besser auf Veränderungen eingehen könnte. Hat man zum Beispiel zwei konkrete Termine mit angegebenen Orten und eine Anreisezeit von einer halben Stunde zum zweiten Termin, so könnte der Kalender bei einem Löschen des ersten Termins auch automatisch nachfragen, ob sich dadurch die Anreisezeit zum zweiten Termin verändert.

Die Grundidee Anreisezeiten auch als solche auszuzeichnen, habe ich von dem Romanautoren Ben Brooks [9], der aus eigener Erfahrung im Umgang mit Kalenderprogrammen Ideen zusammengetragen hat, übernommen. Er würde es auch begrüßen, wenn Kalender eingegebene Orte erkennen würden, um so selbständig zu eruieren, ob die notwendigen Fahrzeiten plausibel sind, was sicher auch eine interessante Erweiterung darstellt, für diese Arbeit aber aufgrund des hohen Aufwandes nicht weiter beachtet wird.

# 7. Beschäftigt/Verfügbar

Oft werden Termine in einen Kalender eingetragen, die nur zur Information dienen, aber keine persönliche Interaktion erfordern. Als Student ist es beispielsweise gut zu wissen, wann Vorlesungen stattfinden, auch wenn man diese aufgrund anderer Verpflichtungen nicht besuchen kann. Durch die Unterscheidung, ob man beschäftigt oder verfügbar ist, soll besser ersichtlich sein, welche Termine man tatsächlich oder bevorzugt wahrnehmen muss.

Nach meinen eigenen Beobachtungen werden solche Einstellungen kaum verwendet, weil die Benutzer sich nicht bewusst sind, dass diese überhaupt existieren. Aus diesem Grund wäre es sinnvoll, wenn der Kalender zumindest bei Terminkollisionen selbstständig nachfragt, welcher der kollidierenden Termine wahrgenommen werden soll, um anschließend die anderen Termine auf »Verfügbar« zu setzen, sodass zumindest an diesen Stellen deutlich sichtbar bleibt, wo die eigene Anwesenheit tatsächlich erforderlich ist.

# 8. Wiederholung

Um regelmäßige Termine nicht händisch eintragen zu müssen, sollte es die Möglichkeit geben Serientermine festzulegen. Neben einfachen Intervallen wie jährlicher, monatlicher, wöchentlicher und täglicher Wiederholung sollten auch komplexere Formen – beispielsweise »der letzte Freitag im Monat« definiert werden können, da sonst beispielsweise ein Arbeitsmeeting auf einen Sonntag fallen könnte.

Da bei vielen Serienterminen, insbesondere bei Jubiläen wie Geburts- oder Hochzeitstagen, auch der aktuelle Zähler der Iteration von Bedeutung ist, sollte es möglich sein sich diese Zahl anzeigen zu lassen.

#### 9. Erinnerungsmeldung

In vielen Fällen kann es sinnvoll sein, sich durch eine Meldung automatisch an einen Termin erinnern zu lassen, um sich zum Beispiel auf ein Meeting vorzubereiten oder um ein Geschenk für einen Geburtstag zu besorgen.

#### 10. Privatsphäre

In einem Kalender, der für andere Benutzer freigegeben ist, sollte es die Möglichkeit geben Termine oder deren Details vor anderen Benutzern zu verbergen, um die eigene Privatsphäre zu schützen.

# 5 Methodik

Im vorangegangenen Kapitel wurde festgelegt, welche Ziele diese Arbeit und das daraus hervorgehende Artefakt verfolgen sollen. Um diese zu erreichen, werden im weiteren Verlauf folgende Arbeitsschritte durchgeführt:

- 1. Bestehende Kalenderdarstellungen werden vorgestellt und so weit wie möglich bezüglich ihrer jeweiligen Tauglichkeit für die im Vorkapitel definierten Anwendungsfälle und Anforderungen analysiert. Dabei werden im ersten Schritt die klassischen Ansichten (Tag, Woche, Monat) anhand einer Referenzimplementierung und im zweiten Schritt bereits existierende Alternativkonzepte betrachtet.
- Aus den Analysen der bestehenden Darstellungen werden jene Eigenschaften aufgegriffen, welche für die Anwendungsfälle am effektivsten sind. Diese Eigenschaften werden zu einem vorerst theoretischen Konzept für eine neue Kalenderdarstellung zusammengefügt.
- Basierend auf dem Konzept wird ein Low-Fidelity-Prototyp erstellt, der aus künstlich geschaffenen Screenshots besteht und alle ursprünglich angeführten Anwendungsfälle abdeckt.
- 4. Durch einen Cognitive-Walkthrough werden die einzelnen Arbeitsschritte der wichtigsten Anwendungsfälle durchgegangen und der Durchführungsaufwand der klassischen Referenzimplementierung (siehe Schritt 2) gegenübergestellt, um tatsächliche Verbesserungen aufzuzeigen und gegebenenfalls Nachjustierungen durchzuführen.
- 5. Basierend auf dem Low-Fidelity-Prototypen wird ein High-Fidelity-Prototyp in HTML und JavaScript erstellt, der vorerst nur die wichtigsten Funktionen (Termindarstellung und Navigation) abdeckt, sodass das grundlegende Konzept anschaulich dargestellt wird. Gleichzeitig soll dieser aber schon so gestaltet werden, dass er später zu einem voll funktionsfähigen Kalender erweitert werden kann.
- 6. Der High-Fidelity-Prototyp und der Referenzkalender werden einem kleinen Usability-Test mit sechs Teilnehmern (zuzüglich eines Pilottests) unterzogen, um Praxistauglichkeit und Akzeptanz an realen Benutzern zu testen.

Die Einschränkung der betrachteten Anwendungsfälle in späteren Schritten dient vor allem dazu, den Aufwand dieser Arbeit in einem angemessenen Rahmen zu halten.

# 6 Klassische Kalenderansichten

In diesem Kapitel werden die klassischen Kalenderansichten analysiert, welche die Metapher des Kalenderblattes in gewohnter Art aufgreifen. Ziel ist es aufzuzeigen, worin die Stärken und Schwächen der einzelnen Ansichten liegen.

# 6.1 Referenzkalender: Lightning

Obwohl die meisten Kalender auf dem selben Paradigma aufbauen, gibt es in den zahlreichen Implementierungen doch diverse Unterschiede in Details, sodass man über diese nicht generalisiert diskutieren kann. Aus diesem Grund wird für die nachfolgenden Analysen auf eine konkrete Implementierung Bezug genommen.

Die Wahl fällt dabei auf Lightning [10] in Version 1.9.1 [11]. Lightning ist eine Erweiterung für den E-Mail-Client Thunderbird und ebenso wie dieser ein Projekt der Mozilla Foundation. Die Wahl fällt aus drei Gründen auf diesen Kalender. Erstens ist er frei verfügbar. Zweitens stammt er von einem etablierten Herausgeber – von der Mozilla Foundation wird immerhin auch der erfolgreiche Browser Firefox herausgegeben. Drittens ist bei einer solchen Erweiterung die Wahrscheinlichkeit gegenüber marktführenden Kalendersystemen wie Microsoft Outlook geringer, dass ein potentieller Benutzer diesen speziellen Kalender schon aus eigener Erfahrung kennt.

Der letzte Punkt als Vorteil mag vorerst merkwürdig klingen, lässt sich aber durch den weiteren Verlauf der vorliegenden Arbeit erklären. Im letzten Schritt soll ein Usability-Test durchgeführt werden, in dem der Referenzkalender zum Vergleich gegen das neu entwickelte Artefakt getestet wird. Wenn der benutzte Referenzkalender dem Benutzer hierbei noch unbekannt ist, sollten bereits erlernte Gewohnheiten das Vergleichsergebnis nicht ganz so stark verzerren wie das bei einem bekannten Kalendersystem der Fall wäre.

Verlässliche Zahlen zum tatsächlichen Marktanteil von Lightning sind kaum zu finden, es gibt aber Fakten, die eine relativ geringe Bedeutung nahelegen. Zum einen muss Lightning separat für Thunderbird nachgerüstet werden und ist nicht wie etwa der Kalender in Microsoft Outlook von vornherein vorhanden. Zum anderen ist letztes Jahr die funktionale Weiterentwicklung von Thunderbird offiziell beendet worden – unter anderem, weil man eine verstärkte Tendenz zur Nutzung von Web-Clients beobachtet [12].

Zur Vollständigkeit sei erwähnt, dass Lightning dem E-Mail-Client nicht nur einen Kalender, sondern auch auch eine Aufgabeverwaltung hinzufügt. Diese ist für die vorliegende Arbeit aber nicht von Bedeutung.

#### 6.2 Persistente Elemente in allen Ansichten

Das Kalenderfenster von Lightning ist grundsätzlich in drei Bereiche unterteilt (siehe Abbildung 4). Den größten Teil nimmt dabei die jeweils ausgewählte Detailansicht (1a) ein. Diese Ansichten werden in den nachfolgenden Kapiteln näher besprochen, zunächst folgt aber eine allgemeine Analyse jener Elemente, die immer erhalten bleiben oder in jeder Darstellungsform identisch aussehen.

Ein Wechsel zwischen den Ansichten erfolgt durch Reiter gleich über dem Hauptbereich am rechten Rand (1b). Links davon wird angegeben, welcher Zeitraum aktuell angezeigt wird. Diese Ausgabe enthält einerseits das konkrete Anfangs- und End-Datum inklusive Wochentag und andererseits den Bereich der angezeigten Kalenderwochen. Je nach gewählter Ansicht werden diese Angaben sinnvoll abgekürzt, sodass zum Beispiel in der Tagesansicht kein Von-Bis-Bereich, sondern nur ein einziges Datum ausgegeben wird.

Diese Informationen stellen schon den ersten Bestandteil dar, der kritikwürdig ist. Wie in den Anwendungsfällen beschrieben, ist es Basisaufgabe

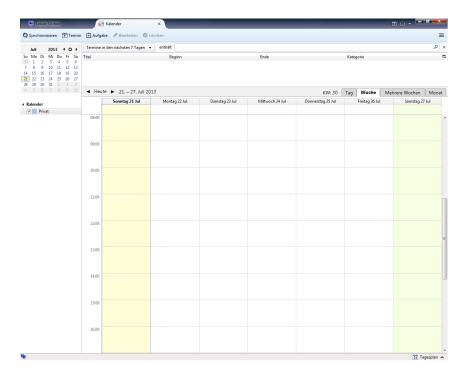


Abbildung 3: Lightning in Mozilla Thunderbird

jeder Kalenderdarstellung, Zuordnungen – zum Beispiel den heutigen Wochentag – daraus ablesen zu können. Das sollte also in der zuvor genannten Detailansicht zu jedem beliebigen Tag problemlos möglich sein. Wenn solche Basisdaten abseits dieser Darstellung noch einmal gesondert angeführt werden müssen, deutet das darauf hin, dass die Detailansicht ihren Basiszweck nicht gut erfüllt.

Im vorliegenden Fall ist die Angabe der angezeigten Tage eigentlich redundant, da man diese in allen Ansichten durchaus auch aus der eigentlichen Darstellung ablesen kann. Die zusätzliche Anzeige könnte bestenfalls noch als Feld interpretiert werden, dass die sonst so unterschiedlichen Ansichten in einer einheitlichen Form zusammenfassen soll. Da sich die Formatierung dieses Strings aber ebenfalls mit jeder Ansicht verändert, ist auch diese Sichtweise fraghaft.

Die Kalenderwoche ist dagegen tatsächlich eine Information, die sonst nicht ersichtlich wäre. In der Monatsansicht wäre diese Information aber besser in der Kalenderdarstellung selbst untergebracht, da sich aus einer Angabe wie  $\gg$ KW: 14-18« nur sehr umständlich ablesen lässt, in welcher Kalenderwoche ein konkretes Datum liegt.

Links von den bisher beschriebenen Objekten befinden sich drei Steuerelemente: Ein Pfeil nach links, der Schriftzug »Heute« und ein Pfeil nach rechts. Über »Heute« gelangt man immer auf das virtuelle Kalenderblatt des aktuellen Tages und der entsprechende Tag wird markiert während man mittels der Pfeile in den drei klassischen Ansichten durch die einzelnen Blätter navigieren kann.

Unter dem Titel »Mehrere Wochen« bietet Lightning eine zusätzliche Ansicht, die in einem späteren Kapitel noch näher beleuchtet wird und sich nicht

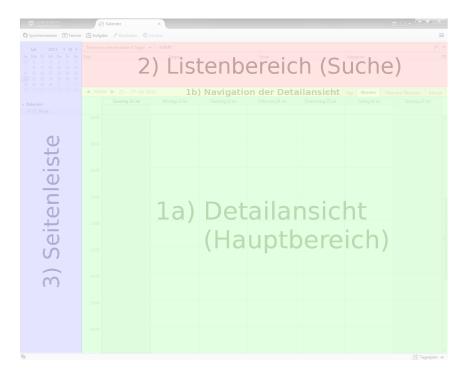


Abbildung 4: Wesentliche Bereiche in Lightning

so stark wie die anderen der Kalenderblatt-Metapher unterwirft. In dieser Ansicht ist das Mapping der Pfeiltasten aber irreführend, weil der Pfeil nach rechts eigentlich in der Ansicht nach unten scrollt, während der linksgerichtete Pfeil nach oben scrollt.

Oberhalb der bisherigen Elemente befindet sich ein Listenbereich (2), in dem Termine untereinander aufgereiht werden. In mehreren Spalten sieht man standardmäßig Titel, Startzeit, Endzeit und Kategorie aller angezeigten Termine. In der Voreinstellung sind hier alle bevorstehenden Termine der nächsten sieben Tage gelistet, über ein Pulldownfeld darüber kann man aber auch einige vordefinierte andere Zeiträume, zum Beispiel »Heutige Termine« oder »Alle zukünftigen Termine« auswählen und in ein Textfeld rechts neben dieser Auswahl kann man Suchbegriffe eingeben, um die Liste zu filtern.

Grundsätzlich ist so eine Listenform für sich gesehen wieder eine ganz eigene Darstellung der Kalenderdaten, die sich allerdings durch ihre Textlastigkeit deutlich von den anderen unterscheidet. Vorteilhaft ist dabei, dass man – sofern man sich nur offene Termine in chronologischer Reihenfolge anzeigen lässt – sehr schnell sieht, was als nächstes fällig ist. Der näheste und damit interessanteste Termin ist immer zuoberst gelistet, danach folgen alle anderen in ihrer zeitlichen Relevanz absteigend. Stark nachteilig ist dagegen, dass ohne grafische Darstellung die Zeitintervalle nur schwer erfassbar sind. Man muss Start- und Endzeiten als Zahlen lesen und sich daraus im Kopf berechnen, wie lange ein Termin dauert beziehungsweise wie viel Zeit zwischen zwei Terminen frei bleibt.

Für die alltägliche Terminplanung ist die Listenform daher ungeeignet. Wahrscheinlich ist sie deshalb auch nicht wie die anderen Ansichten als Regis-

terkarte umgesetzt und wird mitsamt ihren Zusatzfeldern als separate Suchfunktion behandelt. Dass der Bereich auch von den Entwicklern primär für die Suche gedacht ist, lässt sich daraus ableiten, dass dieser in einem Menü ausgeblendet werden kann und dort als »Termine suchen« bezeichnet wird.

Fraghaft bleibt dabei, ob eine Suchfunktion in einem Kalender so wichtig ist, dass sie einen so prominenten Platz sogar noch oberhalb der eigentlichen Kalenderdarstellung verdient. Aus meiner eigenen Erfahrung ist mir nicht ein einziger Fall in Erinnerung, in dem ich oder einer meiner Kollegen eine Suchabfrage auf einen einzelnen Kalender durchgeführt hätten. Wenn ein Termin abgesagt, verschoben oder verändert wird, bekommt man mit den entsprechenden Informationen in der Regel auch schon die entsprechenden Zeitdaten mitgeliefert und Zeiträume von ein paar Tagen bis wenigen Wochen überblickt man meistens auch ohne Suchabfragen recht einfach. Suchfunktionen dienen eher für nicht-alltägliche Tasks, in denen große Zeiträume analysiert oder die Kalender mehrerer Personen durchsucht werden müssen. Bei wirklich komplexen Analysen stößt so eine Suchmöglichkeit dagegen schnell an ihre Grenzen und man wäre eigentlich besser damit beraten, Daten aus dem relevanten Zeitraum zu exportieren und anderweitig, etwa in einem Tabellenkalkulationsprogramm, weiterzuverarbeiten. Deshalb wird der Such-Funktionalität im Rahmen dieser Arbeit auch nur eine vergleichsweise geringe Priorität zugewiesen.

Im linken Bereich des Programmfensters (3) findet man letztlich noch eine kleine Monatsübersicht, die auch zur Navigation genutzt werden kann. Klickt man hier einen Tag an, gelangt man auch in der ausgewählten Kalenderansicht zu diesem Tag. Klickt man einen Monat oder das Jahr an, erhält man hier über ein Dropdown-Menü die Möglichkeit schnell in einen anderen Monat oder ein anderes Jahr zu wechseln. Daneben gibt es auch hier wieder die Möglichkeit ein Kalenderblatt vor- oder zurückzublättern und zum aktuellen Tag zu wechseln. Die Verknüpfung zum aktuellen Tag wird hier aber – vermutlich aus Platzgründen – durch einen Kreis dargestellt. Warum dieses Symbol für » Heute« steht, erschließt sich zumindest mir nicht, aber zumindest weist ein Tooltip auf die Funktion hin.

Die Darstellung des aktuellen und des jeweils markierten Tages ist hier etwas verwirrend. Fallen beide Tage zusammen, ist das entsprechende Feld gelb hinterlegt und blau umrandet. Bewegt man den Mauszeiger über einen anderen Tag, wird dieser auch blau umrandet. Klickt man ihn aber an und bewegt den Mauszeiger weg, ist dieses Feld anschließend gelb hinterlegt und umrandet, während der aktuelle Tag nun komplett blau dargestellt wird. Das ist vor allem auch deshalb irreführend, weil – zumindest unter Microsoft Windows – in der Standardeinstellung markierte Elemente blau dargestellt werden. Hier wird mit dieser Konvention gebrochen: Der markierte Tag ist gelb, der aktuelle blau. Dieses Farbschema setzt sich auch in den einzelnen Kalenderansichten fort, in der Listendarstellung und anderen Standard-Programmelementen ist dagegen wie gewohnt Blau die Farbe markierter Elemente.

Die Funktion der kleinen Monatsübersicht auf der linken Seite ist grundsätzlich zu hinterfragen. Ob sie sinnvoll ist oder nicht, hängt davon ab, in welcher Ansicht man sich befindet. Betrachtet man seinen Kalender in der Monatsansicht, sind sämtliche Daten redundant, in der Tages- und Wochenübersicht erhält man dagegen Informationen zum Kontext und muss den Tag beziehungsweise die Woche nicht isoliert betrachten. Sofern man nicht jeden Tag Termine hat, ist es diesbezüglich auch hilfreich, dass Tage mit Terminen hier in höherem

Schriftgewicht dargestellt werden.

Unterhalb der Monatsübersicht findet man eine Übersicht aller Kalender, auf die man Zugriff hat und kann jeweils durch eine Checkbox auswählen, ob die Termine aus dem entsprechenden Kalender angezeigt werden sollen. Anders als etwa in Microsoft Outlook, wo mehrere Kalender nebeneinander angeordnet sind, werden hier die Termine aller Kalender gemeinsam in einer einzigen Ansicht dargestellt.

Die Unterscheidung mehrerer Kalender erfolgt hierbei durch frei vergebbare Farben, was sich allerdings mit dem Konzept schlägt für Terminkategorien ebenfalls frei Farben vergeben zu können. Damit man zwischen den beiden Farbgebungskonzepten unterscheiden kann, wird ein Termin grundsätzlich immer in der Kalenderfarbe dargestellt und nur wenn ein Termin eine Kategorie mit einer zugewiesenen Farbe hat, wird rechts ein schmaler Streifen entsprechend eingefärbt.

Für Benutzer, die nur einen einzigen Kalender zu verwalten haben, ist diese Kalenderverwaltung ungünstig. Der schmale Farbstreifen stellt in diesem Fall das einzige optische Unterscheidungsmerkmal zwischen Terminen dar und die Kalenderliste besteht zum überwiegenden Teil aus ungenutztem Raum, welcher der eigentlichen Kalenderdarstellung mehr Platz bringen könnte.

Oberhalb aller anderen Elemente bietet der Kalender letztendlich eine Menüleiste mit Schaltflächen zum Synchronisieren von Kalendern sowie zum Hinzufügen, Bearbeiten oder Löschen eines Termins. Das Synchronisieren führt dazu, dass die Kalender neu geladen werden. Aus Usability-Sicht wäre es allerdings besser, wenn die Synchronisation laufend automatisch im Hintergrund geschehen würde und der Benutzer sich dadurch nicht selbst damit befassen muss. Bei Benutzern, die keine externen Kalender angelegt haben, ist diese Schaltfläche weitgehend sinnlos, wird aber trotzdem angezeigt. Die restlichen Schaltflächen versammeln die wichtigsten Funktionen noch einmal an prominenter Stelle, wären grundsätzlich aber auch verzichtbar, da die entsprechenden Aktionen auch auf andere Arten durchgeführt werden können.

# 6.3 Tageskalender

Die Tagesansicht bietet aufgrund ihrer Beschränkung auf einen einzigen Tag den meisten Platz, um Details in der Darstellung unterzubringen. Das Problem dabei ist, dass dieser Platz in Lightning überhaupt nicht genutzt wird. Selbst wenn ein Termin ausreichend Platz für etliche Zeilen Text einnimmt, wird stets nur der Termintitel angezeigt. Will man den Ort oder den Beschreibungstext sehen, muss man die Maus kurz über den Termin halten, bis ein Tooltip mit diesen Zusatzinformationen auftaucht.

Eine wesentliche Stärke der Tagesansicht, die auch in Lightning zur Geltung kommt, ist dagegen die einfache Zuordnung von Uhrzeiten. Diese kann man links an der senkrechten Achse ablesen und solange es keine Überschneidungen gibt, grenzt jeder Termin direkt an die beschriftete Achse an, sodass man dort sehr gut ablesen kann, zu welcher Zeit welcher Termin stattfindet beziehungsweise wann keine Termine eingetragen sind.

Nur im Fall von Uberschneidungen wird das Ablesen etwas schwieriger, weil Termine dann horizontal nebeneinander angezeigt werden. Hier sind vorhandene Hilfslinien nützlich, die jeweils bei einer vollen Stunde eingezeichnet sind.

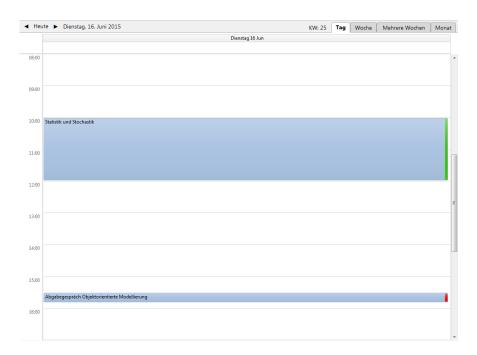


Abbildung 5: Tagesansicht

In Lightning ist es etwas ungünstig, dass die Hilfslinien links genau an der senkrechten Achse enden statt ein wenig über diese hinauszustehen. Bei einem sehr vollen Kalender erschwert dieser Umstand das genaue Ablesen der Uhrzeiten, weil man dann zwar sieht, dass irgendwo eine volle Stunde angeschrieben ist, aber man weiß nicht, ob diese Stunde genau mittig neben der angeschriebenen Zeit, ein wenig darüber, oder ein wenig darunter beginnt. (siehe Abbildung 6)

Bei Unsicherheiten werden die exakte Start- und Endzeit eines Termins aber ebenso wie die anderen Daten in dem Tooltip angezeigt, wenn man den Mauszeiger kurz über den entsprechenden Termin hält.

Standardmäßig wird der Zeitraum von acht bis siebzehn Uhr, also eine typische Bürozeit angezeigt. Über einen Scrollbalken erreicht man auch die anderen Uhrzeiten, die aber etwas abgedunkelt dargestellt sind. An einem Werktag sind diese Zeiten grau und an einem Wochenende grün hinterlegt. Sowohl die Anfangs- und Endzeit als auch die Anzahl der angezeigten Stunden können in den Einstellungen des Kalenders geändert werden.

Die voreingestellte Beschränkung des angezeigten Zeitraums ist durchaus sinnvoll, da bei den meisten Benutzern wohl die Nachtstunden laufend terminfrei sein werden und damit nur unnötigen Platz verbrauchen würden. Problematisch kann sie in dieser Implementierung aber werden, wenn in seltenen Fällen doch einmal ein Abend- oder Morgentermin eingetragen wird, weil man ihn in der Tagesansicht mit hoher Wahrscheinlichkeit übersehen würde. In Lightning müsste man zusätzlich laufend die Listenansicht oberhalb des eigentlichen Kalenders kontrollieren oder am aktuellen Tag ganz nach unten sowie am folgenden ganz nach oben scrollen, um mit Sicherheit keinen Termin zu verpassen.



Abbildung 6: In vollen Kalendern kann das exakte Ablesen der Zeiten schwierig sein, weil die Hilfslinien (fast) vollständig verdeckt sind. Die Bürozeit reicht in diesem Beispiel bis 13:15 Uhr, obwohl man auf den ersten Blick meinen könnte, dass das Ende an der Linie für die volle Stunde (13:00 Uhr) anliegt.

Dass standardmäßig die typische Bürozeit hervorgehoben wird, ist für Leute, die davon tatsächlich betroffen sind, sicher hilfreich. Für andere Personen kann das aber verwirrend wirken und die Einstellung, um das zu ändern ist relativ umständlich zu erreichen. (Im Menü Extras/Einstellungen/Kalender/Ansichten)

Ein wenig gegen bestehende Konventionen ist hier wie schon bei markierten Terminen wieder die Farbgebung. Wenn sich Wochenenden farblich von der Arbeitswoche unterscheiden, werden sie traditionell rot gekennzeichnet. Im Sinn der Arbeit ist das auch sinnvoll, weil man diese Warnfarbe mit einer Sperre verbindet – wie etwa bei Verbotstafeln – und das Wochenende für die meisten Branchen gewissermaßen eine solche Sperre darstellt. Lightning benutzt stattdessen die Komplementärfarbe Grün.

Ein praktisches Hilfsmittel in der Tagesansicht ist eine kleine rote Markierung an der Zeitachse. Diese gibt die aktuelle Uhrzeit an, die bei der Benutzung eines Kalenders natürlich immer wieder von Bedeutung ist.

Oberhalb des Bereiches für zeitliche Detailinformationen befindet sich ein schmaler Streifen für ganztägige Termine, der durch eine etwas dickere Linie vom Rest abgegrenzt ist. Dieser Bereich wird vom Scrollen nicht beeinflusst, sodass man die ganztägigen Daten sinnvollerweise auch in jedem Ausschnitt parat hat und wird automatisch vergrößert, wenn man mehrere ganztägige Termine einträgt, sodass diese untereinander dargestellt werden können.

Oberhalb dieses Bereiches befindet sich nur noch die klarerweise notwendige Bezeichnung des aktuellen Tages mit Wochentag, Monatstag und Monat.

Einen wesentlichen Vorteil von Darstellungsformen mit Zeitachse stellen er-

weiterte Möglichkeiten für Drag-and-Drop-Funktionalitäten dar. Grundsätzlich können in allen Ansichten Termine per Doppelklick erstellt und durch Ziehen mit der Maus verschoben werden. In Ansichten mit Zeitachse ist es aber darüber hinaus möglich Termine zu erstellen, indem man auf der Höhe der gewünschten Startzeit auf eine freie Stelle im Kalender klickt, die Maustaste gedrückt hält und bis zur gewünschten Endzeit nach unten zieht. Sobald man auslässt, ist der vorgegebene Termintitel »Neuer Termin« markiert und man kann ihn sofort umschreiben. Damit lässt sich im Handumdrehen ein Termin erstellen ohne ein neues Fenster dafür öffnen zu müssen.

Ebenso einfach kann man die Terminzeiten nachträglich ändern. Hält man den Mauszeiger über einen Termin, werden oben und unten Markierungen angezeigt, die man mit der Maus verschieben kann, um Start- und Endzeit zu verändern.

Leider gibt es für das komfortable »Aufziehen« eines Termines keine Hinweise im Programm, sodass diese praktische Funktion im Wesentlichen nur nutzen kann, wer sie bereits irgendwoher kennt oder sie durch Zufall entdeckt hat. Auch ich habe diese Funktion nur gefunden, weil mir im Rahmen dieser Arbeit die Idee gekommen ist, dass so etwas praktisch wäre. Andernfalls hätte ich das in Lightning gar nicht erst probiert. Vielleicht würde schon ein veränderter Mauszeiger, etwa in Form eines Fadenkreuzes, besser darauf hinweisen.

Gängige Tastenkombinationen lassen sich auch in Lightning anwenden, um Termine zu löschen oder zu kopieren. Was allerdings fehlt, ist die Unterstützung des Kopierens, indem man einen Termin zieht und dabei die Taste »Strg« gedrückt hält. Stattdessen wird der entsprechende Termin nur verschoben.

Die größte Schwäche der Tagesansicht insgesamt ist der fehlende Kontext. Man betrachtet immer nur einen isolierten Tag respektive einen Ausschnitt davon und muss durch die virtuellen Kalenderseiten blättern, um Informationen zu einzelnen anderen Tagen zu erhalten. Um zwei Tage miteinander zu vergleichen, muss man immer einen der beiden Tage möglichst vollständig im Gedächtnis behalten. Es lässt sich deshalb nur sehr mühsam ablesen, wie es um die Auslastung über mehrere Tage oder gar Wochen bestellt ist, was für die alltägliche Terminplanung ein großes Problem darstellt.

# 6.4 Wochenkalender

Die Wochenansicht kann man grundsätzlich als Erweiterung der Tagesansicht betrachten, in der nicht mehr ein einzelner Tag dargestellt wird, sondern stattdessen die Tage einer Woche horizontal aneinandergereiht sind. Dadurch betrachtet man den aktuellen Tag nicht mehr komplett isoliert, hat aber mit anderen Nachteilen zu kämpfen.

Allem voran wäre grundsätzlich natürlich weniger Platz für Details einzelner Termine vorhanden. Da Lightning aber ohnehin nur den Titel anzeigt, ist das in diesem Fall kein großer Verlust.

Ungünstig für die meisten Personen ist dagegen auch in dieser Implementierung, welche Tage hier angezeigt werden. Am ersten Tag der Woche – in der Standardeinstellung von Lightning also am Sonntag – ist die Ansicht noch ganz nützlich: Man sieht die Termine des aktuellen Tages genauso wie in der Tagesansicht, aber hat rechts daneben auch noch die kommenden Tage der Woche im Blick, um zu wissen, was auf einen zukommt und um mehrere Tage im Voraus planen zu können. Am Montag hat man dann allerdings immer noch den

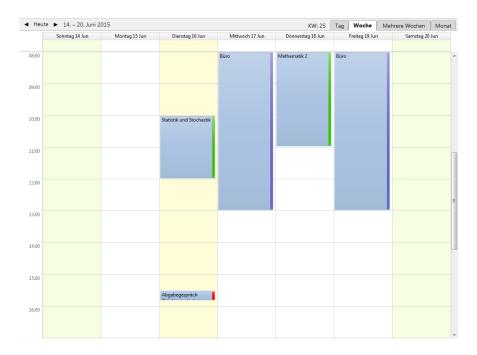


Abbildung 7: Wochenansicht

Sonntag in der Ansicht, der eigentlich schon abgeschlossen und damit für die Terminplanung nicht mehr relevant ist, während man im Gegenzug um einen Tag weniger in die Zukunft sehen kann. Diese Entwicklung geht bis ans Ende der Woche weiter, sodass man am letzten Tag schließlich neben dem aktuellen Tag nur noch vergangene Termine sieht. Der Großteil des Platzes wird in diesem Fall für Information verwendet, die nicht mehr relevant ist. Man muss weiterblättern, wenn man die nächsten Tage sehen will, aber wieder zurück, um zum aktuellen Tag zu kommen.

Mit Fortschreiten des Wochentages wird es in der Wochenansicht auch immer schwieriger die Uhrzeit korrekt abzulesen. Während in der Tagesansicht noch jeder überschneidungslose Termin links an die Zeitachse angrenzt, ist das in der Wochenansicht nur noch am Sonntag der Fall und ab Montag bewegen sich die Einträge immer weiter von dieser Achse weg. Die Hilfslinien werden hier unverzichtbar, um die Zeit ohne Tooltip korrekt abzulesen, allerdings werden sie auch hier wie im Tageskalender durch Termine überdeckt, sodass sie in einem stark verplanten Kalender kaum noch sichtbar sind.

Leicht verändert verhält sich die Anzeige der aktuellen Uhrzeit. Zusätzlich zu der Markierung direkt an der Achse wird hier nämlich auch eine rote Linie auf der selben Höhe eingezeichnet – allerdings nur in der Spalte des aktuellen Tages. Diese zusätzliche Hilfslinie gegenüber der Tagesansicht lässt darauf schließen, dass auch den Entwicklern dieses Kalenders bewusst war, dass die Zeitachse an den weiter rechts gelegenen Tagen nur schwer ablesbar ist.

Was in der Wochenansicht noch auffällt, ist das Verhalten ganztägiger Termine, die sich über mehrere Tage erstrecken. Statt dass diese sich auch in ihrem zugewiesenen Bereich über mehrere Spalten erstrecken, wird in jeder Spalte ein

separater Eintrag mit dem gleichen Titel angezeigt. Ein Eintrag über mehrere Spalten hätte hier den klaren Vorteil, dass der Titel nur einmal angeschrieben werden müsste und darüber hinaus länger sein könnte ohne aus Platzgründen abgeschnitten zu werden.

# 6.5 Monatskalender

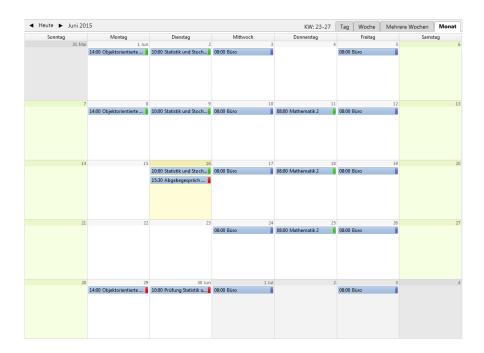


Abbildung 8: Monatsansicht

In der Monatsansicht werden wie in der Wochenansicht die Tage einer Woche in der Horizontalen aneinandergereiht, zusätzlich werden aber alle Wochen des Monats in der Vertikalen untereinander gestellt, sodass man eine Tabelle mit sieben Spalten und fünf Zeilen erhält. Die Wochentage sind als Spaltenüberschriften definiert, die Zeilen haben dagegen keine Bezeichnungen, obwohl die Kalenderwoche hier durchaus sinnvoll wäre. Jedes Tagesfeld ist rechts oben mit einer Zahl gekennzeichnet, die den Tag des Monats repräsentiert.

Da maximal 31 Tage pro Monat nie alle 35 Felder in dieser Tabelle ausfüllen können, sind immer auch Tage mindestens eines angrenzenden Monats enthalten. Diese werden zur Unterscheidung aber mit Grautönen hinterlegt, während der aktuell betrachtete Monat weiß beziehungsweise an Wochenenden grün hinterlegt ist. Am ersten und letzten Tag jedes Monats scheint zur Verdeutlichung auch der Monatsname im Feld auf, damit man die Tage leichter ihrem Monat zuordnen kann.

Was in dieser Ansicht vollkommen wegfällt, ist die Zeitachse. Stattdessen werden die Termine eines Tages hier ähnlich wie in der Listendarstellung, die in Lightning oberhalb der eigentlichen Kalenderansicht liegt, angezeigt. Das bedeutet, dass die Termine eines Tages ihrer Reihenfolge nach untereinander

aufgelistet sind und jeweils nur eine Textzeile für sich beanspruchen. Die Dauer eines Termins oder die freien Zeiten zwischen zwei Terminen kann man hier nicht ablesen. Die einzige zeitliche Information, welche diese Darstellung enthält, ist die Startzeit der Termine. Diese wird jeweils vor dem Termintitel angehängt. Die Endzeit erfährt man nur, wenn man die Maus über einen Termin hält, sodass der übliche Tooltip mit den Detailinformationen auftaucht. Durch das Fehlen der Zeitachse ist es hier auch nicht möglich Termine durch Ziehen mit der Maus zu erstellen oder zu verändern und man muss das separate Detailfenster öffnen um die Zeit festzulegen.

Etwas anders als in den bisher vorgestellten Ansichten werden hier die ganztägigen Termine behandelt. Diesen wird hier kein eigener Bereich mehr eingeräumt, was aus Platzgründen gut nachvollziehbar ist. Stattdessen werden sie nun mit einem dünnen schwarzen Rahmen von den anderen Terminen unterschieden und immer zuoberst gelistet. Anders als bei den anderen Terminen wird hier natürlich auch keine Uhrzeit vor dem Titel dargestellt, da eine solche nicht existiert.

Der Effekt bleibt dadurch ein ähnlicher wie in den anderen Darstellungen, da der Bereich für ganztägige Termine diese auch von den anderen differenziert und immer oberhalb liegt. Etwas nachteilig ist dagegen in der Monatsansicht, dass ganztägige Termine hier horizontal in einer Linie mit konkret terminisierten Einträgen stehen, wenn es an einem benachbarten Tag keinen ganztägigen Termin gibt. Die ganztägigen Termine sind dadurch insgesamt weniger prominent als in den anderen Ansichten und können leichter übersehen werden.

Insgesamt ist es eines der größten Probleme der Monatsansicht, dass für jeden Tag nur sehr wenig Platz verfügbar ist. Obwohl die Termindarstellung schon auf ein Mindestmaß reduziert ist, sind die Felder immer noch zu klein, um an dicht verplanten Tagen alle Termine darzustellen. In solchen Fällen wird innerhalb des entsprechenden Feldes ein Scrollbalken eingeblendet, um die nicht angezeigten Termine erreichen zu können.

Die Monatsansicht ist deshalb wohl vorrangig dazu geeignet sich einen groben Überblick zu verschaffen, welche Tage besonders ausgelastet sind, aber selbst dieser Zweck wird nicht besonders gut erfüllt, weil die Dauer der einzelnen Termine nicht ablesbar ist.

In der Monatsansicht wird außerdem ein Problem noch größer, das schon in der Wochenansicht besteht: Mit dem Fortlaufen der Tage liegen immer mehr der angezeigten Informationen in der Vergangenheit, die für die Terminplanung nicht mehr relevant ist. Im Extremfall ist somit am Monatsletzten nur noch ein einziges der 35 Felder von Bedeutung – also weniger als drei Prozent der gesamten Fläche.

# 6.6 Mehrere Wochen

Neben den klassischen Papierkalender-Metaphern bietet Lightning auch eine Ansicht, die als »Mehrere Wochen« bezeichnet wird und die Flexibilität eines Computers etwas besser nutzt. Grundsätzlich basiert diese auf dem System des Monatskalenders, ändert diesen aber in den Standardeinstellungen so ab, dass zwei der beschriebenen Schwächen abgemildert werden.

Die erste Verbesserung besteht darin, dass die Anzahl der angezeigten Wochen von fünf auf vier reduziert wird. Dadurch hat man nach wie vor Überblick

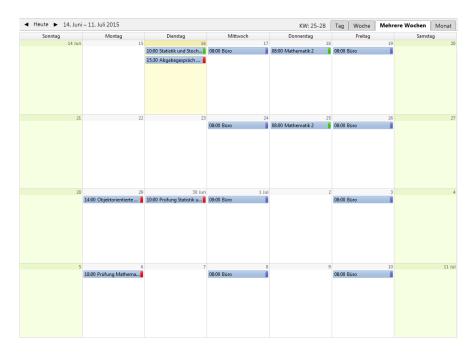


Abbildung 9: Mehrere Wochen

über einen mehrwöchigen Zeitraum, gewinnt aber Platz, um eine etwas größere Anzahl an Terminen pro Tag ohne Scrollbalken darstellen zu können.

Die zweite Verbesserung liegt darin, dass die oberste Zeile immer die aktuelle Woche repräsentiert und man dadurch stets die kommenden Wochen im Blick hat. Dadurch sind zu jedem Zeitpunkt mindestens 22 der 28 Felder für die Terminplanung relevant – das ergibt im jeweils schlechtesten Fall einen Anteil der Gesamtfläche von fast 79 Prozent gegenüber einem Anteil von nur knapp drei Prozent im Monatskalender.

In dieser Wochenansicht blättert man auch nicht durch Seiten, sondern man scrollt in einem kontinuierlich fortlaufenden Raum jeweils um eine Zeile nach unten oder oben und kann sich somit viel flexibler einen bestimmten Zeitraum ansehen. Grundsätzlich könnte man als Benutzer sogar die Anzahl der gleichzeitig angezeigten Wochen verändern, allerdings ist diese Option für den alltäglichen Gebrauch so schwer zugänglich, dass sie wahrscheinlich kaum genutzt wird. (Im Menü Extras/Einstellungen/Kalender/Ansichten)

Durch die Kontinuität und das Wegfallen der Kalenderblätter werden hier natürlich auch keine Monate mehr in Grautönen angezeigt, wie das in der Monatsansicht der Fall ist.

Trotz der Verbesserungen bleiben aber nach wie vor die wesentlichen Probleme der Monatsansicht bestehen. Der Platz pro Tag ist beschränkt, Termindauer und freie Zeiten sind nicht sichtbar, Termine müssen über ein eigenes Fenster erstellt werden und es werden weiterhin – wenn auch weniger – vergangene Tage dargestellt, die für die Terminplanung nicht mehr von Bedeutung sind.

#### 6.7 Termindetails

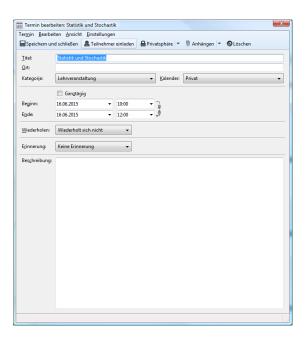


Abbildung 10: Erstellen oder Bearbeiten eines Termins in Lightning

Abgesehen von den verschiedenen Möglichkeiten sich die Terminübersicht anzeigen zu lassen, soll hier auch noch kurz das Interface vorgestellt werden, in dem Termindetails festgelegt werden. Wie in den vorangegangenen Kapiteln beschrieben ist es in der Tages- und Wochenansicht möglich, Termine mit Uhrzeit und Titel zu erstellen ohne dazu ein separates Fenster mit Details nutzen zu müssen. Ebenso lassen sich die Termine dort durch Drag-and-Drop sowie zweimaliges Anklicken ohne Detailfenster bearbeiten. In anderen Ansichten oder für erweiterte Einstellungen ist man aber immer auf die Benutzung des Detailfensters angewiesen.

Wird dieses geöffnet, ist immer automatisch das Textfeld mit dem Titel markiert, das auch das oberste Element darstellt. Das ist sinnvoll, da es sich dabei auch neben dem Zeitraum um die wichtigste Information handelt.

Unter diesem Feld folgen ein Textfeld für den Ort und zwei Dropdown-Felder für die Kategorie und den Kalender – also recht wichtige oder häufig genutzte Ergänzungsdaten.

Die allerwichtigste Information, nämlich die Zeit, wird aber erst nach diesen Elementen angeführt und durch Trennlinien abgegrenzt. Nach der Wichtigkeit gereiht müsste diese Information eigentlich noch vor dem Titel oder zumindest unmittelbar danach stehen, da diese beiden Informationen die einzigen darstellen, die zu einer sinnvollen Termindefinition unbedingt erforderlich sind.

Das Zeitintervall wird durch vier Felder angegeben – jeweils Datum und Uhrzeit für Beginn und Ende. Das Datumsfeld bietet die Unterstützung einen kleinen Monatskalender einblenden zu lassen wie er auch links im Hauptfenster aufscheint, sodass man dort ein Datum auswählen kann ohne es händisch

eintippen zu müssen.

Auf ähnliche Weise kann man sich bei der Uhrzeit eine Hilfe einblenden lassen, um Stunden und Minuten mit der Maus auswählen zu können. Ob das jedoch sinnvoll ist, sollte hinterfragt werden. Ein eingeblendeter Kalender ist praktisch, weil Monate unregelmäßig viele Tage haben und einzelne Tage eines Monats immer wieder auf andere Wochentage fallen. Weiß zum Beispiel jemand, dass er kommenden Dienstag für zwei Tage ein Seminar hat, muss er nicht wissen, welche konkreten Tage das sind, weil er sie im Kalender nachschlagen kann. Bei einer Uhrzeit ist das anders, da diese immer die gleiche Struktur hat. Eine Uhrzeit kennt man oder man kennt sie nicht.

Durch eine Checkbox kann man auswählen, ob ein Termin als ganztägiger Termin gilt. Ist das der Fall, werden die Felder für die Uhrzeiten ausgegraut.

Mittels eines Kettensymbols kann man bestimmen, ob die Termindauer bei Veränderung von Beginn oder Ende gleich bleiben soll, sodass sich der jeweils andere Wert automatisch anpasst. Mir persönlich ist diese Symbolik auch aus einem anderen Programm bekannt, ich kann mir jedoch vorstellen, dass weniger versierte Computernutzer diese noch nicht kennen. Für diesen Fall ist die Kette aber zumindest mit einem Tooltip hinterlegt.

Unterhalb des Zeitraums befindet sich ein Dropdown-Feld für die Wiederholung des Termins. Neben ein paar Standardvorgaben wie »täglich«, »wöchentlich« und »monatlich« findet sich dort auch der Eintrag »Benutzerdefiniert...«. Wählt man diesen aus, öffnet sich ein separates Fenster für komplexere Einstellungen, mit denen man auch so etwas wie etwa »jeden zweiten Donnerstag« definieren kann. Dieses Fenster verwendet aber so viele verschiedene und voneinander abhängige Elemente, dass eine detaillierte Analyse den Rahmen dieser Arbeit sprengen würde, zumal solche Automatismen nicht als Schwerpunkt vorgesehen sind.

Unter dem Feld für Wiederholungen befindet sich ein ähnlich aufgebautes Feld für Erinnerungen, das aus ähnlichen Gründen nicht näher betrachtet wird.

Als unterstes Element findet sich ein großes Textfeld für beliebige Ergänzungen. Meiner Erfahrung nach ist das vor allem im Büroumfeld ein sehr häufig genutztes Feld und sollte demnach eigentlich weiter oben gelistet sein. Aufgrund seiner Größe gegenüber den anderen Elementen ist es hier aber trotz der Reihung am Ende sehr präsent.

Relativ willkürlich erscheint die Integration weiterer Einstellungen. Für die Privatsphäre gibt es in der Menüleiste eine eigene Schaltfläche zur Auswahl zwischen »Öffentlicher Termin«, »Nur Zeit und Datum anzeigen« sowie »Privater Termin«. Die gleiche Funktionalität findet sich noch einmal im Menü unter »Einstellungen«, wo sich auch Priorität und die Stati Vorläufig/Bestätigt/-Abgesagt und Beschäftigt/Verfügbar befinden. Vermutlich ist den Entwicklern selbst bewusst, dass die Status-Funktionen in der Praxis kaum benutzt werden, weshalb sie diese nur in einem unauffälligen Untermenüeintrag einbringen.

In der Menüleiste befinden sich außerdem Schaltflächen zum Speichern und Schließen, zum Löschen des Termins, zum Einladen weiterer Teilnehmer und zum »Anhängen einer Website«, was im Wesentlichen nichts weiter macht als einen Hyperlink einzufügen.

Insgesamt ist dieses Fenster etwas unintuitiv, weil man es von oben nach unten ausfüllt und am Ende für Detaileinstellungen oder das Speichern wieder ganz nach oben muss. Praktisch wäre die Fenstereinteilung, wenn man nur einen Titel einschreibt und dann gleich die Speicherschaltfläche darüber betätigt, aber

für so wenige Informationen müsste man wie bereits beschrieben in der Tagesoder Wochenansicht das Detailfenster gar nicht erst öffnen.

# 7 Alternative Ansätze

In diesem Kapitel werden Kalenderdarstellungen vorgestellt und analysiert, die von dem typischen Schema, wie es im vorangegangenen Kapitel an einer Referenzimplementierung beschrieben wurde, abweichen. Als Quelle wird dabei vorwiegend auf wissenschaftliche Arbeiten verwiesen, die das allgemeine Konzept vorstellen, da konkrete Implementierungen aufgrund der geringeren Verbreitung hier schwieriger zu beziehen sind.

# 7.1 Semantic Zooming

Unter dem Namen Pad wurde 1993 ein Computerinterface-Modell als Alternative zum klassischen WIMP-Desktop vorgestellt. [1] In diesem Modell werden alle Inhalte, genannt Objekte, grundsätzlich auf einer einheitlichen zweidimensionalen Fläche dargestellt. Da physische Anzeigeflächen in ihrer Größe natürlich begrenzt sind, spielt in einem solchen System Zooming, also das Vergrößern und Verkleinern von Ansichten, eine besondere Rolle.

Zu diesem Zweck gibt es in dem Konzept sogenannte Portale, die man sich wie Lupen auf der Arbeitsfläche vorstellen kann. Durch diese Portale kann man die Zoomstufe, in der man Objekte betrachtet, beliebig anpassen. So ist es etwa denkbar Textdateien erst nur ganz klein auf der Arbeitsfläche zu sehen, aber deren Inhalt durch ein Portal in lesbarer Größe anzeigen zu lassen.

Ein weiterer Begriff, der im Rahmen dieses Gesamtkonzeptes fällt, ist jener des Semantic Zooming. Damit wird der Umstand beschrieben, dass die Darstellung eines Objektes von der Zoomstufe abhängig ist, mit der es dargestellt wird. Von einer sehr klein abgebildeten Textdatei könnte etwa erst nur der Titel angezeigt werden, in einer vergrößerten Darstellung eine Zusammenfassung des Inhalts und schließlich der gesamte Text.

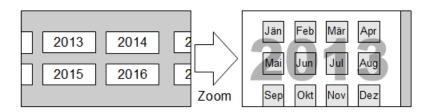


Abbildung 11: Semantic Zooming wie von Perlin et al. [1] beschrieben am Beispiel eines Kalenders: Zoomt man an eine Jahreszahl heran, werden die Monate sichtbar

Als konkrete Beispielanwendung innerhalb dieses Konzeptes wird ein Kalender angeführt. Ausgangspunkt ist ein Objekt am Desktop, das mit »calendar« beschriftet ist. Zoomt man an dieses heran, wird der ursprüngliche Schriftzug allmählich blasser und dahinter scheinen als hinzukommende Details Felder

mit Jahreszahlen auf. Zoomt man noch weiter an diese Felder heran, verblassen auch die Jahreszahlen langsam und man gelangt in eine Aufstellung der Monate. So lässt sich der Kalender bis auf die Tagesebene weiter vergrößern.

Der Benutzer kann dabei in jeder Zoomstufe etwas in den Kalender schreiben oder zeichnen, was dann ebenso verblasst, sobald man weiter hinein zoomt.

Der Vorteil dieser Darstellungsform liegt darin, dass die einzelnen Ansichten logisch miteinander verbunden sind. Während in klassischen Kalendern die Ansichten eher als Alternativen nebeneinander existieren, gibt es hier ein durchgängiges Konzept, das sich auch in einer einheitlichen Darstellung und Bedienung widerspiegelt.

Durch mehrere Portale ist es auch möglich einen Kalender in unterschiedlichen Vergrößerungsstufen gleichzeitig zu betrachten, sodass man sowohl Detailals auch Kontextinformationen direkt zur Verfügung hat. Durch freies Zoomen und eine beliebige Anpassung der Portale lässt sich auf diese Art recht flexibel eine Gesamtsicht auf das System einrichten, welche den persönlichen Bedürfnissen und Vorlieben entspricht.

Sofern es keine Funktionen gibt, die automatisch bestimmte Portale einrichten, ist das persönliche Konfigurieren des Kalenders allerdings nicht nur eine Option, sondern eine Notwendigkeit, um Details und Kontext gleichzeitig im Blick zu haben. Ohne automatische Portale würden die meisten Nutzer sich wahrscheinlich kaum die Mühe einer komplexen Einrichtung machen und stattdessen nur ein Portal benutzen, um den Kalender zu erkunden. Das würde dazu führen, dass sie wieder an die Beschränkungen einer einzigen Ansicht gebunden wären und sich entscheiden müssten, ob ihnen Details oder der Kontext wichtiger sind. Ohne zumindest ein zweites Portal ist die Ansicht sogar noch eingeschränkter als in einem klassischen Kalender, der zumindest in jeder Ansicht zusätzlich eine Monatsübersicht bereitstellt.

Während die Ansichten an sich logisch miteinander verbunden wären, wird die logische Verknüpfung der selbst eingefügten Inhalte in unterschiedlichen Ansichten aber erschwert, wenn diese – wie beschrieben – in größeren Zoomstufen wieder ausgeblendet werden. Man müsste wissen, in welcher Stufe man welche Notizen gemacht hat, um diese wiederzufinden. Kennzeichnet man sich etwa in der Übersicht aller Monate ein, dass im August etwas Bestimmtes zu beachten ist, sieht man diesen Hinweis nicht, wenn man den Kalender auf Tagesebene bedient.

Dass Informationen aus niedrigeren Zoomstufen verblassen statt gleich komplett zu verschwinden, wirkt diesem Problem nur in begrenztem Rahmen entgegen. Darüber hinaus können diese überlagernden Informationen störend wirken, wenn sie noch zu deutlich sichtbar sind und damit die Details verdecken. In den Original-Skizzen zu diesem Kalender sind diese Daten zudem redundant. So steht etwa oberhalb der Übersicht aller Monate die Jahreszahl, obwohl diese auch noch im Großformat über die gesamte Tabelle eingeblendet wird.

Was die Anordnung einzelner Kalenderelemente betrifft, kommt auch hier wieder das Problem zu tragen, dass die Darstellung insgesamt zu stark an der Kalenderblatt-Metapher festhält. Alle Tage sind wie in einer klassischen Monatsansicht in einer Tabelle gelistet. Hat man recht nahe an einen Tag herangezoomt und befindet sich am Ende der Woche, muss man aus dem Kalender herauszoomen oder bis an den Tabellenrand nach links scrollen, um den nachfolgenden Tag zu erreichen. (Die Abbildung in der Konzeptvorstellung lässt vermuten, dass die Woche dort mit Sonntag beginnt.)

Nicht zuletzt hat das Pad-Konzept mit solchen Elementen wie Kalendern ganz allgemein ein konzeptionelles Problem. Zeit, die von einem Kalender ja repräsentiert wird, ist nämlich etwas Kontinuierliches ohne Anfang und Ende und lässt sich damit in einem endlichen Raum nicht vollständig darstellen. In der 1993 veröffentlichten Arbeit sieht man als oberste Kalenderübersicht eine Aufstellung, die gerade einmal die Jahre 1990 bis 1999 beinhaltet. In Anbetracht der Tatsache, dass Hard- und Software oft erneuert werden und man nur selten Jahre im Voraus plant, ist so ein Zeitraum zwar in den meisten Fällen ausreichend, aber erweiterte Potentiale bleiben dadurch ungenutzt. Mit umfangreicheren Kalendern könnte man etwa langfristige Ziele planen (und bei einem eventuellen Systemwechsel mitexportieren) oder den Wochentag eines historischen Datums eruieren.

# 7.2 Spiral Calendar

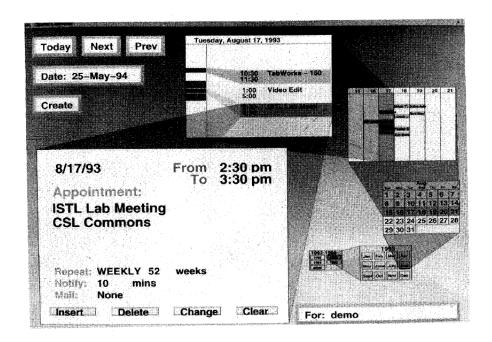


Abbildung 12: Spiral Calendar (Mackinlay et al. 1994 [5])

Der Spiral Calendar [5] verfolgt ein ähnliches Grundprinzip wie der Kalender, der im vorangegangenen Kapitel vorgestellt wurde. Auch hier werden einzelne Ansichten nicht als nebeneinander stehende Alternativen behandelt, sondern als ein Gesamtkonzept, in dem jeweils eine Ansicht eine detaillierte Perspektive auf einen Ausschnitt einer anderen Ansicht darstellt.

Anders als im Pad-Konzept wird hier aber versucht ohne umständliche Konfigurationen den Kontext zu bewahren. In der Praxis sieht das so aus, dass es mehrere diskrete Zoomstufen gibt. Wählt man – um ein Beispiel zu nennen – in der Monatsansicht eine Woche aus, öffnet sich daraufhin die Wochenansicht, aber auch die Monatsansicht bleibt weiterhin erhalten und wird nur etwas ver-

kleinert dargestellt, um Platz für die detaillierteren Informationen zu machen. Danach gelangt man aus der Wochenansicht auf gleiche Weise in die Tagesansicht. Durch eine halbtransparente Markierung zwischen zwei Ansichten wird dabei jeweils angedeutet, welchen Bereich aus der vorhergehenden die aktuelle Ansicht repräsentiert. Klickt man sich durch mehrere Zoomstufen, werden die immer kleiner werdenden Ansichten spiralförmig angeordnet, was dem Kalender seinen Namen gibt.

Wie bei der Evaluierung dieses Konzepts 1994 festgestellt wurde, liegt die Stärke vor allem darin, durch die übergeordneten Ansichten, die weiterhin bedienbar bleiben, schnell zu einem weiter entfernten Datum wechseln zu können. Insgesamt sei es auch im Vergleich zu dem in der Studie benutzten Vergleichskalender Sun CM intuitiver, dass man nur durch Klicks auf die Ansichten durch den gesamten Kalender navigieren kann.

Der letztgenannte Vorteil lässt sich auch an Lightning gut nachvollziehen, da dort der Wechsel mittels Registerkarten erfolgt und damit der Zusammenhang der einzelnen Ansichten nicht sehr offensichtlich ist.

Als nachteilig muss man es betrachten, dass der Kalender aufgrund seiner Spiralform sehr viel Platz benötigt. Für Geräte mit kleinen Displays, zum Beispiel Smartphones, ist dieses Konzept daher ungeeignet. Da die Spiralform aber grundsätzlich eine sehr effiziente Form ist, um Elemente mit absteigender Größe anzuordnen, wäre auch eine andere Anordnung nicht sinnvoll – das Problem liegt vielmehr im Grundprinzip des Kalenders.

Was sich ohne eine Implementierung nur schwer beurteilen lässt, ist das tatsächliche Verhalten dieses Kalenders. Da er ja offenbar je nach Anzahl offener Ansichten in seiner Gesamtheit recht unterschiedliche Formen annehmen kann, indem sich die einzelnen Ansichten verschieben und skalieren, wird es mitunter gegenüber eines eher statischen Systems erschwert bestimmte Elemente wiederzufinden.

Nicht zuletzt bleiben auch die Kritikpunkte zu den einzelnen Ansichten an sich bestehen. Im Spiral Calendar wurden diese bewusst aus klassischen Kalendersystemen übernommen, damit Benutzer nicht umlernen müssen. Die Spiralanordnung macht die Schwächen zwar etwas wett, weil alle Ansichten direkt ansprechbar und damit auch ihre Vorteile nutzbar sind, insgesamt wird die Bedienung dadurch aber auch komplexer, weil unterschiedliche Kalenderdarstellungen gleichzeitig angezeigt werden.

# 7.3 Fisheve Calendar und DateLens

1986 stellte George W. Furnas das Konzept sogenannter Fisheye Views für Darstellungen auf Computern vor. [2] Die Analogie, die dabei verwendet wird, ist jene des in der Fotografie verwendeten Fischaugenobjektives, einer speziellen Form eines Weitwinkelobjektives, durch die das aufgenommene Bild stark verzerrt wird. Objekte in der Bildmitte werden dabei sehr groß dargestellt, mit zunehmender Entfernung wird aber alles Andere im Aufnahmebereich auf immer engeren Platz zusammengestaucht.

Umgelegt auf Informationsvisualisierung bedeutet das, dass zentrale beziehungsweise wichtige Elemente sehr detailliert dargestellt werden, aber trotzdem auch der Kontext erhalten bleibt. Das entspricht laut Furnas auch der Art, wie Menschen mental große Strukturen erfassen – als Beispiel werden geographische Kenntnisse genannt, die in der Regel mit zunehmender Entfernung vom

Wohnort geringer werden.

Decemi	ber 19	86 M	7	w	Th	F	3
Dec 16	*CLEAP (leavi *BELLC 4-6pi with *DINNE Deboi Basil 10t *FINISH	11:30 Lunch  **LEAVE MCC Pack Office Turn in: Badge, keys  **MEET W/RAY ALLARD 3om	17 "Leave Austin 6:30a.m. To North Carolina American Ingl 287 (4 days vacation)	18 "VACATION North Carolina Coas	19 *VACATION North Carolin	20 *VACA North	21 *N.J. A 2:30p Sue at *FURNI put it
Dec 22	22 *BROOI Dinne 6:30 *PACK for CI	10:30a.m. United flight 1037	24 "CHRISTMAS EVE Midnight Church Service	26 "CHRISTMAS "CHRISTMAS @Parent's House 10AM "TOM'S BIRTHDAY Get him a present After Lunch "DINNER W/DAYE Coming over at 6:50 "NUTCRACKER BALLE" 8:30pm		27 *RETUF IV 1:1 Unite Arr	
Dec 29	29	30 "MOVERS Furniture Arrives Find out time "START ARRANGING FURNITURE "-only 3 days to get settled	31	1 *NEW YEARB (Hoorayf) *PARTY at Tom&Lynn's Bpm	2 *BACK TO WO *MARIA'S FIRS At Belicore	3	4
ian 6	6	8	7 *MGC PTAG Starts	8 *MCC PTAC continues	9 °MCC PTAC continues	*MCC ends	11
Jan 12	12	13	14	18	16	17	18

Abbildung 13: Kalender mit Fisheye-Optik (Furnas 1994 [2])

Als einen konkreten Anwendungsfall der Fisheye Views im Computerbereich stellt Furnas am Ende seiner Arbeit kurz einen Kalender vor.

Verglichen mit Lightning erinnert dieser Kalender in seiner Grundstruktur am ehesten an die Ansicht »Mehrere Wochen«, weil er aus einer Tabelle mit sieben Spalten für die Wochentage und mehreren Zeilen für einzelne Wochen besteht, wobei die angezeigten Wochen auch über einen Monat hinaus reichen können.

Der wesentliche Unterschied liegt darin, dass nicht alle Felder gleich groß dargestellt werden. Stattdessen wird dem aktuellen Tag besonders viel Platz eingeräumt, sodass man an diesem alle eingetragenen Details sehen kann. Tage und Wochen danach erhalten je nach ihrer Entfernung immer weniger Raum, vergangene Zeiträume werden offenbar besonders schmal dargestellt.

Durch diese Strukturierung erhält man auf einen Blick die Detailinformationen, die man am häufigsten oder am ehesten benötigt, hat aber trotzdem auch Informationen über einen größeren Gesamtkontext.

Als ungünstig erweist sich in diesem Grundkonzept allerdings die Tabellenform des Kalenders. Wenn hier einem Tag sowohl in der Breite als auch in der Höhe besonders viel Platz eingeräumt wird, betrifft jeweils eine der beiden Dimensionen auch alle anderen Zellen in der selben Zeile oder Spalte. Dadurch wird vor allem auch Tagen besonders viel Breite zugewiesen, die erst in ferner Zukunft liegen. Ist etwa der aktuelle Tag ein Dienstag, dann muss auch der darauf folgende Dienstag die gleiche Breite haben und nimmt damit mehr Platz ein als der Montag davor, der in Hinsicht auf seine zeitliche Nähe aber relevanter wäre.

Damit wird das angedachte Grundprinzip der relevanzabhängigen Größe nur

bedingt umgesetzt. Wirklich konsequent funktioniert es nur innerhalb einer Wochenzeile, weil die Tage hier nebeneinander gereiht sind und nur in ihrer Breite skaliert werden.

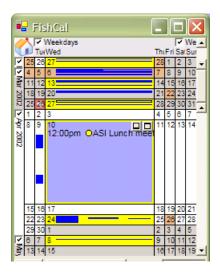


Abbildung 14: Fisheye View in FishCal (Bederson et al. 2003 [3])

In leicht abgewandelter Form wurde das Prinzip der Fisheye Views später wieder aufgegriffen. Benjamin B. Bederson et. al. entwickelten 2003 einen Kalender, den sie erst FishCal [3], später DateLens [4] nannten. Ziel dieser Arbeit war es vor allem ein Interface zu erstellen, das für kleine Displays – damals vorwiegend für PDAs – geeignet ist.

Auch dieser Kalender besteht aus einer mehrwöchigen Ansicht in Tabellenform, in der aber vorerst alle Zellen gleichmäßig verteilt sind. Erst wenn man eine dieser Zellen auswählt, wird diese vergrößert, während alle anderen entsprechend verkleinert werden, um Platz zu machen. Durch diese Änderung geht der Vorteil verloren sofort beim Öffnen des Kalenders die aktuellsten Daten in Blick zu haben, im Gegenzug kann der Benutzer die Darstellung so aber flexibler steuern.

Anders als im ursprünglichen Fisheye-Konzept werden andere Zellen als die aktuelle nicht mit zunehmender Entfernung immer kleiner dargestellt, sondern werden allesamt auf die kleinstmöglichen Maße reduziert, sodass man in den meisten nur noch die Zahl des Monatstages und sonst nichts mehr ablesen kann. Das ist sinnvoll, wenn man bedenkt, dass dieser Kalender vor allem auf Geräte mit kleinen Displays ausgerichtet ist und das aktive Feld sonst nicht ausreichend Platz hätte um sich auf eine gut lesbare Größe auszubreiten. Die Probleme, die schon im ursprünglichen Konzept bestehen, werden dadurch aber noch verschärft. Befindet man sich etwa an einem Dienstag, kann man zwar Informationen zum Dienstag in vier Wochen ablesen, aber keinerlei Informationen zum kommenden Montag. In dieser Hinsicht ist zumindest das manuelle Vergrößern und die Standardansicht mit gleichmäßig verteilten Zellen gut geeignet, um dieses Problem etwas zu verkleinern.

Aufgrund des geringen verfügbaren Platzes kommt in DateLens auch eine

Form von Semantic Zooming zum Einsatz. Da einzelne Termine in der Standardansicht oder in schmäleren Zellen ohnehin nicht lesbar wären, werden diese nur als einfarbige Streifen dargestellt. So kann man trotz der geringen Größe zumindest noch klar ablesen wie stark ein Tag verbucht ist und ob es Terminlücken oder -überschneidungen gibt.

Neben der Vergrößerung eines einzelnen Tages gibt es noch eine andere Möglichkeit die Zellenausmaße zu manipulieren. Am oberen Rand des Programms befinden sich über den entsprechenden Wochentagen zwei Checkboxen für Arbeitstage und Wochenenden, am linken Rand befinden sich Checkboxen für jeden einzelnen Monat. Deaktiviert man ein solches Element, werden die jeweils zugehörigen Zellen auch in der Standardansicht nur noch mit minimaler Breite beziehungsweise Höhe angezeigt.

So etwas könnte zum Beispiel die Übersicht verbessern, wenn man Bürotermine in bestimmten Monaten organisieren will. Bei Wochentagen kann das sicher in einigen Fällen sinnvoll sein und ist in ähnlicher Form ja schon von Papierkalendern bekannt, die Samstag und Sonntag oft verkleinert untereinander statt nebeneinander darstellen. Bei den Monaten wird sich diese Funktion aber nur in wenigen Fällen als sinnvoll erweisen, da man im Alltag ja meistens nur einen oder wenige aufeinanderfolgende Monate benötigt und dann die Anzahl der angezeigten Wochen entsprechend anpassen kann, um nur diesen Bereich im Blick zu haben. Das Deaktivieren ganzer Monate wäre nur interessant, wenn einzelne Monate dazwischen nicht relevant wären, was eher einen Sonderfall darstellen sollte.

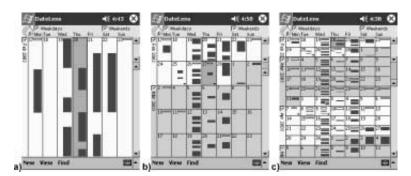


Abbildung 15: Anpassung der angezeigten Wochen in DateLens, a) eine Woche, b) ein Monat, c) drei Monate (Bederson et al. 2004 [4])

Im Gegensatz zu Lightning muss man bei DateLens übrigens in kein Einstellungsmenü wechseln, um die Anzahl der angezeigten Wochen zu manipulieren, sondern kann diese Anpassung direkt im Interface vornehmen. Zu diesem Zweck wird die Scrollleiste erweitert, die sich rechts befindet und in erster Linie dazu dient durch die Wochen zu scrollen. In dieser Leiste befinden sich zwei zusätzliche Elemente: Eines grenzt oben an den verschiebbaren Balken, das andere unten. Das obere ist dabei mit einem Pfeil nach unten gekennzeichnet, das untere dagegen mit einem Pfeil nach oben. Verschiebt man eines dieser Elemente in der Vertikalen, wird dadurch die Größe des Scrollbalkens und die Anzahl der angezeigten Wochen verändert. (siehe Abbildung 15)

Das Mapping hinter dieser Idee ist durchaus nachvollziehbar. Nimmt man

an, dass die gesamte Scrollleiste einen Raum fixer Höhe abdeckt, dann bewirkt eine Verkleinerung des Balkens, dass man weiter scrollen muss, um eine Ende zu erreichen und daher auch der gezeigt Bildausschnitt kleiner sein muss.

Trotzdem sind diese speziellen Steuerelemente auch problematisch. Mit ihren Pfeilen sehen die beiden zusätzlichen Elemente genauso aus wie die Schaltflächen am oberen und unteren Ende der Scrollleiste. Auf den ersten Blick könnte man deshalb glauben, dass hier zwei getrennte Scrollleisten mit fehlendem Balken dargestellt werden. Änderungen an solchen Standard-Elementen wie Scrollleisten sind generell heikel, weil Benutzer diese Elemente kennen und aufgrund ihrer Erfahrungen eine bestimmte Erscheinung sowie ein bestimmtes Verhalten erwarten.

Hinzu kommt, dass die beiden Zusatzelemente hier so weit zusammengeschoben werden können, bis der Balken dazwischen komplett verschwindet (zu sehen in [4] Fig. 2.a.) und damit nicht mehr nutzbar ist. Lässt man sich also nur eine einzige Woche anzeigen, kann man nur noch über die Schaltflächen an den Enden der Scrollleiste navigieren.

Auch ganz allgemein sollte hinterfragt werden, ob Scrollleisten ein geeignetes Mittel sind, um durch kontinuierliche Datenbestände wie Kalender zu navigieren. Aufgrund ihrer Beschaffenheit mit einem Balken, den man bis an eine obere und eine untere Grenze schieben kann, suggerieren Scrolleisten, dass der Inhalt, durch den man scrollt, einen Anfang und ein Ende hat. In einem Kalender existieren solche Grenzen allerdings nicht.

Weil es insgesamt recht vielseitige Möglichkeiten zur Änderung der Darstellung gibt, verfügt DateLens in der linken oberen Ecke über einen Link, der die Standardansicht wiederherstellt. Dieser wird durch ein Symbol in Form eines Hauses repräsentiert. Diese Symbolwahl ist nicht ganz gut nachvollziehbar. Zwar wird auch im Web oft »Home« als Metapher verwendet, um zu dem zurückzukehren, was man ursprünglich gesehen hat – entweder zur Startseite des Browsers oder zur Startseite einer Website – allerdings handelt es sich dort um eigenständige Seiten an ganz speziellen Speicherorten, was das Mapping zu »Home« erleichtert. In DateLens jedoch wird diese Raum-Metapher auf die Zeit angewendet, da man auch wieder in den Bereich des aktuellen Tages gelangt. Raum und Zeit werden allerdings viel eher als zwei getrennte Dinge verstanden als das bei Raum und Speicherplätzen der Fall ist.

Wie man auf mehreren Seiten im Web [13, 14, 15] lesen kann, gibt es seit 2008 auch einen Kalender von YAHOO!, der das Prinzip der Fisheye Views auf eine ähnliche Art wie DateLens aufgreift. Auch hier gibt es eine Monatsansicht, in der man einzelne Tage vergrößern kann, sodass diese eine Tagesansicht zeigen. Allerdings befindet sich dieser Kalender nach wie vor in einer geschlossenen Beta-Phase und wird daher hier nur zur Vollständigkeit erwähnt.

# 7.4 Gantt-Diagramm

Gantt-Diagramme sind zwar grundsätzlich keine Kalender im engeren Sinn, wie es bei den bisher vorgestellten Konzepten der Fall ist, allerdings haben sie ein paar ganz ähnliche Eigenschaften und Probleme. In einer wissenschaftlichen Arbeit aus dem Jahr 2010 [7] zur Verbesserung von Gantt-Diagrammen und Mosaic-Interfaces werden etwa Fokus, Kontext und Zooming diskutiert – also ganze ähnliche Themen wie sie schon in den vorangegangenen Kapiteln immer wieder aufgetaucht sind. Diese Arbeit zeigte unter anderem, dass es sinnvoll

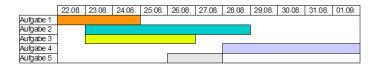


Abbildung 16: Einfaches Beispiel eines Gantt-Diagramms: Aufgaben werden Zeiträume zugewiesen

sein kann, Elemente, die gerade nicht im Fokus sind, je nach Aufgabenstellung entweder ganz oder nur teilweise auszublenden – abhängig davon, ob der Kontext von Bedeutung ist.

Grundsätzlich sind Gantt-Diagramme vor allem ein Management-Werkzeug für langfristige Zeitplanung. Auf der senkrechten Diagramm-Achse werden dabei Aufgaben gelistet, die zu erledigen sind. Die horizontale Achse fungiert als Zeitachse – hier wird zu jeder Aufgabe eingezeichnet, wann sie beginnt und wann sie endet, sodass für alle Aufgaben Balken entstehen.

In vorangegangenen Kapiteln wurden gelegentlich Umbrüche innerhalb von Kalendern kritisiert – etwa der Zeilenumbruch innerhalb einer Tabelle oder die Aufteilung von Informationen auf mehrere Seiten. Ein Gantt-Diagramm entspricht mit seiner fortlaufenden Zeitachse einer Kalenderdarstellung, in der diese Umbrüche komplett wegfallen. (Natürlich gibt es auch hier in der Realität unzählige Varianten an Implementierungen, aber für diese Arbeit wird davon ausgegangen, dass der angezeigte Zeitraum frei wählbar ist ohne zwischen Seiten blättern zu müssen.)

Durch diese Kontinuität ergibt sich eine optimale Nachvollziehbarkeit in Bezug auf die Reihenfolge von Aufgaben beziehungsweise Terminen. Auch wenn ein Termin beispielsweise außerhalb der üblichen Bürozeiten liegt, würde man zwangsweise an diesem vorbeikommen, wenn man von einem Arbeitstag zum nächsten scrollt. Ein Übersehen von Terminen, wie es etwa in der Analyse der Tagesansicht in Lightning beschrieben wurde, ist damit nicht möglich.

Durch die grafische Darstellung ist diese Form aber auch gegenüber einer rein textlichen Auflistung von Terminen – wie es sie in Lightning gibt – von Vorteil, weil man trotz der klaren Abfolge auch auf einen Blick Termindauer und terminfreie Zeiten erfassen kann.

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
	Ausarbeitung Projektkonzept								Design	Studies	
	Ort. Büro								Ort: Hörs	saal 7	
1											
١	Projekt startet kommenden Montag								Kapitel 7	'-8	
ı	Teilnehmer: Hr. Jambos, Fr. Bergen										

Abbildung 17: Ein Kalender nach dem Muster eines Gantt-Diagramms könnte den vertikal verfügbaren Platz für Termindetails nutzen.

Da die Aufgaben auf der senkrechten Achse bei einer Nutzung als Terminkalender nicht benötigt werden, könnte man die schmalen Balken auf die gesamte Höhe erweitern und in diesen sämtliche Termindetails unterbringen ohne dafür eine andere Ansicht oder ein separates Fenster öffnen zu müssen (siehe Abbildung 17). Nur im Fall von Terminüberschneidungen müsste dieser Raum geteilt werden.

Noch ein Vorteil wäre, dass man keinen gesonderten Bereich für ganztägige oder gar mehrtägige Termine benötigt, weil man diese in einer solchen Darstellungsform problemlos wie jeden anderen Termin auch eintragen könnte, ohne dass er irritierend wirkt.

Auf der anderen Seite würde eine solche Darstellung aber natürlich auch Nachteile mit sich bringen.

Allem voran bildet die kontinuierlich fortlaufende Zeit den üblichen Ablauf eines menschlichen Lebens nicht gut ab. Jeder Mensch benötigt Schlaf und wird sich für gewöhnlich auch in seiner Wachzeit nur begrenzte Zeitfenster mit Terminen belegen. Verplant man nur Arbeitstage von acht bis siebzehn Uhr, bleiben fast 63% der Kalenderfläche eines Tages ungenutzt – diese Flächen müssten aber bei der Darstellung mehrerer Tage trotzdem mit angezeigt werden. In der ursprünglichen Nutzung von Gantt-Diagrammen besteht dieses Problem nur deshalb nicht, weil dort in der Regel nur langfristige Aufgaben eingetragen werden, die ohnehin über mehrere Tage reichen.

Dieser Tageszyklus bringt auch Aufgaben mit sich, die mit einer solchen kontinuierlichen Zeitachse nur schwer zu bewältigen sind. Zum Beispiel lässt sich nur umständlich feststellen, an welchem Tag einer bestimmten Woche man schon am frühesten keine Termine mehr hat, weil man an jedem einzelnen Tag die entsprechende Zeit ablesen und mit allen anderen Zeiten vergleichen muss. In einem klassischen Wochenkalender wäre so etwas dagegen schon auf einen Blick zu sehen, weil die einzelnen Tage zum Vergleichen günstig angeordnet liegen, indem gleiche Zeiten auf der selben (gedachten) Linie liegen.

Auch die vertikale Platznutzung könnte recht ineffizient sein. Man hätte zwar ausreichend Raum für viele Termindetails, wenn man sich allerdings nicht viele Details notieren muss, bleibt dieser Raum weitgehend ungenutzt.

Was insgesamt problematisch sein könnte, ist das Verhältnis zwischen Details und Kontext. Damit Termine auch horizontal ausreichend Platz haben, um darin den Titel und eventuell weitere Details lesen zu können, müsste so ein Kalender in einer recht hohen Zoomstufe angezeigt werden, in der dann aber nur noch wenige Termine nebeneinander passen – im Extremfall kann dann nicht einmal mehr ein ganzer Tag gleichzeitig angezeigt werden.

# 8 Prinzip eines Tagbalkenkalenders

Basierend auf den Analysen der bisherigen Kapitel wird nun versucht ein Konzept aufzustellen, das die individuellen Stärken der einzelnen Systeme in einer neuen Darstellungsform vereint. In diesem Kapitel werden erst die groben Eigenschaften umrissen, bevor dann nachfolgend in einem Prototypen weitere Details ausgearbeitet werden.

Wie ausführlich diskutiert wurde, sind Umbrüche in den meisten Fällen ein Störfaktor, weil man zwischen mehreren Seiten wechseln oder mit dem Blick in eine neue Zeile wechseln muss, um mehrere Tage miteinander vergleichen zu können. Am Beispiel des Gantt-Diagramms ist andererseits aber ersichtlich, dass eine durchwegs kontinuierliche Zeitachse für die alltägliche Terminplanung ungeeignet ist, weil diese Darstellung den Vergleich einzelner Tage erschwert.

Einzelne Tage sollten deshalb wie in einer klassischen Wochenansicht nebeneinander gelegt werden. Allerdings soll dabei die Seitenaufteilung der Kalenderwochen aufgelöst werden, sodass man am Ende der Woche nicht neben dem aktuellen Tag nur noch vergangene Termine im Blick hat. Stattdessen sollte ein freies Scrollen möglich sein. So kann man sich den Kalender grob als einen unendlich langen, verschiebbaren Balken vorstellen, auf dem Tage nebeneinander angeführt sind. Aus diesem Grund soll dieses neue Konzept Tagbalkenkalender, oder kurz TaBaKa, genannt werden.

Durch die freie Verschiebbarkeit hat man eine ähnliche Flexibilität wie in der mehrwöchigen Ansicht von Lightning. Durch die Struktur, die aber von der Wochenansicht übernommen wird, ist es möglich sich zu jedem Zeitpunkt nur zukünftige Daten anzeigen zu lassen und alles Vergangene vollständig auszublenden. Standardmäßig soll deshalb auch der aktuelle Tag immer ganz links liegen, sodass sich vergangene Tage nicht mehr im Sichtfeld befinden und der Zeitraster direkt an dem aktuellen Tag anliegt.

Auch wenn die Darstellung am ehesten der Wochenansicht entspricht, soll aber nicht zwangsläufig immer ein Zeitraum von sieben Tagen angezeigt werden, sondern es soll wie in DateLens die Möglichkeit geben den angezeigten Bereich direkt über das Interface zu manipulieren, sodass man flexibel zwischen mehr Details oder mehr Kontext entscheiden kann.

Damit der Kalender trotzdem übersichtlich bleibt, sollen sich die in der Übersicht angezeigten Details einzelner Termine im Sinne von Semantic Zooming an den verfügbaren Platz anpassen. So können bei ausreichend Platz Termintitel, Ort und zumindest die ersten Worte des Beschreibungstextes angezeigt werden, bei weniger Platz soll erst der Beschreibungstext und dann der Ort eingespart werden. Es wäre auch möglich automatische Abkürzungen zu implementieren, sodass zum Beispiel bei ganz wenig Platz nur noch die Anfangsbuchstaben eines längeren Termintitels angezeigt werden.

Aus dem Kalenderprinzip der Fisheye Views soll wie in DateLens zumindest eine zweistufige Unterscheidung übernommen werden, sodass dem aktuellen Tag mehr Breite eingeräumt wird als dem Rest, damit man an diesem Tag fast die gleichen Platzvorteile wie in einer klassischen Tagesansicht hat. Da alle Tage nebeneinander liegen und nur in ihrer Breite skaliert werden müssen, fallen alle beschriebenen Nebeneffekte weg, die mit der Anwendung in einer Tabellendarstellung einhergehen.

Insgesamt soll es ein wesentliches Merkmal von TaBaKa sein, dass im Gegensatz zu klassischen Kalendern wie Lightning nur noch eine einzige Ansicht benötigt wird. Damit fallen viele beschriebene Probleme wie redundante Informationen und fehlerhaftes Mapping automatisch weg. Diese Einzelansicht führt weiter auch dazu, dass die Bedienung des Kalenders immer gleich ist. Das direkte Erstellen und Bearbeiten von Terminen durch Ziehen mit der Maus kann daher für das ganze Kalendersystem als allgegenwärtige Option zur Verfügung stehen.

Wie bei der Analyse von DateLens beschrieben ist eine Scrollleiste kein allzu passendes Steuerelement, um durch einen kontinuierlich fortlaufenden Kalender zu navigieren. Stattdessen soll ein Element eingeführt werden, das in seiner Funktion etwa dem kleinen Monatskalender entspricht, der in Lightning unabhängig von der ausgewählten Ansicht links eingeblendet wird. Da in Ta-BaKa aber diese klassische Tabellenform fallengelassen werden soll, wird dieses Konzept auch auf dieses Element übernommen, sodass man statt eines Monats-

kalenderblattes auch hier einen Tagesbalken erhält. Im Wesentlichen soll dieses Steuerelement also eine stark verkleinerte Version der eigentlichen Ansicht sein, sodass man zwar keine Termindetails mehr darauf sieht, aber Kontextinformationen über Monate und Wochentage geliefert bekommt. Auf diesem Element kann man entweder beliebige Tage anklicken, um zu diesen zu wechseln oder den Balken mit der Maus nach links oder rechts ziehen, um andere Zeiträume in den Fokus zu rücken und damit gleichzeitig den angezeigten Bereich des Kontextbalkens zu verändern.

Um das Mapping zwischen dem Navigationsbalken und der detaillierteren Ansicht zu erleichtern soll die Darstellung ähnlich erfolgen, wie das im Spiral Calendar zwischen zwei aufeinander folgenden Ebenen geschieht. Das bedeutet, dass eine Art Trichter dazwischen eingezeichnet wird, um zu verdeutlichen, welchen Bereich des Kontextbalkens man gerade im Detail betrachtet.

Der Raum, den der Kalender einnimmt, soll in maximalem Ausmaß für die Hauptanwendungsfälle – also im Wesentlichen für das Lesen und Bearbeiten von Terminen – genutzt werden. Erweiterte Funktionen wie das Anzeigen freigegebener Kalender oder das Durchführen von Suchabfragen sollen in eigene Menüs ausgelagert werden, um Benutzern, die dafür keine Verwendung haben, nicht im Weg zu sein.

# 9 Low-Fidelity-Prototyp

Dieser erste Prototyp besteht aus Grafiken, die durch Darstellung der Benutzeroberfläche eine möglichst vollständige Vorstellung von der geplanten Funktionsweise des Programms vermitteln sollen.

TaBaKa ist in diesem und dem nachfolgenden Kapiteln grundsätzlich als Webanwendung konzipiert, weil dadurch die Portierbarkeit auf unterschiedliche Plattformen sowie die Freigabe für andere Personen erleichtert werden und eine eventuelle, aufwändige Synchronisation zwischen mehreren Geräten damit obsolet wird.

## 9.1 Hauptfenster

In Abbildung 18 ist ein Zeitraum von sieben Tagen ausgewählt, sodass sich die Darstellung gut mit herkömmlichen Wochenansichten vergleichen lässt. Abbildung 19 zeigt den gleichen Bereich in einem Bildschirmformat für tragbare Geräte und Abbildung 20 zeigt einen Kalender, in dem ganze 31 Tage dargestellt werden. Trotz dieser Unterschiede kommen alle drei Anwendungsarten mit einer einzigen, flexibel skalierbaren Ansicht aus.

Die Grundstruktur sollte den meisten Benutzern schnell vertraut sein, da sie einem klassischen Kalender auf den ersten Blick recht ähnlich sieht.

Die wesentlichsten Unterschiede sind dabei die unterschiedlichen Spaltenbreiten und die Tatsache, dass der Kalender nicht zwangsweise mit einem Montag beginnt und dadurch über eine Kalenderwoche hinausgehen kann.

Die Spaltenbreiten sind dabei wie in den vorangegangenen Kapiteln beschrieben grob an das Prinzip von Fisheye Views [2] angelehnt. Der zentral betrachtete Tag – standardmäßig der aktuelle – erhält immer mehr Platz als die restlichen Tage und ist ganz links angeordnet, sodass man ein gutes Verhältnis aus Detailund Kontextinformationen erhält.

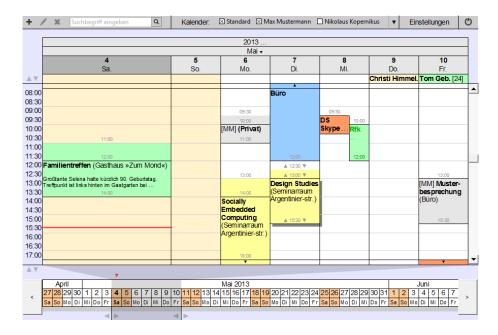


Abbildung 18: Low-Fidelity-Prototyp: Hauptfenster mit 7-Tage-Intervall

## 9.1.1 Navigation

Die Navigation innerhalb des Kalenders erfolgt primär durch die Leiste unterhalb, die eine weitläufigere aber undetaillierte Übersicht über den Kontext liefert. Eine graue Markierung zeigt dabei an, welcher Teilbereich dieser Leiste gerade ausgewählt ist und oberhalb in höherem Detailgrad angezeigt wird.

Um einen anderen Bereich in den Fokus zu bekommen und Bereiche zu sehen, die aktuell außerhalb des Sichtbereiches liegen, kann man die Leiste mit der Maus anklicken und bei gedrückter Maustaste nach links oder rechts verschieben. Der graue Bereich bleibt dabei an der selben Stelle, sodass man einen beliebigen anderen Zeitraum in diesen hineinschieben kann. Alternativ kann man die Leiste auch mit den Schaltflächen am linken und rechten Rand verschieben.

Eine Frage bei der Entwicklung dieses Prototypen war es, ob es sinnvoller wäre statt der gesamten Leiste nur den grauen Bereich durch Ziehen verschiebbar zu machen. So könnte man auch den angezeigten Bereich verändern. Der Vorteil dieser Variante könnte sein, dass sie einem Benutzer aufgrund ihrer Ähnlichkeit mit einem Scrollbalken vertrauter ist. Die Wahl ist jedoch auf die andere Variante gefallen, weil diese flexiblere Bewegungen ermöglicht. Wie später noch näher beschrieben wird, lässt sich auch der graue Bereich anpassen und es könnte dadurch etwa zu einer Situation kommen, in der er bis zum linken Rand reicht. In dieser konkreten Situation könnte man den Bereich nicht mehr weiter nach links verschieben, ohne zuvor mit der entsprechenden Schaltfläche weiter nach links zu scrollen. In der gewählten Lösung kann man die Ansicht dagegen zu jedem Zeitpunkt um eine gesamte Breite nach links oder rechts verschieben, indem man einfach den Balken an einem Ende anklickt und ans andere hinüber zieht.

Alternativ wäre auch noch eine Mischform denkbar, in der man mit einem

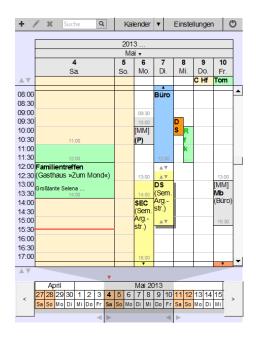


Abbildung 19: Low-Fidelity-Prototyp: Hauptfenster mit 7-Tage-Intervall auf einem kleineren Bildschirm im Hochformat

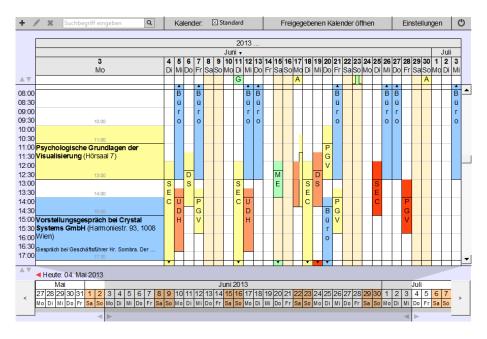


Abbildung 20: Low-Fidelity-Prototyp: Hauptfenster mit 31-Tage-Intervall

Klick auf den grauen Bereich diesen und bei einem Klick auf einen anderen Bereich den ganzen Balken verschieben kann. Vorerst soll es aber bei der einfacheren Variante bleiben, um die Benutzung nicht zu komplex werden zu lassen. Im späteren Usability-Test soll sich zeigen, ob die gewählte Variante von realen Benutzern angenommen wird oder eine Mischform doch intuitiver wäre, weil Personen versuchen den grauen Bereich zu verschieben.

Neben der Zieh-Operation kann man auch durch die Leiste navigieren, indem man einen bestimmten Tag nur kurz anklickt. In diesem Fall wird der Balken automatisch so verschoben, dass dieser Tag an der Position zu liegen kommt, an der ursprünglich der aktuelle Tag war. So kann man sich jeden Tag auch in der breiteren Spalte anzeigen lassen.

Der aktuelle Tag ist in der Navigationsleiste in Fettschrift geschrieben und wird noch einmal separat durch einen roten Pfeil gekennzeichnet. Verschwindet der aktuelle Tag im Zuge des Navigierens aus dem sichtbaren Bereich, zeigt der rote Pfeil am entsprechenden Ende der Leiste in die Richtung, in welcher sich der Tag befindet und unmittelbar daneben wird das entsprechende Datum angeschrieben (zu sehen in in Abbildung 20). Somit sieht man allein aufgrund der Position dieses Elementes, ob man sich gerade in der Zukunft oder in der Vergangenheit befindet. Ein Klick darauf rückt den entsprechenden Tag wieder in den Fokus.

Die Navigation über die Leiste setzt voraus, dass man immer zu einem Datum navigieren möchte, das in zeitlicher Nähe liegt. Diese Annahme wird zwar in den meisten Fällen korrekt sein, wenn man doch einmal einen größeren Zeitraum überspringen will, ist dieser Umstand allerdings ungünstig. Aus diesem Grund kann man auch die Jahres- und Monatsangabe oberhalb der Terminübersicht direkt manipulieren. Neben dem Monatsnamen befindet sich ein Pfeil nach unten; Klickt man den Monat an, öffnet sich ein Pulldown-Menü mit allen zwölf Monaten, aus denen man einen beliebigen auswählen kann. Klickt man dagegen auf die Jahreszahl, wird diese markiert und man kann sie direkt umschreiben. Nach einer solchen Änderung wird darunter der gleiche Zeitraum wie zuvor aus dem gewählten Monat oder Jahr angezeigt.

## 9.1.2 Termindarstellung



Abbildung 21: Termindarstellung im Low-Fidelity-Prototypen: Je enger eine Spalte ist, desto weniger Termindetails werden angezeigt.

Der verfügbare Platz innerhalb einer Spalte bestimmt jeweils, wie viele Details eines Termins angezeigt werden (siehe Abbildung 21). Bei besonders viel Platz werden zu jedem Termin Titel, Ort, Beschreibungstext, Start- und Endzeit angegeben, wobei längere Beschreibungstexte bei Bedarf abgeschnitten werden und eine Ellipse andeutet, dass der Text noch länger ist. Bei zunehmendem Platzmangel werden erst Beschreibungstext, dann die Endzeit und dann der

Ort eingespart. In noch härteren Fällen wird der Titel abgekürzt und schließlich komplett weggelassen, sodass letztendlich nur noch ein farbiger Balken übrig bleibt, der zumindest noch einen Hinweis auf eine belegte Zeit gibt. Damit trotzdem immer alle wesentlichen Informationen schnell zugänglich bleiben, werden diese wie in Lightning in einem Tooltip angezeigt, wenn man seine Maus ein paar Sekunden über einem Termin hält.

Überschneiden sich zwei oder mehr Termine, wird automatisch beim Benutzer nachgefragt, welcher davon tatsächlich wahrgenommen werden soll. Diesem Termin wird dann mehr Platz eingeräumt und die anderen, deren Zeitstatus automatisch auf »Verfügbar« gesetzt wird (siehe auch spätere Erläuterungen zu Termineigenschaften), werden blasser dargestellt.

Die Darstellung der einzelnen Termininformationen richtet sich ebenso nach deren Wichtigkeit wie die Reihenfolge des Verschwindens bei wenig Platz. Der Titel wird in Fettschrift dargestellt, der Ort in normalem Schriftgewicht und der Beschreibungstext in geringerer Schriftgröße.

Start- und Endzeiten werden in kleiner, grauer Schrift dargestellt, weil sie nur ein Hilfsmittel sind. Grundsätzlich könnte man zwar jede Zeit an der Leiste links ablesen, aber wie in der Analyse von Lightning schon festgehalten wurde, wird das mit zunehmender Entfernung nach rechts immer schwieriger. Uhrzeiten werden immer einheitlich vor der Linie angeschrieben, welche die entsprechende Zeit repräsentiert, aber weggelassen, wenn sich dort kein Platz findet.

Neben der Start- und Endzeit kann ein Termin auch über eine Anreisezeit verfügen. Diese wird vor dem Termin in der gleichen Farbe wie der Termin selbst eingezeichnet, hat im Gegensatz zu diesem aber keinen eigenen Rahmen, sodass zwar die Zugehörigkeit erkennbar bleibt, aber doch ein deutlicher Unterschied zwischen Termin und Anfahrt sichtbar ist.

Neben Fahrzeiten kann der Kalender auch einen Zähler bei sich wiederholenden Terminen ausgeben. In Abbildung 18 befindet sich am Freitag ein Geburtstag als ganztägiger Termin, bei dem auch das Alter angegeben wird.

Da in Lightning die Färbung eines markierten Termins verwirrend sein kann, behalten markierte Termine in TaBaKa ihre Farbe, die sich nach der jeweiligen Kategorie richtet und werden nur durch einen Schattenwurf hervorgehoben.

In diesem aktiven Status werden bei allen zugehörigen Uhrzeiten Pfeile eingeblendet, die andeuten, dass man wie in Lightning die Zeiten durch ein Ziehen mit der Maus verändern kann. Bei wenig Platz werden die Uhrzeiten komplett durch Pfeile ersetzt, es wird aber ein Tooltip mit der Zeit angezeigt, sobald man seine Maus darüber hält. Fallen beim Verschieben die Abreisezeit und die Startzeit zusammen, bedeutet das, dass keine Anreise mehr eingeplant ist und man diese bei Bedarf im Termindetailfenster neu festlegen muss.

Weil es in Lightning problematisch ist, wenn Termine außerhalb des dargestellten Uhrzeit-Intervalls liegen und diese dadurch nicht sichtbar sind, gibt es in TaBaKa oben und unten am Kalender eine schmale Zeile, in der auf solche Termine hingewiesen wird. Ist etwa ein Termin am späten Abend eingetragen, wird die Zeile am entsprechenden Tag in der Farbe des Termins eingefärbt und ein Pfeil nach unten symbolisiert, dass sich unterhalb des angezeigten Bereiches noch etwas befindet. Bei mehreren Terminen erhält der Hinweis die Farbe des zeitlich nähesten. Damit man nicht extra scrollen muss, um an weitere Informationen zu diesen Terminen zu gelangen, wird ein Tooltip angezeigt, wenn man seine Maus über die Hinweisfläche hält. Dieser zeigt die Informationen des nächsten Termines und die Gesamtzahl aller nicht angezeigten Termine im

selben Bereich an.

Werden Kalender anderer Personen eingeblendet, werden die entsprechenden Termine grundsätzlich immer in Grau dargestellt, weil davon ausgegangen wird, dass jeder Benutzer seinen Bedürfnissen entsprechend individuelle Kategorien und Farben einrichtet, sodass andernfalls Inkonsistenzen herrschen würden. Nur dann, wenn man den eigenen Kalender ausblendet und sich nur einen einzigen externen Kalender anzeigen lässt, wird dieser in seinen üblichen Farben dargestellt.

Damit mehrere gleichzeitig eingeblendete Kalender trotzdem unterscheidbar bleiben, wird in allen externen Kalendern vor dem Termintitel ein automatisch generiertes Namenskürzel in eckigen Klammern eingeblendet. Im Standardfall sind das die Initialen; Sollten diese nicht eindeutig sein, werden weitere Buchstaben des Namens einbezogen.

Der Grauton externer Kalender ist so gewählt, dass der selbe Farbton nicht als Option für eine Kategoriefarbe zur Verfügung steht, sodass man seinen eigenen Kalender immer auf einen Blick von externen Kalendern unterscheiden kann.

Externe Kalender in einheitlichen Grautönen sind untereinander vielleicht nicht so gut unterscheidbar wie durch verschiedene Farben, aber wie in der Analyse von Lightning schon festgestellt wurde, sind zu viele Farbkodierungen eher verwirrend als hilfreich. TaBaKa geht daher den Weg, Farben für Termine ausschließlich als Kategoriekennzeichnung einzusetzen.

Für die häufigsten Anwendungsfälle sollte die einheitliche Graukodierung kein Problem darstellen, weil bei mehr als zwei dargestellten Kalendern ohnehin sehr schnell der Platz zu eng wird und man diese Einstellung daher nicht oft verwenden wird. Lässt man sich doch mehr Kalender gleichzeitig anzeigen, dient das in den meisten Fällen wohl dafür, um eine gemeinsame, freie Terminzeit zu finden – in diesem Fall wäre es aber nur nebensächlich, welcher Termin aus welchem Kalender stammt und das Kürzel im Titel ist ausreichend.

### 9.1.3 Anpassung der angezeigten Intervalle

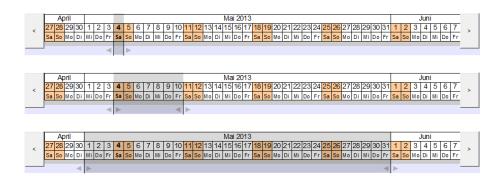


Abbildung 22: Anpassung des angezeigten Zeitraums im Low-Fidelity-Prototypen: Durch Verschieben der linken und der rechten Grenze in der Navigationsleiste lässt sich ein beliebiges Intervall auswählen. Auch die klassische Tages- oder Wochenansicht lässt sich bei Bedarf so nachbilden.

Am Rand des grauen Bereiches in der Navigationsleiste sind Pfeile eingeblendet, die darauf hindeuten, dass man die Ränder mit der Maus verschieben kann. Dadurch lässt sich ein beliebiger Zeitraum eingrenzen, der oberhalb mit Termindetails dargestellt wird. In den beiden Extremen erhält man durch diese Anpassung entweder eine klassische Tagesansicht, oder eine Übersicht über den gesamten Bereich der Navigationsleiste.

Wie viele Tage die Navigationsleiste umfasst, ist dabei von der Auflösung und Breite des benutzten Monitors beziehungsweise der Breite des Browserfensters abhängig. Auf einem Breitbildmonitor kann man somit einen sehr weiten Überblick generieren, während man auf einem Smartphone mit wenigen Wochen auskommen muss. Verkleinert man das Browserfenster so, dass der gewählte Bereich nicht mehr ausreichend Platz findet, wird dieser automatisch verkleinert.

In der Voreinstellung ist der graue Bereich immer so gewählt, dass der aktuelle Tag ganz links liegt und die Navigationsleiste ist so positioniert, dass maximal sieben Tage vor dem aktuellen Datum angezeigt werden, sodass der Großteil der Leiste die weitaus relevantere Zukunft anzeigt. Anders als in der Detailansicht wird hier die Vergangenheit nicht als vollkommen irrelevant betrachtet, weil sie hier nur sehr wenig Platz benötigt und man bei Bedarf mit nur einen Klick etwas nachschlagen kann.

Auf ähnliche Weise lässt sich auch direkt in der Hauptansicht der Bereich der dargestellten Uhrzeiten konfigurieren. Auch diese vertikale Leiste hat an beiden Enden Pfeile, die eine Möglichkeit zum Verschieben anzeigen. Sobald man einen solchen Bereich anklickt, wird als Tooltip die aktuelle Grenzzeit angegeben und wenn man die Maus mit gedrückter Taste nach oben oder unten zieht, ändert sich die Zeit im Tooltip entsprechend. Sobald man die Taste wieder loslässt, wird die neue Zeit als Grenze gesetzt und die Kalenderdarstellung entsprechend angepasst.

Die Intervalle lassen sich dabei in Stunden-Schritten anpassen. Wie weit man die Maus für eine bestimmte Einstellung bewegen muss, ist von der aktuellen Darstellung abhängig. Bewegt man die Maus zur Verengung des Intervalls nach innen, ist die neue Grenzzeit mit der jeweils auf der Leiste angeschriebenen Zeit identisch. Bewegt man die Maus nach außen, ist der verfügbare Spielraum in gleichmäßige Intervalle zwischen 00:00 beziehungsweise 24:00 Uhr und der aktuellen Grenzzeit unterteilt.

## 9.1.4 Menüleiste

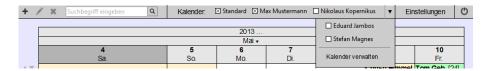


Abbildung 23: Low-Fidelity-Prototyp: Menüleiste

Am oberen Seitenrand befindet sich im Hauptfenster eine Menüleiste.

Ganz links sind dort Symbole für die Basisoperationen zur Terminverwaltung angebracht. Mit diesen kann man wie in Lightning neue Termine erstellen, bearbeiten oder löschen. Grundsätzlich kann man zwar auch Termine durch Ziehen mit der Maus erstellen, durch Doppelklick zum Bearbeiten öffnen und durch

Markieren und Drücken der Entfernen-Taste löschen, aber weil diese Möglichkeiten nicht direkt sichtbar sind, werden auch diese Alternativen geboten.

Unmittelbar daneben findet sich ein kleines Suchfeld. Wie in vorangegangenen Kapiteln beschrieben, wird der Suche im Rahmen dieser Arbeit nur untergeordnete Bedeutung zugemessen, weshalb sich auch die Funktionalität nur auf das Wesentlichste beschränkt. Gibt man hier einen Suchbegriff ein und startet eine Suche, werden alle Termine, die den gesuchten Begriff in keinem Feld enthalten, nur noch blass dargestellt, sodass die Suchtreffer hervorstechen.

Nach einer solchen Suche wird das Lupensymbol zu einem Löschsymbol. Klickt man auf dieses, wird das Suchfeld wieder zurückgesetzt und der Kalender wieder normal dargestellt. Alternativ kann man auch einfach den Inhalt des Textfeldes löschen. Ändert man den Suchtext nach einer durchgeführten Suche noch einmal, wird das Lupensymbol wieder hergestellt, um eine neue Suche starten zu können.

Neben der Suche findet man die Liste aller freigegebenen Kalender, die man durch Checkboxen anzeigen oder ausblenden kann. Durch die horizontale Anordnung wird vermieden, dass bei nur wenigen Kalendern unnötig viel Platz vergeudet wird, wie das in Lightning der Fall ist. Gerade im Büro, wo man besonders oft Kalender von mehreren Personen verwaltet, sollte genügend Breite am Bildschirm zur Verfügung stehen, um zumindest die wichtigsten dort anzeigen zu können. Hat man mehr Kalender als Platz zur Verfügung, lassen sich jene, die vorerst nicht sichtbar sind, durch einen Klick auf den Erweiterungspfeil einblenden (siehe Abbildung 23). Hier findet sich auch ein Link zur Kalenderverwaltung, in der man freigegebene Kalender modifizieren kann.

Ist der Platz sehr knapp bemessen, werden dementsprechend auch mehrere der Kalender ins Erweiterungsmenü verschoben, bis im Extremfall (siehe Abbildung 19) alle Einträge dort zu finden sind.

Hat man keine oder noch keine freigegebenen Kalender geöffnet (siehe Abbildung 20), findet sich an der selben Stelle statt der anderen Kalender eine Schaltfläche, die direkt das Fenster zum Hinzufügen freigegebener Kalender öffnet, sodass man sich einen unnötigen Zwischenschritt über ein anderes Fenster erspart.

Ganz rechts in der Menüleiste befinden sich eine Schaltfläche, um ins Einstellungsmenü zu gelangen und eine Schaltfläche, um sich auszuloggen. Die Einstellungen werden in einem späteren Kapitel noch genauer beleuchtet.

## 9.2 Terminfenster

Erstellt man einen neuen Termin über die Menüleiste oder öffnet man einen bestehenden Termin zum Bearbeiten, gelangt man immer in das Terminfenster (Abbildung 24). Die Felder im Terminfenster sind strikt nach ihrer Wichtigkeit sortiert. Zeit und Titel sind deshalb als die ersten Elemente angeführt und farblich vom Rest getrennt, weil das auch die einzigen Pflichtfelder sind, die außerdem wie in Lightning auch direkt im Kalender eingegeben und verändert werden können ohne dieses Terminfenster dafür öffnen zu müssen.

Neu hinzu kommt in diesem Fenster jedoch ein Feld für eine Abkürzung des Termintitels. Standardmäßig wird jeder Termin automatisch vom System durch Anfangsbuchstaben je nach vorhandenem Platz abgekürzt. Gibt man jedoch hier manuell eine Abkürzung ein, wird bei Platzproblemen auf diese zurückgegriffen, solange diese nicht immer noch zu lang für die verfügbare Breite ist.

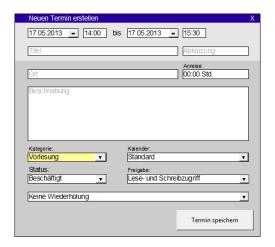


Abbildung 24: Low-Fidelity-Prototyp: Terminfenster

Das kann sinnvoll sein, weil es oft offizielle Abkürzungen gibt und automatische Abkürzungen mitunter schwierig zu interpretieren sind. So wird etwa eine Lehrveranstaltung mit dem Titel »Einführung in das Programmieren« von der Veranstaltungsleitung selbst als »EProg« abgekürzt, sodass der wesentliche Begriff des Programmierens noch erkennbar bleibt, während für einen automatischen Algorithmus in der einfachsten Form »EidP« naheliegender wäre.

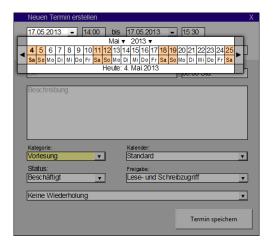


Abbildung 25: Low-Fidelity-Prototyp: Datumsauswahl in Datumsfeldern

Da die klassischen Kalenderformen in TaBaKa durchgehend überwunden werden sollen, sind auch die Datumsfelder hier anders als man das aus bisherigen Programmen kennt (siehe Abbildung 25). Die Schaltfläche, die normalerweise eine Tabelle zeigt und nach einem Klick einen kleinen Monatskalender öffnet, wird hier sinngemäß durch ein waagrechtes Balkensymbol ersetzt, das eine kleinere Version der Navigationsleiste aus dem Hauptfenster öffnet. Monat

und Jahr lassen sich hier auch durch Pulldown-Menüs mit der Maus verändern, um schneller zu einem anderen Datum wechseln zu können und im unteren Bereich findet sich die Angabe des aktuellen Datums, zu dem man auch mit einem einzigen Klick springen kann. Der Rest des Fensters wird während der Datumswahl ausgegraut, damit der Fokus auf das aktive Element leichter fällt.

Nach den essentiellen Termindaten folgen Ort und Beschreibung, weil das besonders oft genutzte Felder sind. Direkt neben dem Ort kann hier bei Bedarf eine Anreisezeit eingegeben werden, die im Kalender ebenfalls graphisch dargestellt und blockiert wird. In der Voreinstellung steht diese immer auf 00:00. Sollte sich im späteren Verlauf der Kalendernutzung irgendwann etwas ändern, was die Fahrzeit verändern könnte – etwa die Absage eines Vortermines – wird der Benutzer automatisch gefragt, ob er auch die Anreisezeit dieses Termins verändern möchte. Bestätigt er diese Frage mit »Ja«, wird wieder dieses Terminfenster geöffnet.

Unter dem Beschreibungsfeld folgen die Detaileinstellungen. Die Bedeutung dieser Einstellungen ist im Wesentlichen die gleiche wie in Lightning.

Wichtig ist, dass in der Kategorie auch ein Eintrag in der Dropdown-Liste vorhanden ist, mit dem man direkt eine neue Kategorie erstellen kann, indem man in einem neuen Fenster eine Farbe und einen Namen angibt. Damit kann man direkt dort eine neue Kategorie anlegen, wo man bemerkt, dass man diese benötigt und muss nicht wie in Lightning erst in das Einstellungsmenü wechseln. Damit die ausgewählte Kategorie auf einen Blick gut erkennbar ist, wird das Feld auch mit ihrer zugewiesenen Farbe hinterlegt.

Unter »Kalender« kann man auswählen, für welchen der freigegebene Kalender der Termin gilt, unter »Status« wird angegeben, ob man während dieses Termins beschäftigt oder verfügbar ist und unter »Freigabe« wird festgelegt, ob Benutzer, die auf den entsprechenden Kalender generell Zugriff haben, diesen Termin bearbeiten oder lesen können. Ist der Lesezugriff verwehrt, wird externen Benutzern nur eine blockierte Zeit im Kalender angezeigt.

Als letztes Feld findet sich im Terminfenster der Wiederholungsmodus. Hier hat man die Auswahl zwischen täglicher, wöchentlicher, monatlicher, jährlicher sowie benutzerdefinierter Wiederholung. Wie in Lightning öffnet sich bei der Auswahl einer benutzerdefinierten Wiederholung ein eigenes Fenster, in dem man diese Wiederholung genauer spezifizieren kann.

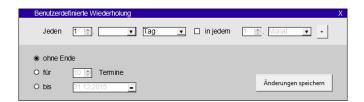


Abbildung 26: Low-Fidelity-Prototyp: Benutzerdefinierte Wiederholung: Vorgegebene Einstellung

Anders als in Lightning, wo diese Konfiguration ein recht komplexes und vielschichtiges Interface hat, lassen sich diese Einstellungen in TaBaKa durch ein einheitliches Interface regeln, in dem durch mehrere Felder ein Satz erstellt wird, der das gewünschte Intervall beschreibt. Im ersten Feld kann man dabei angeben

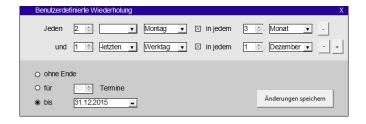


Abbildung 27: Low-Fidelity-Prototyp: Benutzerdefinierte Wiederholung: Anwendungsbeispiel

wie viele potentielle Tage jeweils übersprungen werden sollen. Das dritte Feld ist voreingestellt auf »Tag«, sodass beliebig viele Tage übersprungen werden können, kann aber auch auf »Werktag« geändert werden. Außerdem lässt sich dieses Feld noch auf einen spezifischen Wochentag umstellen, sodass aus der ursprünglich täglichen Wiederholung automatisch eine wöchentliche Wiederholung wird.

Zusätzlich kann man mittels Checkbox einen ähnlich aufgebauten Bereich dazuschalten, der ursprünglich auf »Monat« eingestellt ist, aber auf einen spezifischen Monat umgestellt werden kann, um so zu einer jährlichen Wiederholung zu führen. Ist dieser Zusatzbereich aktiviert, ändert sich auch die Bedeutung des ursprünglichen Bereiches so, dass ein zusätzliches Feld aktiv wird, mit dem man auch von rückwärts die Tage eines Monats ansprechen kann.

Durch Plus- und Minus-Schaltflächen am rechten Rand kann man zusätzliche Zeilen hinzufügen, um noch komplexere Wiederholzyklen zu erstellen.

Abbildung 27 zeigt ein konkretes Beispiel, in dem ein Termin alle drei Monate am zweiten Montag und zusätzlich am letzten Werktag des Jahres stattfindet. In solcher Komplexität ist dieses System auch Lightning überlegen, weil man dort strikt entscheiden muss, ob eine Wiederholung auf täglicher, wöchentlicher, monatlicher oder jährlicher Basis stattfindet, wodurch Mischformen ausgeschlossen sind.

Die Einstellungen zum Ende der Wiederholung sind dagegen exakt wie bei Lightning ausgestattet, da alle drei Optionen sinnvoll und einfach zu verstehen sind. Weil es sich beim Ende um eine fakultative Zusatzoption handelt, ist diese Funktionalität optisch von der obligatorischen Intervalldefinition abgegrenzt.

# 9.3 Kalenderverwaltung

Öffnet man über die Menüleiste die Kalenderverwaltung, werden dort in einer Tabelle alle Kalender angezeigt, die derzeit zum Anzeigen bereit stehen. Rechts von dieser Tabelle stehen vier Schaltflächen bereit, mit denen Einträge hinzugefügt, gelöscht oder in ihrer Reihenfolge verschoben werden können.

Das Verschieben kann dabei wichtig im Hinblick auf die Darstellung im Hauptfenster sein, weil dort nur die ersten Kalender dieser Liste unmittelbar angezeigt werden, wenn nicht für alle ausreichend Platz zur Verfügung steht.

Grundsätzlich wird jeder Kalender mit dem Namen angezeigt, mit dem er auch im System registriert ist. Markiert man einen Kalender in der Liste, wird der aktuelle Name aber auch in ein Textfeld unterhalb der Tabelle geladen, wo

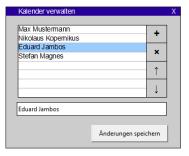




Abbildung 28: Low-Fidelity-Prototyp: Kalenderverwaltung: links die Übersicht verknüpfter Kalender, rechts das Menü zum Hinzufügen zusätzlicher Kalender

man ihn umschreiben kann. Dadurch könnte man zum Beispiel Namen abkürzen, um mehrere Kalender unmittelbar in der Hauptansicht unterbringen zu können.

Will man einen zusätzlichen Kalender öffnen, der freigegeben ist, öffnet sich dafür ein neues Fenster mit zwei Registerkarten.

Standardmäßig ist der Reiter »Gleiche Plattform« ausgewählt, in der sich eine alphabetisch sortierte Liste aller Kalender im gleichen System befindet, die einem Zugriffsrechte eingeräumt haben. Um auch in einer langen Liste den gewünschten Kalender rasch zu finden, kann man oberhalb der Tabelle in ein Suchfeld einen beliebigen Begriff eingeben und die Liste wird automatisch danach gefiltert.

Alternativ kann man zum Reiter »per Adresse« wechseln. Dort befindet sich ein einzelnes Textfeld, in das man den (Web-)Pfad eines öffentlichen Kalenders in einem bekannten Standard-Format eingeben kann.

# 9.4 Einstellungen

Das Einstellungsfenster (siehe Abbildung 29) verfügt wie in den meisten verbreiteten Anwendungen über mehrere Reiter für die einzelnen Unterkategorien. Damit sie ausreichend Platz haben, sind sie vertikal angeordnet.

Die einzelnen Unterpunkte verfügen zumeist nur über relativ wenige Elemente und folgen verbreiteten und schon vorgestellten Konzepten und Normen. Um nicht zu weit vom Kernthema dieser Arbeit abzuschweifen, werden die Einstellungsmöglichkeiten deshalb nachfolgend nur relativ kurz vorgestellt und vor allem dort detaillierter betrachtet, wo unmittelbar auf die Darstellung der Terminübersicht Einfluss genommen wird.

### • Kategorien:

Weil davon ausgegangen wird, dass sich jeder Benutzer nach Bedarf seine eigenen Terminkategorien anlegt, ist diese Registerkarte beim Öffnen der Einstellungen vorausgewählt.

Die Kategorieverwaltung gleicht in ihrem Aufbau der Kalenderverwaltung mit dem einzigen Unterschied, dass zusätzlich zu jedem Eintrag eine Farbe festgelegt werden kann. Klickt man die aktuelle Farbe im Bearbeitungsbereich an, öffnet sich ein Raster mit vorgegeben Kategoriefarben, aus denen

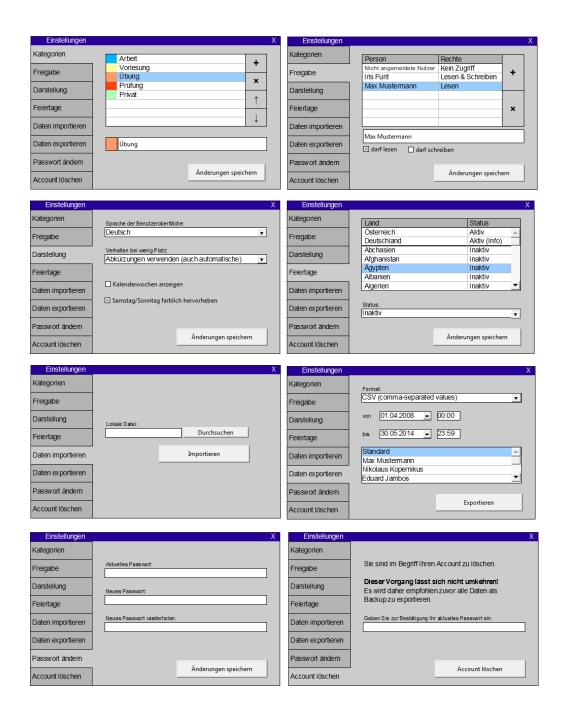


Abbildung 29: Low-Fidelity-Prototyp: Alle verfügbaren Reiter im Einstellungsfenster

man eine auswählen kann. Eine Farbe darf dabei auch für mehrere Kategorien vergeben werden, um zum Beispiel verschiedene private Aktivitäten unter einer Farbe zusammengefasst von Beruflichem zu unterscheiden.

Beim Anlegen einer neuen Kategorie wird hier anders als in der Kalenderverwaltung kein neues Fenster geöffnet, sondern stattdessen sofort ein neuer Eintrag angelegt und der Cursor in das Textfeld gesetzt, um eine Bezeichnung einzugeben.

# • Freigabe:

Jedem Benutzer im selben System können Lese- und Schreibrechte zuge- ordnet werden. Auch hier folgt der Aufbau dem bereits bekannten Tabellenschema. Von Beginn an besteht ein nicht löschbarer Eintrag ≫Nicht angemeldete Nutzer≪, der den Zugriff für Besucher der Plattform regelt, die sich nicht eingeloggt haben. In diesem Eintrag wird in der Standardeinstellung aus Gründen des Datenschutzes kein Zugriff gewährt. Das Ändern dieser Vorgabe kann beispielsweise sinnvoll sein, wenn man − etwa im Rahmen einer Vereinstätigkeit − einen eigenen Kalender für öffentliche Veranstaltungen führt.

## • Darstellung:

Hier können ein paar Grundeinstellungen der Kalenderdarstellung verändert werden. Die Sprache lässt sich umstellen, die Nummer der jeweiligen Kalenderwochen ein- oder ausblenden und auch die farbliche Hervorhebung von Wochenenden lässt sich ausschalten, falls Wochenenden für die eigene Tätigkeit aus irgendeinem Grund nicht relevant sind.

Auch kann hier festgelegt werden, wie sich der Kalender verhalten soll, wenn nur wenig Platz zur Verfügung steht, um Termine darzustellen. Man kann hier auswählen, dass sich der Kalender wie klassische Varianten verhält und zu langen Text einfach abschneidet, dass nur selbst festgelegte Abkürzungen genutzt werden sollen oder dass auch automatisch generierte Abkürzungen eingesetzt werden. In der Voreinstellung sind auch automatische Abkürzungen erlaubt, damit Termintitel, die mit den gleichen Worten beginnen, unterschieden werden können.

### • Feiertage:

Hier kann nach Ländern festgelegt werden, welche gesetzlichen Feiertage im System angezeigt werden. Dazu sind in einer Tabelle alle Ländernamen gelistet. Wählt man eines aus, kann man den entsprechenden Status auf »Inaktiv«, »Aktiv« oder »Aktiv (Info)« setzen. »Inaktiv« bedeutet, dass die Feiertage nicht angezeigt werden. »Aktiv« bedeutet, dass die Feiertage angezeigt werden und als Kategoriefarbe den gleichen Farbton haben, mit dem sonst Wochenenden hinterlegt sind und der nicht zur freien Vergabe für Kategoriefarben zur Verfügung steht. »Aktiv (Info)« bedeutet, dass der Feiertag zwar angezeigt wird, aber nur mit der Standard-Terminfarbe (Weiß) hinterlegt ist. Das ist beispielsweise sinnvoll, wenn man Geschäftsbeziehungen zu anderen Ländern unterhält: In einem solchen Fall sollte man die entsprechenden Feiertage kennen, aber nicht mit den eigenen verwechseln.

Zur leichteren Übersicht werden alle aktiven Länder am Beginn der Liste separat angeführt, sodass man diese nicht zwischen unzähligen anderen Einträgen suchen muss.

### • Daten importieren:

Der Datenimport erfolgt durch ein einfaches Auswahlfeld, mit dem die entsprechende Importdatei ausgewählt wird. Kann die Datei nicht gelesen oder verarbeitet werden, gibt das System eine entsprechende Fehlermeldung aus.

## • Daten exportieren:

Hier lassen sich Format, Zeitraum und Kalender für einen gewünschten Export auswählen und dieser durchführen. Als Zeitraum ist immer jener Zeitraum vorausgewählt, in dem sich tatsächlich Daten befinden, sodass man im Zweifelsfall den gesamten Datenbestand zur Verfügung hat.

### • Passwort ändern:

Die Änderung des Zugangspasswortes erfolgt wie im Web üblich durch Eingabe des alten und neuen sowie zur Absicherung auch der Wiederholung des neuen Passwortes.

#### • Account löschen:

Das Löschen des eigenen Accounts erfolgt nur nach Eingabe des Passwortes und nochmaliger Bestätigung, dass man diesen Vorgang durchführen möchte. Dieses Fenster ist bewusst textlastig, um auf die Auswirkungen hinzuweisen und zu verhindern, dass jemand glaubt sich auf einem anderen Reiter zu befinden und dann vielleicht aus Unachtsamkeit eine Löschung durchführt.

# 10 Cognitive Walkthrough

In diesem Kapitel wird der Low-Fidelity-Prototyp anhand konkreter Aufgabenstellungen zu definierten Anwendungsfällen aus dem Kapitel »Ziele« evaluiert. Dabei wird durchgedacht, durch welche Schritte die jeweilige Aufgabe im Referenzkalender Lightning sowie in TaBaKa zu lösen ist, um den Prototypen direkt gegen einen klassischen Kalender auf seine Tauglichkeit zu testen.

Um diese Arbeit in einem vernünftigen Rahmen zu halten, wird dabei vor allem auf jene Anwendungsfälle eingegangen, die unmittelbar mit der Kalenderansicht zusammenhängen. Anwendungsfälle, die in der ursprünglichen Liste weiter hinten gelistet wurden und nicht primär mit der Auflösung der Kalenderblattmetapher zusammenhängen, werden hier nicht weiter beachtet.

Bei der Analyse von TaBaKa wird grundsätzlich davon ausgegangen, dass dieser Kalender sich in seiner Ausgangsposition befindet, wie sie in Abbildung 18 dargestellt wird – das heißt, dass ein Zeitraum von sieben Tagen selektiert ist, der aktuelle Tag am linken Rand liegt und die Navigationsleiste auch sieben Tage vor dem markierten Bereich anzeigt. Sowohl TaBaKa als auch Lightning in der Wochen- und Tagesansicht zeigen standardmäßig das typische Büro-Zeitintervall von 08:00 bis 17:00 Uhr an.

Da die Darstellung vor allem in klassischen Ansichten je nach Datum sehr unterschiedlich sein kann, wird in allen Aufgabestellungen davon ausgegangen, dass aktuell Donnerstag, der 30. Mai 2013, ist.

# 10.1 Zuordnung von Wochen- und Monatstagen ablesen

Aufgabenstellung: Man will das aktuelle Datum wissen.

- Lightning: Nur in der Tagesansicht wird das aktuelle Datum an einer einheitlichen Stelle direkt angeschrieben. In allen anderen Ansichten muss man erst nach der Markierung des aktuellen Tages suchen und dann in der entsprechenden Spalte den Wochentag ablesen.
- TaBaKa: Der aktuelle Tag lässt sich ganz links ablesen.
- Fazit: In unserer Kultur, in der man von links nach rechts liest, hat man in TaBaKa das aktuelle Datum immer sofort im Blick, was in Lightning nur in der Tagesansicht möglich ist.

Aufgabenstellung: Man will nachsehen, welches Datum der kommende Dienstag trägt.

- Lightning: In der Ansicht »Mehrere Wochen« wechselt man einfach mit dem Blick in die zweite Tabellenzeile und die Spalte für Dienstag. Dort steht die gesuchte Zahl 4. Fällt einem aber spontan nicht ein, welcher Monat dem Mai folgt, muss man mit dem Blick zurück ans Ende der ersten Tabellenzeile, weil nur dort beim Monatsersten auch der Monat selbst angeschrieben ist.
  - In allen anderen Ansichten lässt sich das Datum nicht direkt aus der Großansicht ablesen, weil man erst über die Schaltflächen links oben weiterblättern muss. Alternativ zum Weiterblättern kann man in diesem Fall
    aber auch mit dem Blick zum kleinen Navigationskalender wechseln, der
    sich ebenfalls links oben befindet und in diesem konkreten Beispiel auch
    die nächste Woche noch umfasst (– andernfalls müsste man auch hier erst
    weiterblättern).
- TaBaKa: Man geht die Wochentage entweder in der Großansicht oder im grauen Bereich der Navigationsleiste von links nach rechts bis zum Dienstag durch, direkt darüber stehen Monatstag, Monatsname und Jahr.
- Fazit: In TaBaKa muss bei der Lösung dieses Problems nicht zwischen unterschiedlichen Ansätzen unterschieden werden. Da es keine Tabelle wie in einer Monatsansicht gibt, muss man auch nicht in einem zweidimensionalen Raum nach dem Datum suchen, sondern nur entlang einer linearen Auflistung. Durch die fortlaufende Darstellung erspart man sich außerdem das Blättern.

Aufgabenstellung: Man will nachsehen, welcher konkrete Tag der Mittwoch vor zwei Wochen war.

• Lightning: In der Monatsansicht oder dem kleinen Navigationskalender schaut man ausgehend vom aktuellen Datum zwei Zeilen weiter hinauf

und eine Spalte weiter nach links, um den entsprechenden Monatstag zu finden. In allen anderen Ansichten müsste man zurückblättern oder – im Fall von »Mehrere Wochen« zurückscrollen.

- TaBaKa: Man schiebt die Navigationsleiste ein wenig nach rechts, so dass der entsprechende Zeitbereich sichtbar wird, zählt vom aktuellen Tag die Wochen zurück und sucht in der entsprechenden Woche den Mittwoch. Der gesuchte Monatstag steht direkt darüber.
- Fazit: In diesem konkreten Beispiel ist Lightning von Vorteil, weil man hier die Lösung direkt ablesen kann ohne eingreifen zu müssen. Dieser Vorteil ist allerdings nicht permanent und kann in anderen Situationen wegfallen. Wäre der aktuelle Tag etwa der erste Mai statt des dreißigsten, könnte man in der Monatsansicht nicht einmal mehr die Vorwoche sehen ohne auch in Lightning zurückblättern zu müssen. TaBaKa ist durch seine fortlaufende Darstellung einheitlicher in der Bedienung.

Aufgabenstellung: Man will wissen, welcher Wochentag der 11. September 2001 war.

- Lightning: Im kleinen Navigationskalender klickt man auf die Jahreszahl. Es öffnet sich eine Liste mit Jahreszahlen, die bis 2009 zurückgehen. Man hält die Maus kurz über einen Pfeil, um weiter nach oben zu scrollen, bis das Jahr 2001 sichtbar wird und klickt dieses an. Danach klickt man auf den Monatsnamen. Es öffnet sich ein Pulldown-Menü, aus dem man den September auswählt. Am Ende sucht man aus dem Navigationskalender die Zahl 11 heraus und liest an der Spaltenbeschriftung den Wochentag ab.
- TaBaKa: In der Detailansicht klickt man auf die Jahreszahl. Daraufhin wird diese markiert und man tippt 2001 ein. Danach klickt man auf den Monatsnamen von Juni (– weil der 30. Mai ganz links liegt, gehört schließlich der größte Bereich des Kalenders schon zum Juni –) und wählt dort im Pulldown-Menü den September aus. In der Navigationsleiste sucht man nach der 11 und liest direkt darunter den Wochentag ab.
- Fazit: Die Vorgehensweise ist in beiden Kalendern grundsätzlich sehr ähnlich, wenn man die entsprechenden Funktionalitäten kennt. Lightning hat in dieser Hinsicht den Nachteil, dass es anders als durch die Symbole in Ta-BaKa keinen Hinweis darauf gibt, dass man Jahreszahl und Monat anklicken kann erst wenn man die Maus über die entsprechende Stelle hält, wird diese als Schaltfläche dargestellt.

Etwas schwierig gestaltet sich auch das Scrollen durch die Jahreszahlen, da man die Scrollgeschwindigkeit bei Aktivierung der Scrollschaltfläche nicht beeinflussen kann. Grundsätzlich lässt sich zwar auch mit dem Mausrad scrollen, aber da man bei der Bewegung des Zeigers fast zwangsläufig über das Scrollfeld muss, das auch sofort bei Mauskontakt zu scrollen beginnt, ist es naheliegender dieses zu benutzen. TaBaKa geht stattdessen den Weg die Jahreszahl direkt per Tastatur einzugeben. Das klingt im ersten Moment vielleicht etwas behäbiger, umgeht aber die genannten Probleme des Scrollens und ermöglicht es auch weit entfernt liegende Jahre unmittelbar anzusteuern.

Als etwas nachteilig könnte man in TaBaKa die Orientierungsmöglichkeit auffassen, wenn ein Termin durch eine zeitliche Entfernung in mehreren Wochen angegeben ist. In einem Wochenkalender kann man die entsprechende Anzahl an Seiten vor- oder zurückblättern, um die entsprechende Woche zu erreichen und in »Mehrere Wochen« kann man die Anzahl an den Zeilen abzählen, in TaBaKa sind die Wochen dagegen aufgrund der fortlaufenden Darstellung nicht so klar abgegrenzt, wodurch man sich leichter verzählen kann. Da in der Praxis solche Angaben aber selten über drei bis vier Wochen hinausgehen sollten, dürfte sich dieser Nachteil in Grenzen halten.

# 10.2 Terminübersicht anzeigen

Aufgabenstellung: Man will eruieren, an welchem der kommenden fünf Tage man noch am meisten unverplante Zeit hat.

- Lightning: In der Wochen- und der Monatsansicht muss man eine Seite weiter blättern, um drei der fünf relevanten Tage zu sehen, in der Tagesansicht müsste man sich einzeln durch alle Tage blättern und in »Mehrere Wochen« muss man an den Beginn der nächsten Tabellenzeile wechseln.
- TaBaKa: Die fünf relevanten Tage liegen direkt nebeneinander und sind damit auf einen Blick vergleichbar.
- Fazit: Die fortlaufende Darstellung in TaBaKa ist hier klar von Vorteil.

Aufgabenstellung: Man trägt sich unter einer eigenen Kategorie immer Zeiten ein, die man im Büro verbringt. Für gewöhnlich sind diese Zeiten jeden Werktag von 08:00 bis 13:00 Uhr eingetragen, manchmal gibt es aber Abweichungen. Man will alle Abweichungen im Mai 2013 finden.

- Lightning: Man blättert einzeln durch alle Mai-Blätter der Wochenansicht und merkt oder notiert sich die Ausreißer, die durch ihre veränderte Position gleich auffallen. Alle anderen Ansichten sind für diese Aufgabe schlecht geeignet, weil man in der Tagesansicht durch alle 31 Tage einzeln blättern müsste und in der Monatsansicht sowie unter »Mehrere Wochen« die Endzeit eines Termins nicht direkt angezeigt wird.
- TaBaKa: Man schiebt die Navigationsleiste so zurecht, dass der ganze Mai im Blickfeld ist. Dann verschiebt man die linke Grenze des grauen Bereichs an den Monatsersten und die rechte Grenze an den Monatsletzten. Alle Ausreißer fallen jetzt wie im Wochenkalender durch ihre Position direkt auf.
- Fazit: Das Einrichten der Ansicht ist in TaBaKa etwas umständlicher als sich durch die Wochenblätter zu klicken, führt allerdings zu dem großen Vorteil, dass man sich keine Informationen aus anderen Blättern merken oder notieren muss.

Aufgabenstellung: Es ist 12:30 Uhr. Am Vormittag hatte man drei Termine und um 20:00 Uhr hat man noch einen Auswärtstermin. Man will im Kalender nachsehen, wann man seinen nächsten Termin hat und wann man zu diesem aufbrechen muss.

• Lightning: In der Monatsansicht oder unter »Mehrere Wochen« muss man erst die Startzeiten aller Termine lesen, weil es keinen grafischen Indikator dafür gibt, welche davon noch offen sind. Die aktuelle Zeit muss man gegebenenfalls von der Systemuhr oder einem anderen Gerät ablesen, um eruieren zu können, welcher Termin der nächste ist.

Alternativ kann man die Tages- oder Wochenansicht nutzen, in der die aktuelle Zeit markiert wird. Allerdings muss man hier erst weiter hinunter scrollen ohne vorab zu wissen, ob sich dort etwas befindet, weil der Termin außerhalb des angezeigten Intervalls auf der Zeitachse liegt.

Um die Abfahrtszeit zu eruieren, muss man die Maus kurz über den Termin halten oder das Terminfenster öffnen, sodass man den eingetragenen Ort sieht. Wenn die Dauer der Anreise zu diesem Ort nicht bekannt ist, muss man diese anschließend noch recherchieren.

- TaBaKa: Die aktuelle Uhrzeit ist markiert und ein Pfeil am unteren Ende des Kalenders zeigt an, dass außerhalb des angezeigten Bereichs noch ein Termin folgt. Man scrollt entsprechend weit nach unten, bis dieser angezeigt wird. Hatte man beim Eintragen des Termins schon eine Fahrzeit angegeben, ist der Abfahrtszeitpunkt schon eingezeichnet und kann direkt abgelesen werden. Andernfalls kann man direkt nach dem Termintitel den Ort ablesen und die Fahrzeit gegebenenfalls recherchieren.
- Fazit: TaBaKa liefert hier zusätzliche Informationen, welche die Lösung der Aufgabe erleichtern.

Aufgabenstellung: In sechs Tagen ist ein Termin (13:00 bis 16:00 Uhr) mit angegebenem Ort eingetragen. Man will ablesen, von wann bis wann dieser dauert.

• Lightning: Man hält den Mauszeiger über den Termin, sodass ein Tooltip erscheint. Mit dem Blick geht man in diesem Tooltip hinunter bis in die dritte Zeile, wo Start- und Endzeit angeschrieben sind. Hier ist zuerst noch einmal das Datum ablesbar und danach am Zeilenende stehen die konkreten Uhrzeiten.

Alternativ kann man in der Tages- oder Wochenansicht weiterblättern und dann anhand der Hilfslinien die Zeit an der vertikalen Achse ablesen, was in der Wochenansicht aber relativ mühsam ist, weil man die Hilfslinie erst bis zum linken Rand zurückverfolgen muss.

- TaBaKa: Die Uhrzeiten sind direkt an den Termingrenzen angeschrieben und können direkt abgelesen werden. Sollte der Kalender so voll sein, dass die Zeiten aus Platzgründen nicht angeschrieben sind, kann man sich auch hier einen Tooltip anzeigen lassen oder mit einem Klick in der Navigationsleiste diesen Tag ganz nach links schieben und dort die Zeit an der Achse ablesen.
- Fazit: Die zusätzlich angezeigten Informationen in TaBaKa erleichtern das Ablesen der wesentlichen Informationen und die Möglichkeit jeden beliebigen Tag ganz nach links verschieben zu können lässt einen auch in vollen Kalendern jede relevante Zeit schnell ablesen.

# 10.3 Termin eintragen

Aufgabenstellung: Man möchte einen Termin der Kategorie »Besorgung« mit Titel, Ort und längerer textueller Beschreibung anlegen. Für andere Personen mit Zugriffsrechten soll dieser Termin nicht lesbar sein. Die Kategorie »Besorgung« wurde bisher noch nie benutzt und muss neu hinzugefügt werden.

• Lightning: Im ersten Schritt muss die neue Kategorie angelegt werden. Man drückt die Taste »Alt«, damit die Menüleiste eingeblendet wird, die standardmäßig ausgeblendet ist. Von dort aus geht man in das Menü Extras/Einstellungen, dort auf »Kalender« und schließlich auf das Registerblatt »Kategorien«. Dort klickt man auf die Schaltfläche »Hinzufügen...«. Es öffnet sich ein neues Fenster, in das man den Namen der neuen Kategorie eingibt und mit »OK« bestätigt. Danach schließt man das Einstellungsfenster wieder.

Einen neuen Termin mit Details legt man durch Klick auf eine persistente Schaltfläche an, die links oben zwischen weiteren Schaltflächen liegt, oder man macht einfach einen Doppelklick auf die Kalenderfläche. Dadurch öffnet sich das Fenster für die Termindetails.

Das Feld für den Titel befindet sich ganz oben und ist schon selektiert, sodass man diesen gleich eintippen kann. Unmittelbar darunter kann man den Ort eingeben, darunter aus einem Pulldown-Feld die neu angelegte Kategorie auswählen und darunter wiederum die Start- und Endzeit eingeben. Dann folgen zwei Felder, die man überspringt, um zum Feld zu gelangen, in das man den Beschreibungstext einträgt.

Anschließend muss man zurück an den Anfang des Fensters, wo sich in der Symbolleiste als dritter Eintrag der Punkt »Privatsphäre« befindet. Hier wählt man aus einem Pulldown-Menü »Privater Termin« und speichert den Termin schließlich über die entsprechende Schaltfläche ganz links im selben Menü.

 TaBaKa: Einen neuen Termin mit Details legt man durch Klick auf die erste Schaltfläche links oben im Menü oder einen Doppelklick auf die Kalenderfläche an. Dadurch öffnet sich das Fenster für die Termindetails.

Das Feld für das Start-Datum befindet sich ganz oben und ist schon markiert, sodass man dieses gleich eingeben kann. Daneben befinden sich Felder für die genaue Uhrzeit und das Terminende. Darunter folgen der Reihe nach die Felder für Titel, Ort, Beschreibung und Kategorie.

Unter »Kategorie« wählt man ganz unten in der Dropdown-Liste den abgesetzt dargestellten Eintrag »Neue Kategorie«. Es öffnet sich ein neues Fenster, in das man die Bezeichnung eingibt und mit »OK« bestätigt.

Rechts unterhalb des Kategorie-Feldes befindet sich das Freigabe-Feld. In diesem stellt man auf »Kein Zugriff« um. Das letzte Feld im Fenster überspringt man und speichert den Termin über die Schaltfläche am Fensterende.

• Fazit: Lightning betrachtet die Kategorien offenbar als eine voreingestellte Konfiguration, die von Benutzern nur selten verändert wird. Dementsprechend umständlich ist es auch, dort eine neue Kategorie anzulegen.

TaBaKa ermöglichst es dagegen neue Kategorien auch ganz spontan anzulegen, wenn man beim Erstellen eines neuen Termins bemerkt, dass keine der bestehenden geeignet ist.

Im Ablauf der Dateneingabe ist TaBaKa konsequenter als Lightning, weil man alle Informationen hier von oben nach unten eintragen kann und dann auch unten den Vorgang abschließt, statt wieder zurück hinauf zu müssen.

# 11 High-Fidelity-Prototyp

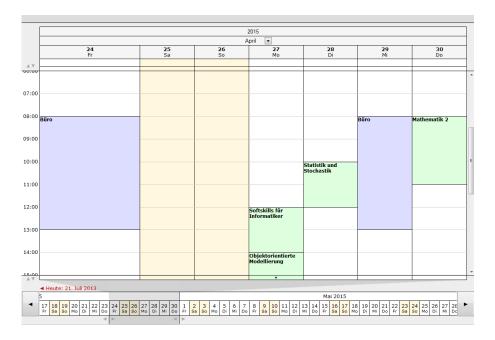


Abbildung 30: High-Fidelity-Prototyp: 7 Tage im Fokus

Nachdem der Cognitive Walkthrough aufgezeigt hatte, dass das Konzept von Ta-BaKa grundsätzlich funktionsfähig ist, wurde basierend auf dem Low-Fidelity-Prototypen ein High-Fidelity-Prototyp in HTML und JavaScript umgesetzt. Da es sich nur um einen Prototypen handelt, wurde Browserkompatibilität dabei vorerst nicht berücksichtigt und ausschließlich mit der aktuellen Version von Mozilla Firefox getestet.

Implementiert wurden alle Möglichkeiten zur Navigation, die Anpassung der angezeigten Zeitintervalle (Tage und Uhrzeiten) und der Hinweis auf Termine, die außerhalb des aktuell angezeigten Zeitintervalls liegen. Weil das Artefakt auf einem anderen Rechner entwickelt wurde als später für den Usability-Test in Verwendung war, wurde auch darauf geachtet, dass sich der Kalender an unterschiedliche Bildschirmgrößen anpasst.

Nicht implementiert wurden dagegen die Menüleiste inklusive all ihrer Funktionalitäten, das Erstellen und Bearbeiten von Terminen, die Anzeige mehrtägiger Termine, die Anzeige von Start-und Endzeiten sowie die Darstellung von Fahrzeiten und die Abkürzung von Termintiteln bei zu wenig Platz.

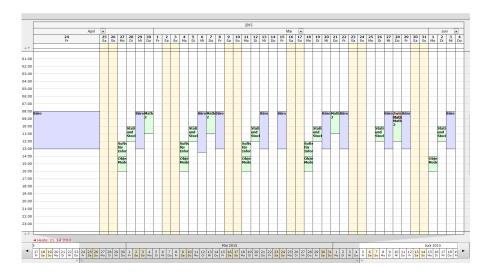


Abbildung 31: High-Fidelity-Prototyp: Breitbild, 42 Tage im Fokus, 00:00 bis 24:00 Uhr

Insgesamt wird durch diesen Prototypen also vorwiegend das Navigationskonzept von TaBaKa abgebildet.

Gegenüber dem ursprünglichen Low-Fidelity-Prototypen wurden bewusst ein paar Änderungen vorgenommen:

- Die Darstellung der Navigationsleiste wurde stärker an die Darstellung der Detailansicht angepasst, damit das Mapping leichter fällt. Die Trennlinie zwischen Tag und Wochentag wurde entfernt, die Hinterlegung von Wochenenden hat nun den gleichen Farbton und der aktuelle Tag wird wie in der Detailansicht grau hinterlegt statt in höherem Schriftgewicht geschrieben.
- Im Low-Fidelity-Prototypen befand sich ein grauer Bereich unterhalb der Navigationsleiste, der eigentlich einen Schatten darstellen sollte, um die Leiste stärker als anfassbares und manipulierbares Objekt hervorzuheben. Allerdings erzeugte dieser nicht die erhoffte grafische Wirkung und wirkte im Zusammenspiel mit der ebenfalls grauen Kennzeichnung des selektierten Zeitraumes eher irritierend, weshalb er im High-Fidelity-Prototypen entfernt wurde.
- Die Zeitleiste wurde übersichtlicher gemacht, indem nur noch die vollen Stunden angeschrieben werden und sich die Beschriftungen unmittelbar neben den zugehörigen Hilfslinien befinden. Detailliertere Beschriftungen (und Hilfslinien) könnten zwar sinnvoll sein, wenn sehr kleine Zeitintervalle angezeigt werden da so eine Ansicht in der Praxis aber nur selten genutzt wird, wurde das für diesen Prototypen nicht berücksichtigt.
- Zusätzlich zur Navigation über die Navigationsleiste kann man auch einen anderen Tag in der Detailansicht in den Fokus rücken, indem man dessen Tabellenkopf anklickt.

Der Prototyp macht außerdem Verwendung von positionsabhängigen Mauszeigern, um mögliche Interaktionen anzuzeigen. Verschiebbare Elemente werden durch einen Doppelpfeil in der der entsprechenden Lage angezeigt, anklickbare Elemente durch das Standard-Link-Symbol in Form einer Hand. Die Navigationsleiste, die man sowohl verschieben als auch anklicken kann, ist dabei in zwei Bereiche unterteilt. Im Bereich der Monatsüberschrift wird ein horizontaler Doppelpfeil angezeigt, weil man in diesem Bereich nur schieben kann. Hält man die Maus dagegen über einen konkreten Tag, den man auch anklicken kann, wird der Zeiger zu einer Hand, sodass beide Navigationsmöglichkeiten angedeutet werden.

Aus Zeitgründen wurde auf das ursprünglich geplante Punktesymbol bei der Jahreszahl verzichtet, das verdeutlichen soll, dass diese sich ändern lässt. Ebenso sieht das Pfeilsymbol neben den Monaten anders als im ursprünglichen Plan aus, weil hier zur Einfachheit auf ein Standard-Dropdown-Feld zurückgegriffen wurde.

Die Anzeige mehrtägiger Termine ist noch nicht implementiert. Außerdem sind immer mindestens zwei Tage im Fokus, weil es bei nur einem Tag zu einer Fehldarstellung kommen würde, deren Ursache ich noch nicht erhoben habe. Bei der Anpassung des Uhrzeit-Intervalls gibt es derzeit noch keine Blockade, wenn man eine Grenze über eine andere zieht und der Tooltip beim Hinweis auf Termine außerhalb des aktuellen Uhrzeitintervalls gibt vorerst nur einen allgemeinen Hinweis auf auswärtige Termine statt der vorgesehenen konkreten Daten.

# 12 Usability-Test

Basierend auf dem nun bestehenden High-Fidelity-Prototypen wird im nächsten Schritt ein Usability-Test durchgeführt.

# 12.1 Ziele

Ziel des Usability-Tests ist es die Praxistauglichkeit sowie die Akzeptanz des neuen Konzepts von TaBaKa an realen Benutzern zu überprüfen, nachdem diese bisher auf einer ausschließlich theoretischen Ebene analysiert wurden.

Es soll festgestellt werden, ob die der Entwicklung zugrundeliegenden Annahmen zur Kalenderbenutzung korrekt waren und ob durch das neue Artefakt eine entsprechende Verbesserung gegenüber klassischen Kalenderdarstellungen erreicht wurde, die sich durch Geschwindigkeitsvorteile oder Benutzerzufriedenheit nachweisen lässt. Nebenbei wird auch darauf geachtet, ob in der Handhabung spezifische Probleme auftreten, an die im bisherigen Entwicklungsprozess nicht gedacht wurde.

Da im Prototypen bisher nur die Navigation implementiert ist und diese unterschiedliche Interaktionsmöglichkeiten – wie Ziehen der Navigationsleiste oder Klick auf eine Schaltfläche – bereitstellt, soll vor allem auch darauf geachtet werden, welche dieser Möglichkeiten von den Testpersonen wahrgenommen und bevorzugt genutzt werden.

Aus diesen gesammelten Erkenntnissen soll abgeleitet werden, welche Änderungen und Verbesserungen vor beziehungsweise bei der Weiterentwicklung des

Systems vorgenommen werden sollten und auf welche Navigationskonzepte besonders geachtet werden müsste.

### 12.2 Methode

Da vor allem der Vergleich mit klassischen Kalendersystemen interessant ist, wird TaBaKa im Rahmen dieses Tests nicht isoliert betrachtet, sondern es sind ähnliche Aufgaben sowohl in TaBaKa als auch in dem Referenzkalender Lightning zu lösen, um direkte Vergleichbarkeit zu gewährleisten.

Weil der Prototyp in seinem Funktionsumfang auf Navigationsmöglichkeiten beschränkt ist, sind alle Aufgaben für beide Kalender so formuliert, dass die Ergebnisse auf Papier zu notieren sind.

Aufgrund der funktionellen Beschränkung, die nur wenige komplexe Aufgabenstellungen ermöglicht, sind außerdem in zwei von insgesamt drei Aufgaben mehrere ähnliche Unteraufgaben unter einem Punkt zusammengefasst. Das ist vor allem hilfreich, um gemessene Zeiten für die Absolvierung gegenüberzustellen, weil es sonst zu Zeitspannen im Sekundenbereich kommen könnte, die es erschweren würden neben der Zeiterfassung auch noch Notizen zur eigentlichen Vorgehensweise zu machen.

Insgesamt teilt sich der Test in folgende Abschnitte:

- 1. Erfassung demographischer Daten
- 2. Erfassung von Vorkenntnissen im Umgang mit Kalendern
- 3. Test eines der zwei Kalendersysteme
- 4. Test des zweiten Kalendersystems
- 5. Vergleich der beiden getesteten Systeme

An demographischen Daten werden Geschlecht, Alter, Berufsstand, Beruf beziehungsweise Schul- oder Studienrichtung abgefragt.

In den Vorkenntnissen wird eruiert, ob die Testpersonen Kalendersoftware oder Kalender auf Papier nutzen, auf welchen Geräten sie Kalendersoftware einsetzen, welche spezifischen Software-Produkte ihnen bekannt sind und welche Ansicht (Tag, Woche oder Monat) sie in einem klassischen Kalender am Häufigsten einsetzen.

Bei der Frage nach der Bekanntheit verschiedener Kalendersysteme sind bekannte Namen wie Outlook oder Lightning vorgegeben und sollen nach den Stati »unbekannt«, »namentlich bekannt«, »schon benutzt« und »regelmäßige Nutzung« bewertet werden. Besonders interessant ist hierbei natürlich die Fragestellung, ob der anschließend getestete Referenzkalender Lightning oder die frühere Standalone-Version Sunbird schon bekannt sind.

Unter den vorgegebenen Kalendern findet sich hier aber auch TaBaKa, obwohl diesen Kalender aufgrund seiner Neuheit natürlich noch niemand kennen kann. Die Nennung soll in diesem Fall nur als psychologischer Trick dienen, damit TaBaKa in den anschließenden Analysen nicht von vornherein als unbekannte Nebenerscheinung abgetan wird, die sich nicht mit den etablierten Systemen vergleichen lässt. Es soll durch die Nennung also das Gefühl gestärkt werden, dass TaBaKa und Lightning zumindest konkurrenzfähig und damit auch sinnvoll vergleichbar sind.

Ein solcher Vergleich soll nämlich am Ende des Tests angestellt werden. Hier sind eine Reihe von Eigenschaften wie Alltagstauglichkeit, Verständlichkeit und Übersichtlichkeit angegeben und die Testpersonen sollen zu jeder dieser Eigenschaften angeben, in welchem getesteten Kalender diese besser umgesetzt ist beziehungsweise ob beide Kalender gleichauf sind.

Der Pilottest und die darauf basierenden Änderungen fanden am neunten Juli 2013 statt. Alle weiteren Tests wurden danach bis zum 25. Juli durchgeführt.

Da nicht alle Personen am gleichen Tag getestet werden konnten, das aktuelle Datum aber die anfängliche Darstellung beider Kalender unmittelbar nach deren Start beeinflusst, wird in der Angabe ein konkretes Datum als das heutige vorgegeben und auch die Systemzeit am Testrechner ist entsprechend vorkonfiguriert.

# 12.2.1 Testpersonen

Gemäß der Zielgruppendefinition am Beginn dieser Arbeit wurden gezielt junge Menschen getestet, von denen eine gewisse Offenheit gegenüber etwas so Etabliertem wie Kalendersoftware erwartet werden kann. Der Umfang umfasste letztendlich sechs reguläre Testpersonen zuzüglich einer Person für den Pilottest.

Die Rekrutierung erfolgte in meinem eigenen Bekannten- und Verwandtenkreis und umfasste Personen im Alter von 18 bis 25 Jahren, die sich derzeit in einer Ausbildung im Informatik- oder Medienbereich befinden und daher neuen Lösungen gegenüber grundsätzlich aufgeschlossen sein sollten. Eine Person gab kein Alter an, liegt aber grob geschätzt auch im angegebenen Altersbereich.

Die Testperson des Pilottests und vier weitere Personen waren männlich, zwei weiblich.

Sechs der getesteten Personen waren Studenten und einer ein Schüler. Von den Studenten waren vier außerdem unselbständig berufstätig.

Fünf der Testpersonen gaben an Kalendersoftware zu nutzen, die verbleibenden beiden Personen nutzten nur Papierkalender. Die Testperson im Pilottest war die einzige, die angab beides zu nutzen.

Folgende Tabelle gibt an wie viele der Testpersonen Kalendersoftware auf den angeführten Geräten nutzen:

Gerät	nie	manchmal	oft	
PC/Mac	1	0	4	
Laptop	0	2	3	
Tablet	4	1	0	
Smartphone	0	0	5	

Die Bekanntheit konkreter Kalendersysteme verteilt sich folgendermaßen:

Kalender	unbekannt	namentlich bekannt	schon benutzt	regelmäßige Nutzung	
Microsoft Outlook	0	2	4	1	
Apple iCal/Kalender	1	5	1	0	
Mozilla Thunderbird + Lightning	1	4	2	0	
Mozilla Sunbird	4	3	0	0	
Evolution	6	1	0	0	
Kontact	7	0	0	0	

Von den fünf Personen, die Kalendersoftware verwenden, gaben vier an, vorwiegend die Wochenansicht zu nutzen. Die fünfte bevorzugte die Monatsansicht.

Da ich persönlich bekannte Personen getestet habe, hatte ich im Vorfeld mit niemandem viel über das konkrete Thema meiner Bachelorarbeit gesprochen, um Verzerrungen der Endbewertungen aufgrund der persönlichen Beziehung zu vermeiden. Ich hatte grundsätzlich immer nur erwähnt, dass das Thema meiner Arbeit »Kalenderdarstellungen« sind, aber nicht, dass ich selbst einen der beiden Testkalender entworfen und implementiert hatte.

Ganz verhindern konnte ich dadurch aber auch nicht, dass Testpersonen zumindest Theorien darüber aufstellten, was ich eigentlich testen wollte. Eine Person fragte mich beim Test des zweiten Kalenders ganz konkret, welcher eigentlich »meiner« sei. Da die Testperson selbst Informatik studiert, lag wohl aus eigener Erfahrung die Vermutung nahe, dass dem Usability-Test eine praktische Arbeit vorausgegangen war. Ich gab natürlich keine Antwort auf diese Frage.

## 12.2.2 Testszenarien

Wie bereits ein Mal kurz erwähnt ist der eigentliche Test am System in drei Abschnitte unterteilt, von denen die ersten beiden über Unterfragen verfügen. Diese sind nachfolgend je Abschnitt durchnummeriert aufgelistet.

Da zwei unterschiedliche Systeme miteinander verglichen werden, gibt es zu jeder Frage zwei unterschiedliche Variationen, die sich in der grundlegenden Aufgabenstellung gleichen, aber in Details unterscheiden. Es wird nicht zwei Mal die exakt gleiche Frage gestellt, um zu verhindern, dass diese bei der Evaluierung des zweiten Kalenders aus dem Gedächtnis statt aufgrund der Kalendernutzung beantwortet wird. In der nachfolgenden Auflistung werden diese Variationen durch die Großbuchstaben A und B gekennzeichnet. Die Aufgaben 1A und 1B sind also beide die ersten Unterfragen ihres Abschnittes und in Hinsicht auf den beabsichtigten Beobachtungsschwerpunkt äquivalent.

Abschnitt 1: Orientierung

A	ıfgabe	Formulierung/Beschreibung						
	A	Welches konkrete Datum hat der kommende Dienstag?						
1	В	Welches konkrete Datum hat der kommende Montag?						
		Durch diese Frage soll festgestellt werden, wie die Testpersonen ein Datum ablesen, das in der nahen Zukunft liegt. Wird das Datum in TaBaKa eher aus der Detailansicht oder aus der Navigationsleiste abgelesen? Wird es direkt abgelesen oder wird die Navigationsleiste zur Unterstützung verschoben? Welche Ansicht wird in Lightning für diese Fragestellung genutzt?						
	A	Welches konkrete Datum hatte der Donnerstag vorletzter Woche?						
2	В	Welches konkrete Datum hatte der Mittwoch vorletzter Woche?						
		Diese Aufgabenstellung ist ähnlich zu Aufgabe 1, betrachtet aber einen etwas größeren Zeitraum und erfordert eine Bewegung in die entgegengesetzte Richtung. Besonders interessant ist hierbei, ob der Benutzer in TaBaKa über die linke Schaltfläche navigiert oder ob er die Drag-and-Drop-Funktionalität erkennt und nutzt.						
	A	Welches konkrete Datum hatte der erste Montag im März?						
3	В	Welches konkrete Datum hatte der letzte Freitag im Februar?						
		In dieser Aufgabe wird beobachtet, wie die Navigation zu einem anderen Monat erfolgt, was in TaBaKa entweder durch das Dropdown-Menü beim Anklicken des aktuellen Monatsnamens oder durch Scrollen in der Navigationsleiste erreicht werden kann.						
	A	Welcher Wochentag war am ersten Jänner des laufenden Jahres?						
4	В	Welcher Wochentag war am 31. Jänner des laufenden Jahres?						
		Diese Aufgabenstellung ist ähnlich zu Aufgabe 3, nur dass hier nach einem Wochentag statt nach einem konkreten Datum gefragt wird. Durch diese zwei ähnlichen Beispiele lässt sich besser beobachten, ob verschiedene Navigationsmethoden ausprobiert werden oder ob wieder das gleiche Schema angewandt wird.						
	A	Welcher Wochentag war am 14.03.1987?						
5	В	Welcher Wochentag war am 21.10.1992?						
		In der letzten Aufgabe wird überprüft, wie weit entfernt liegende Daten angesteuert werden. Wer bisher nur über die Navigationsleiste navigiert hatte, wird in diesem Beispiel darauf angewiesen sein nach neuen Navigationsmethoden zu suchen. Dabei soll festgestellt werden, ob die Navigation über die Spaltenüberschriften intuitiv verständlich ist.						

Abschnitt 2: Terminplanung

Aı	ufgabe	Formulierung/Beschreibung
1	A	Ihr Arbeitgeber benötigt in den nächsten zehn Tagen (also bis zum 9. Mai) eine Sonderschicht von Ihnen innerhalb der Bürozeiten (Mo. bis Fr. von 08:00 bis 17:00). An welchem dieser Tage hätten Sie grundsätzlich noch die meiste freie Zeit zu Verfügung? (Muss nicht am Stück sein.) Bitte beachten Sie dabei, dass der 1. Mai ein Feiertag ist! (Feiertage sind aus technischen Gründen in diesem Test nicht im Kalender eingetragen.)
	В	Für eine Lehrveranstaltung müssen Sie gemeinsam mit mehreren Kollegen in den nächsten zehn Tagen eine Laborübung absolvieren. Das Labor ist jeweils Montag bis Freitag von 08:00 bis 17:00 Uhr geöffnet. Sie wollen Ihren Kollegen Terminvorschläge machen, aber dabei auf jeden Fall den Tag ausschließen, der bei Ihnen ohnehin schon am dichtesten verplant ist. An welchem der zehn Tage haben Sie am wenigsten freie Zeit zu Verfügung?
		Der Zeitraum ist so gewählt, dass man in einer klassischen Wochenansicht zwischen zwei Seiten blättern müsste. In dieser Aufgabe wird überprüft, ob in TaBaKa die Anpassungsmöglichkeit des Zeitraumes erkannt und genutzt wird.
A 2		Im Juli wollen Sie für vier Tage mit einem Bekannten nach Salzburg fahren. Gehen Sie davon aus, dass Sie frühmorgens losfahren und spätabends zurück kommen – diese vier Tage müssen also komplett frei sein! Wann wäre der frühest mögliche Termin für die Abfahrt, sodass es zu keinen Überschneidungen mit anderen Terminen kommt?
	В	Im August wollen Sie für fünf Tage zu einer großen Veranstaltung nach Deutschland fahren. Gehen Sie davon aus, dass Sie frühmorgens losfahren und spätabends zurück kommen – diese fünf Tage müssen also komplett frei sein! Wann wäre der frühest mögliche Termin für die Abfahrt, sodass es zu keinen Überschneidungen mit anderen Terminen kommt?
		Die Termine in diesen beiden Monaten sind jeweils so angelegt, dass ein Termin außerhalb des standardmäßig angezeigten Zeitbereiches von 08:00 bis 17:00 Uhr liegt. Wird dieser Termin übersehen, gelangt man zu einem falschen Ergebnis. TaBaKa bietet gegen dieses Problem grundsätzlich zwei Maßnahmen, nämlich die Andeutung außerhalb liegender Termine und die einfache Anpassbarkeit des angezeigten Uhrzeit-Intervalls. Es wird überprüft, ob das grundsätzliche Problem als solches erkannt wird und ob die Maßnahmen in TaBaKa wahrgenommen und genutzt werden.

Abschnitt 3: Terminanalyse

	ufgabe	Formulierung/Beschreibung					
1	A	Für eine Gehaltsverhandlung mit Ihrem Arbeitgeber wollen sich einen Überblick verschaffen, wie oft Sie im ersten Quar (= Jänner bis März) des Jahres von Ihren üblichen Arbeitszeit abgewichen sind. Üblicherweise planen Sie immer Arbeitszeit von 08:00 bis 13:00. Zählen Sie alle Tage, an denen es Abweicht gen von diesem Zeitrahmen gegeben hat und notieren Sie hier Gesamtsumme!					
	В	Im Büro wird Ihnen Ihre Unflexibilität vorgeworfen, weil Sie immer nur Arbeitszeiten von 08:00 bis 13:00 Uhr wahrnehmen. Sie wissen aber, dass diese Anschuldigungen nicht stimmen und Sie vor allem am Ende der letzten Sommerferien und am Beginn des darauf folgenden Semesters oft von Ihren üblichen Zeiten abgewichen sind. Zählen Sie alle Tage, an denen es in den Monaten September bis November des Vorjahres Abweichungen von diesem Zeitrahmen gegeben hat und notieren Sie hier die Gesamtsumme!					
		In dieser letzten Aufgabe soll die Testperson einen größeren Zeitraum analysieren, in dem noch einmal die Möglichkeit besteht, die spezifischen Eigenschaften von TaBaKa – diesmal auch in Kombination miteinander – einzusetzen. Vor allem soll wieder darauf geachtet werden, ob die Anpassung des Tagesintervalls und des Zeitrahmens genutzt wird. Da auf dem Testrechner aufgrund der relativ geringen Auflösung nicht drei Monate gleichzeitig in der Navigationsleiste angezeigt und ausgewählt werden können, ist es auch interessant zu beobachten, wie durch den Kalender navigiert wird. Insgesamt soll bei dieser – im Vergleich zu den vorigen Übungen – längeren Aufgabe auch beobachtet werden, ob sich aufgrund des höheren zu erwartenden Aufwandes eine größere Motivation bildet spezifische Möglichkeiten wie die Anpassung des Zeitraumes zu erkunden und ob sich nach etwas längerer Tätigkeit das Nutzungsverhalten ändert.					

Nach jedem Abschnitt – mit Ausnahme des letzten – soll die Testperson wieder zum aktuellen Tag zurückkehren. Dadurch wird einerseits auch die Auffindbarkeit des Heute-Links getestet und andererseits eine einheitliche Ausgangslage für die jeweils nachfolgende Übung geschaffen.

# 12.2.3 Testmaterial und Einrichtung

Alle Tests wurden auf einem Laptop mit folgender Konfiguration abgehalten:

Prozessor: Intel(R) Celeron(R) M CPU 440 @ 1.86 GHz

**RAM: 766 MB** 

Betriebssystem: Microsoft Windows Vista Home Premium, 32 Bit

Bildschirmauflösung: 1.280 x 800 px

Thunderbird-Version: 17.0.7

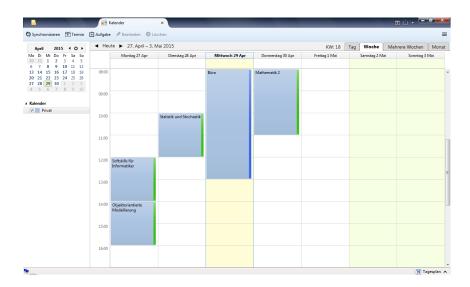


Abbildung 32: Vorkonfiguration Lightning: In dieser Form bekamen Testpersonen Lightning vorgesetzt

Lightning-Version: 1.9.1 Browser: Firefox 22.0

Um die allgemeine Bedienung zu erleichtern, war der Laptop mit einer Maus und einem Mauspad ausgestattet.

Die Angabe erhielt jeder Proband einseitig ausgedruckt auf Papier, links oben geheftet, zusammen mit einem Kugelschreiber, um die Fragen beantworten zu können.

Ich als Tester hatte ebenfalls eine Angabe vor mir, auf der ich mir Notizen machte. Allerdings war diese aus Gründen der Effizienz auf die Kernfragen ohne Erläuterungen beschränkt und beidseitig gedruckt.

Während die Testpersonen die ersten Seiten ausfüllten und lasen, bereitete ich jeweils den Laptop vor, indem ich in der Reihenfolge des Tests die beiden Kalender öffnete und bei Bedarf das Datum wieder auf das in der Angabe vorgegeben Heute-Datum setzte. Sobald die Testperson zur Kurzbeschreibung des ersten zu untersuchenden Kalendersystems gelangte, schob ich ihr den Laptop, auf dem dieses bereits geöffnet war, zu.

In Lightning war dabei standardmäßig immer die Wochenansicht geöffnet und TaBaKa zeigte in der Voreinstellung sieben Tage, sodass die Vorkonfiguration in beiden Systemen möglichst ähnlich und dadurch ein direkter Vergleich gut möglich war. Um trotzdem auch die spezifischen Stärken der beiden Kalender ausspielen zu können, wurde aber in beiden Fällen ausdrücklich darauf hingewiesen, dass man in eine andere Ansicht wechseln beziehungsweise den angezeigten Bereich verändern darf.

Für größere Ähnlichkeit und Vertrautheit wurden in Lightning außerdem der Suchbereich ausgeblendet und die Darstellung so verändert, dass Wochen wie in unserem Umfeld üblich mit Montag – statt wie vorgegeben mit Sonntag – beginnen.

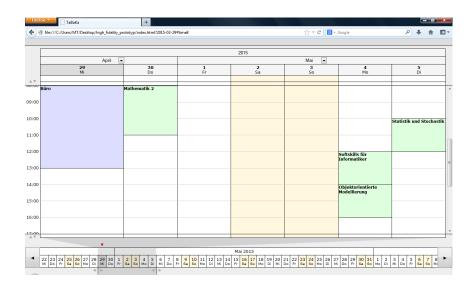


Abbildung 33: Vorkonfiguration TaBaKa: In dieser Form bekamen Testpersonen TaBaKa vorgesetzt

Wie schon erwähnt wurde für alle Tests ein einheitliches Heute-Datum vorgegeben und als Systemzeit voreingestellt, sodass trotz Tests an unterschiedlichen Tagen immer eine einheitliche Ausgangssituation herrschte.

Als Datum wurde hierbei Mittwoch, der 29. April 2015 gewählt. Das Jahr liegt bewusst in der Zukunft, um zu verhindern, dass Leute mit Zeiträumen innerhalb des Tests tatsächliche Ereignisse aus ihrem Leben in Verbindung bringen. Als Monat wurde ein anderer als der tatsächlich aktuelle Juli ausgewählt, sodass deutlicher sichtbar wird, dass das heutige Datum im Test nicht dem tatsächlichen entspricht. Mittwoch, der 29. wurde gewählt, weil es sich dabei um ein Datum handelt, das es bei entsprechenden Aufgabenstellungen durch Wochen- und Monatsumbrüche leicht notwendig macht in klassischen Kalendern vor- oder zurückzublättern und damit den Unterschied zu TaBaKa besonders deutlich aufzeigt.

Als vorgegebene Termine wurden Einträge aus drei verschiedenen Kategorien (Lehrveranstaltungen, Prüfungen und Bürozeiten) eingetragen.

Ich saß bei allen Tests rechts von der Testperson, um mit der rechten Hand etwas abseits Notizen machen zu können und um nach Absolvierung der Aufgaben zum ersten Kalender mit der Maus zum zweiten Kalendersystem wechseln zu können.

### 12.2.4 Design und Messungen

Um Erscheinungen zu vermeiden, die durch bestimmte Anordnungen von Aufgabenstellungen zustandekommen könnten – wie etwa unerwartete Lerneffekte – wurden die Aufgabenstellungen bei allen Testpersonen in einem gewissen Rahmen variiert.

Die einzelnen Abschnitte des Tests bauen grundsätzlich aufeinander auf. Im ersten Abschnitt wird die grundlegende Navigation getestet, im zweiten wird

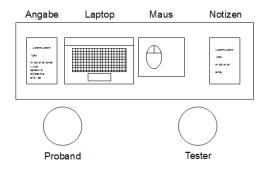


Abbildung 34: Setting während der Tests

diese um Termine erweitert und im dritten werden Navigation und Termine gemeinsam für eine komplexere Aufgabe betrachtet. Da es relativ sinnlos wäre grundlegende Navigationsmechanismen zu testen, nachdem diese schon in einer weitaus aufwändigeren Übung angewandt wurden, bleibt die Abfolge der einzelnen Abschnitte in allen Tests identisch.

Auch die Unterpunkte in Abschnitt 1 sind großteils aufeinander aufbauend, weil man erst durch Wochen-, dann durch Monats- und schließlich durch Jahresintervalle navigiert, wobei die jeweils vorige Stufe aber weiterhin beachtet werden muss. Vertauscht werden können hier aber die Aufgaben eins und zwei sowie drei und vier, da diese jeweils in die gleiche Zeitraum-Kategorie fallen.

Ebenfalls vertauscht werden können die Aufgaben aus Abschnitt 2, da diese voneinander unabhängig und auf unterschiedliche Aspekte ausgerichtet sind.

Leichter als die Reihenfolgen lassen sich die Angaben-Variationen für die beiden Kalender auf unterschiedliche Weise verteilen. Da A und B im Prinzip immer äquivalente Aufgabenstellungen sind, lassen diese sich auch problemlos gegeneinander tauschen und sorgen damit für eine große Diversität in den Tests.

Ebenfalls austauschbar ist die Reihenfolge der zu testenden Kalender. Ob zuerst TaBaKa oder erst Lightning getestet wird, kann mitunter auch einen großen Einfluss darauf haben, wie die beiden Kalender von den Testpersonen wahrgenommen werden, weshalb eine gleichmäßige Verteilung beider Variationen hier wichtig ist.

Alle Variationsmöglichkeiten berücksichtigt wurde folgende Verteilung erstellt:

1. Kalender		Aufgaben für 1. Kalender							
	1. Kalender			1	2		3		
Test 1	Lightning	1A	2A	3A	4A	5A	1A	2A	1A
Test 2	TaBaKa	2B	1A	3A	4B	5A	2A	1B	1B
Test 3	Lightning	1B	2A	4A	3В	5B	1A	2B	1A
Test 4	TaBaKa	2A	1A	3A	4A	5B	2B	1B	1B
Test 5	Lightning	1B	2B	4B	3В	5A	1A	2A	1A
Test 6	TaBaKa	1A	2A	3В	4B	5A	2B	1A	1B
Test 7	Lightning	2B	1B	3A	4A	5B	1B	2A	1A

Die Aufgaben für den zweiten zu testenden Kalender entsprechen jeweils denen des ersten, jedoch mit einem A statt einem B und vice versa.

Test 1 wird abgesetzt dargestellt, weil es sich dabei um den Pilottest handelt. In den restlichen vier Tests wurden Variationen möglichst gleichmäßig umgesetzt, um eine gute Durchmischung zu erzielen.

Bei Test 2 und 3 wurde abwechselnd jeweils eine Aufgabe auf A und eine auf B gesetzt, sodass die beiden Tests genau zueinander konträr sind. Bei Test 4 und 5 haben auf ebenfalls gespiegelte Weise jeweils vier aufeinanderfolgende Aufgaben die selbe Zuweisung aus A und B und in Test 6 und 7 erhielten jeweils zwei Aufgaben die gleiche Variation. Startkalender und Vertauschungen der Reihenfolge wurden anschließend ebenso gleichmäßig eingebracht.

Zeitmessungen wurden jeweils dann gestartet, wenn die Testperson eine Angabe gelesen hatte und sich dem Monitor zuwandte. Gestoppt wurden sie, wenn die letzte Teilaufgabe gelöst und niedergeschrieben war.

Weil ich in meiner ursprünglichen Planung damit gerechnet hatte, dass pro Aufgabe etwa zehn Minuten Arbeit investiert werden müssten, hielt ich die Sekundenwerte in Hinsicht auf signifikante Unterschiede für vernachlässigbar und hatte mich darauf festgelegt die gemessenen Werte auf volle Minuten zu runden. Da die Aufgaben letztendlich aber doch deutlich schneller zu lösen waren, wäre nachträglich betrachtet eine genauere Erfassung wesentlich aussagekräftiger gewesen.

# 12.2.5 Pilottest

Der Pilottest wurde am 09.07.2013 durchgeführt und hatte folgende Änderungen in der Angabe zur Folge:

• In der Einleitung zu Abschnitt 1 stand ursprünglich »Je nach konkretem Anwendungsfall kann es aber oft sinnvoll sein [...] einen bestimmten Wochentag herauszufinden.«. Dies wurde erweitert zu »Je nach konkretem Anwendungsfall kann es aber oft sinnvoll sein [...] einen bestimmten Wochentag (Montag, Dienstag, ...) herauszufinden.«

Ursache dafür war, dass der Testperson bei der ersten Frage nach einem Wochentag nicht klar war, was mit dem Begriff »Wochentag« gemeint ist.

- Die Erklärung, dass man für das Testszenario davon ausgehen soll, dass heute der 29. April 2015 ist, wurde durch ein höheres Schriftgewicht hervorgehoben, weil die Testperson den Text nur überflogen hatte und im späteren Verlauf nachfragte, ob das Datum absichtlich nicht dem tatsächlichen entspricht.
- In Abschnitt 2 Aufgabe 1A stand ursprünglich »Ihr Arbeitgeber benötigt in den nächsten zehn Tagen [...]«. Dies wurde geändert zu »Ihr Arbeitgeber benötigt an einem der nächsten zehn Tage [...]«, weil die Testperson die ursprüngliche Formulierung so verstanden hatte als müsste man an allen dieser Tage arbeiten, was zur Folge hatte, dass die Aufgabenstellung nicht verständlich war.
- In der einführenden textlichen Beschreibung wurde ergänzt, dass die nachfolgenden Übungen zu beiden Kalendern ähnlich, aber nicht identisch sind. Ursache dafür war, dass die Testperson in Abschnitt 2 Aufgabe 2B die Angabe nur überflogen und anschließend vorerst wieder nach vier freien Tagen wie in Aufgabe 1A, statt nach fünf Tagen wie gefordert gesucht hatte.
- In der Kurzeinführung zu TaBaKa wurde die Navigationsleiste etwas genauer beschrieben, weil der Unterschied im Konzept gegenüber klassischen Kalendern offenbar nicht klar genug vermittelt worden war. In Abschnitt 2 Aufgabe 2B fragte der Proband »Wo kann ich mir den Monat anzeigen lassen?« und kam letztlich zu dem falschen Schluss, dass es »nur eine Wochenansicht« gibt.

Insgesamt wirkte die Testperson während des ganzen Tests sehr unkonzentriert, überflog die meisten Texte nur, benötigte anschließend aber im Gegenzug verhältnismäßig viel Zeit zur Lösung der Aufgaben. Als der Proband die Monatsansicht in TaBaKa suchte, sagte er auch selbst, dass die Antwort auf seine Frage möglicherweise irgendwo gestanden sei, er sie aber nicht gelesen hat.

### 12.3 Ergebnisse

Nachfolgend werden die wesentlichen Ergebnisse beschrieben, zu denen der Usability-Test geführt hat. Da die Anzahl der Testpersonen relativ gering war, liegt der Schwerpunkt dabei auf der qualitativen Analyse.

### 12.3.1 Quantitative Analyse

In diesem Kapitel werden die Ergebnisse in Zahlen dargestellt. Der Pilottest ist nicht Teil der quantitativen Analyse, weil nach diesem noch Anpassungen an der Angabe vorgenommen wurden. Dadurch werden insgesamt sechs Tests analysiert.

#### Korrektheit

In der nachfolgenden Tabelle wird angegeben, in wie vielen Tests die jeweilige Aufgabe letztendlich korrekt beantwortet wurde. Der im Kalendervergleich jeweils bessere Wert – sofern nicht beide Zahlen identisch sind – wird dabei zur Verdeutlichung in Fettschrift hervorgehoben.

Aufgabe:	1.1.	1.2.	1.3.	1.4.	1.5.	2.1.	2.2.	3.1.	Summe
Lightning	5	6	6	6	6	4	6	2	41
TaBaKa	6	6	6	5	5	5	3	3	39

Wie man sieht, unterscheiden sich die Ergebnisse zum größten Teil nur minimal. Die einzige deutliche Differenz tritt in Aufgabe 2.2. zugunsten von Lightning auf. Die entsprechende Aufgabenstellung war darauf ausgerichtet die Markierungen für Termine außerhalb des derzeit angezeigten Zeitintervalls in TaBaKa zu testen. Wie sich im Test zeigte, wird diese Markierung jedoch von den meisten Personen übersehen.

Lightning schnitt in dieser Aufgabe dagegen sehr gut ab, weil die Testpersonen die entsprechende Aufgabe hier meistens über die Monatsansicht oder »Mehrere Wochen« lösten, wo aufgrund der wenigen Termine pro Tag immer alles angezeigt werden konnte.

Etwas verwundert bin ich zugegeben über die durchwegs hohe Fehlerquote in Aufgabe 3 in beiden Kalendern. Die meisten Personen zählten hier weniger Termine als tatsächlich zu finden gewesen wären, obwohl die von mir beobachtete Vorgehensweise im Allgemeinen zumeist korrekt war. Ich vermute, dass die Testpersonen sich durch die Beobachtung etwas gestresst fühlten und deshalb eher dazu neigten, etwas zu übersehen.

### Lösungszeiten

Die nächste Tabelle zeigt, wie lang die durchschnittliche Lösungszeit als arithmetisches Mittel in Minuten je Aufgabe war.

In dieser Aufstellung wurden die beachteten Datensätze etwas weiter eingeschränkt. Testperson 6 wurde für die Berechnung von Aufgabe 3 nicht berücksichtigt, weil diese Aufgabe in Lightning nicht beendet wurde (– die Testperson kapitulierte –) und in TaBaKa ein grundsätzliches Missverständnis über die Funktionalität herrschte, sodass die Aufgabe zwar in Rekordzeit, aber mit einer vollkommen falschen Lösung abgeschlossen war. Weiter wurde auch Testperson 7 nicht in die Berechnung für Aufgabe 3 einbezogen, weil ich am Ende dieses Tests vergessen hatte die Zeit zu stoppen.

<u>Durchschnittliche Lösungszeiten</u> in Minuten:

Aufgabe:	1	2	3
Lightning	2,5	3,7	3,0
TaBaKa	3,7	4,0	2,3

Standardabweichung in Minuten:

Aufgabe:	1	2	3
Lightning	0,5	0,5	1,7
TaBaKa	1,2	1,7	0,5

Wie man sieht, benötigten die Testpersonen etwas länger, um sich in TaBaKa zurechtzufinden, was aufgrund der Neuheit zu erwarten war. In der etwas komplexeren Aufgabe 2 gibt es dann aber schon keinen nennenswerten Geschwindigkeitsunterschied mehr und in Aufgabe 3, die bewusst auf die Stärke dieses Alternativkalenders zugeschnitten ist, sieht man schließlich einen Geschwindigkeitsvorteil.

Bedenkt man den reduzierten Datenumfang, ist zwar ebenso die Aussagekraft dieser Zahlen reduziert, aber unter Beachtung der Tatsache, dass ein Datensatz gerade deshalb ausgeschlossen wurde, weil bei Aufgabe 3 in Lightning kapituliert wurde, wird die These, dass TaBaKa hier im Vorteil war, noch gestärkt.

Offen bleibt die Frage, inwiefern sich in diesem Ergebnis auch ein Lerneffekt widerspiegelt, da die drei Aufgaben aufeinander aufbauen. Nachträglich betrachtet wäre es sicher interessant gewesen, nach Aufgabe 3 noch einmal ähnliche Aufgaben wie in Aufgabe 1 zu absolvieren, um zu sehen, wie groß der Unterschied zwischen Lightning und TaBaKa dann noch ist.

Was man an der Standardabweichung recht gut erkennen kann, ist der Umstand, dass einfachere Aufgaben in klassischen Kalendern offenbar von allen Personen in etwa der gleichen Zeitspanne durchgeführt werden können, während dagegen die Einarbeitung in eine neue Darstellungsform für manche einfacher und für andere schwieriger ist. Die größte Spanne gemessener Zeiten brachte hierbei Aufgabe 2 in TaBaKa mit sich: Die schnellste Person benötigte hier nur zwei Minuten zur Lösung, die langsamste sieben.

Nur in Aufgabe 3 ist die Standardabweichung in TaBaKa deutlich geringer, was wohl darauf zurückzuführen ist, dass hier im Vergleich zu Lightning die schwierige Entscheidung wegfiel, welche Ansicht man nutzen sollte, wodurch der Lösungsweg klarer vorgegeben war.

#### Beurteilung

In der nachfolgenden Tabelle kann abgelesen werden, wie oft einzelne Punkte bei der Endbewertung angekreuzt wurden.

Eigenschaft	Lightning	beide gleich	TaBaKa
Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	3	2	1
Alltagstauglichkeit	3	1	2
Navigation	3	2	1
Verständlichkeit	4	1	1
Übersichtlichkeit	3	2	1
Einfachheit	4	0	2
Optik/Ästhetik	4	2	0
Zentrierung auf das Wesentliche	2	2	2
Effiziente Nutzung des vorhandenen Platzes	0	4	2
Bewahrung vor Fehlern	1	4	1
Flexibilität	2	3	1
Kennzeichnung unterschiedlicher Terminkategorien	1	3	2

Die Bewertung fiel im Allgemeinen stark zugunsten von Lightning aus. Wie schon am Beginn dieser Arbeit erwähnt, war dieses Ergebnis zu erwarten, da fast jeder im Umgang mit klassischen Kalendern vertraut ist und es schwer fällt so ein etabliertes Konzept gegen etwas Anderes zu ersetzen. Damit decken sich die Ergebnisse auch mit jenen von FishCal [3], wo trotz messbarer Vorteile in komplexen Aufgaben subjektive Bewertungen – vor allem in allgemeinen Kategorien wie »Attractiveness« und »Behavior« – eher zugunsten herkömmlicher Darstellungen ausfielen.

Eine nicht unbeträchtliche Menge an Personen räumt aber zumindest eine Gleichwertigkeit in vielen Belangen ein und TaBaKa findet auch in fast jedem Punkt zumindest einen Fürsprecher. Das einzige, was niemand an TaBaKa präferiert, ist die Optik, was aber in Hinblick darauf, dass es sich nur um einen Prototypen handelt, eine vorerst vernachlässigbare Kritik ist.

Am vorteilhaftesten wird in TaBaKa nach diesen Zahlen offenbar noch die effiziente Nutzung des Platzes wahrgenommen. Das ist die einzige Eigenschaft, bei der niemand für den sonst präferierten Kalender Lightning gestimmt hat.

In der Kennzeichnung der Terminkategorien ist TaBaKa auch knapp in Führung. Mindestens eine Person bemerkte die dezentere farbliche Unterscheidung der Terminkategorien in Lightning erst in der letzten Aufgabe, was sie durch einen Kommentar bemerkbar machte. Die einzige Person, die hier für Lightning stimmte, begründete diese Entscheidung damit, dass dort auch mehrere Kalender unterscheidbar seien – also mit einer Funktionalität, die im Prototypen noch nicht umgesetzt, aber in vereinfachter Form grundsätzlich geplant

ist.

# 12.3.2 Beobachtungen in Lightning

April 2015 ◀ O ▶	<b>◄</b> Heute	▶ 27. April – 3. Ma	ai 2015			KW: 18	Tag Woche M	ehrere Wochen   Monat
Mo Di Mi Do Fr Sa So		Montag 27 Apr	Dienstag 28 Apr	Mittwoch 29 Apr	Donnerstag 30 Apr	Freitag 1 Mai	Samstag 2 Mai	Sonntag 3 Mai
30 31 1 2 3 4 5								
6 7 8 9 10 11 12								
13 14 15 16 17 18 19	08:00			Büro	Mathematik 2			
20 21 22 23 24 25 26								
27 28 29 30 1 2 3								
4 5 6 7 8 9 10								
	09:00							

Abbildung 35: Bevorzugte Navigationselemente in Lightning (grün gekennzeichnet): Zum Blättern wird vorwiegend der Navigationskalender genutzt, nur der Heute-Button in der Detailansicht ist offenbar intuitiver verständlich als das Kreissymbol mit der selben Funktionalität. Das Datum (rot gekennzeichnet) wurde drei Mal angeklickt, obwohl es mit keiner Funktion hinterlegt ist.

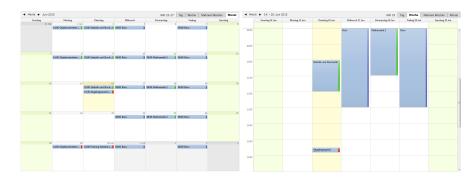


Abbildung 36: Bevorzugte Ansichten für Aufgabe 3 in Lightning: Monatsansicht (links) und Wochenansicht (rechts)

Zu Beginn dieser Arbeit wurde unter anderem Lightning in Bezug auf seine Benutzbarkeit hin diskutiert. Bei dieser ersten Analyse stellte ich gewisse Schwachpunkte fest, die im Usability-Test zum Teil bestätigt werden konnten. Zum Teil konnte ich aber auch Verhalten beobachten, das ich so nicht unmittelbar erwartet hätte. Insgesamt fielen mir bei der Durchführung der Tests nachfolgende Punkte besonders auf:

- Was ich nicht erwartet hatte, ist die Tatsache, dass die Benutzer während des gesamten Tests weitestmöglich auf den Navigationskalender links oben konzentriert waren. Aufgabe 1 wurde in der Regel allein über diesen Kalender bewältigt, die Detailansicht wurde erst beachtet, als die Termine relevant wurden. Bis zum Schluss wurde aber so gut wie ausschließlich über den Navigationskalender navigiert, währen die Buttons zum Blättern in der Detailansicht kaum genutzt wurden.
- Das einzige Navigations-Element, das in der Hauptansicht häufig genutzt wurde, war der Heute-Button. Diese Beobachtung deckt sich mit meiner Vorannahme, dass das Kreissymbol, das im Vorschaukalender die gleiche

Funktion erfüllt, nicht intuitiv verständlich ist. Das Kreissymbol wurde grundsätzlich zwar auch ein paar Mal genutzt, aber davor wurde in der Regel immer der Tooltip abgewartet, um sich der Funktion sicher zu sein.

 Die Benutzer waren oft unentschlossen in welcher Ansicht sie arbeiten sollten. Aufgrund der relativ lockeren Verteilung der Testtermine waren grundsätzlich alle Ansichten außer der Tagesansicht sinnvoll. Die Testpersonen wechselten am Beginn einer Aufgabe oft zwischen mehreren Ansichten hin und her, bis sie sich für eine entschieden – manchmal änderten sie die Ansicht aber auch noch, nachdem sie schon mit der Abarbeitung begonnen hatten.

Eine Person löste Aufgabe 3 fast vollständig in der Monatsansicht, wechselte knapp vor dem Ende aber in die Wochenansicht und wiederholte dann die gesamte Zählung zur Kontrolle in dieser Ansicht. Eine Person verzählte sich auch am Beginn von Aufgabe 3, weil sie glaubte in der Monatsansicht zu sein, sich tatsächlich aber in »Mehrere Wochen« befand und das Scrollen mit dem Mausrad hier eine andere Wirkung als in der Monatsansicht hat. Eine Person löste Aufgabe 2.1B in der Ansicht »Mehrere Wochen«, achtete dort aber nicht auf die Termindauer, sondern nur auf die Anzahl der Termine. (Vielleicht erzeugten die gleich großen Einträge hier den Eindruck, dass die Termine gleich lang seien.)

Obwohl manche Testpersonen im anschließenden Gespräch meinten die Flexibilität mehrerer Ansichten zu bevorzugen, hinterlassen obige Beobachtungen eher den Eindruck, dass diese überfordernd statt fördernd wirken.

- Zwei Personen stellten von sich aus fest, dass Blättern und scrollen teilweise irreführend ist, was sich mit meiner Vorannahme deckt, dass das Mapping je nach Ansicht schwer nachvollziehbar ist.
- Den Pilottest mitgezählt machten drei Testpersonen in Aufgabe 3 Strichlisten oder schrieben sich Zwischenergebnisse auf, wenn Ihnen Lightning als erster Kalender vorgesetzt wurde. Dieses Verhalten wurde anschließend auch in TaBaKa als zweitem getesteten Kalender beibehalten.

Umgekehrt machte sich von den Personen, die zuerst TaBaKa testeten, keine einzige Notizen. Diese Testpersonen zählten alle Zwischenergebnisse im Kopf zusammen und behielten dieses Verhalten anschließend auch in Lightning bei.

In Lightning wurde meistens die Monatsansicht benutzt, in der man die Maus über jeden einzelnen Termin halten musste, um die Endzeit zu sehen. Dadurch hatte man neben dem Zählerstand immer auch andere Zahlen im Kopf. Die Wochenansicht als Alternative dazu wäre in relativ viele Unterseiten zersplittert gewesen, sodass man ständig auf das Datum achten müsste, um nicht zu weit zu zählen.

 Mehrere Personen waren sich in der Wochenansicht des Problems bewusst, dass sich oberhalb oder unterhalb Termine befinden könnten, die im angezeigten Zeitintervall nicht sichtbar sind. Die Maßnahmen gegen diese Ungewissheit waren aber bescheiden. Meistens wurde nur ein kleines Stück hinauf- und hinuntergescrollt und diese Kontrolle wurde meistens schon nach ein bis zwei Kalenderseiten wieder eingestellt. Das bestätigt die Sinnhaftigkeit eigener Markierungen oder einer Zeitraumanpassung, wie sie in TaBaKa grundsätzlich umgesetzt sind.

- Aufgabe 3 wurde in der Regel als mühsam zu lösen empfunden. Mehrere Testpersonen suchten an dieser Stelle des Tests nach zusätzlichen Funktionen ohne fündig zu werden. Eine Person kapitulierte, weil sie die Tooltips nicht bemerkt hatte und glaubte, jeden Termin einzeln öffnen zu müssen.
- Drei Personen versuchten in der Detailansicht durch einen Klick auf den Monatsnamen den Monat zu wechseln und mussten feststellen, dass es sich dabei nur um einen reinen Text handelt. Zwei dieser Personen hatten zuvor TaBaKa benutzt, in dem diese Funktion in der Detailansicht umgesetzt ist.
- Eine Person musste in der Monatsansicht erst den aktuellen Tag finden, indem sie den Kalender von oben nach unten durchging. Das deckt sich mit der These, dass die Anzeige von vergangenen Tagen vor dem aktuellen die Übersicht in der Detailansicht erschwert.
- Eine Testperson war darüber irritiert, dass bei der Navigation durch den Navigationskalender die Detailansicht nicht automatisch angepasst wird und fand erst nach längerem Suchen heraus, dass erst noch ein Klick auf einen spezifischen Tag notwendig ist.

#### 12.3.3 Beobachtungen in TaBaKa

Speziell in TaBaKa wurden während des Usability-Tests folgende Beobachtungen gemacht:

• Die unterschiedlichen Navigationsmöglichkeiten wurden von fast allen Personen genutzt. Als erstes Steuerelement wurde dabei zumeist auf die Schaltflächen links und rechts zugegriffen, was aufgrund der direkten Sichtbarkeit dieser Elemente zu erwarten war. Dass man die Leiste auch schieben oder Tage darauf anklicken kann, wurde in den meisten Fällen erst später entdeckt, aber dann oft auch genutzt, um schneller zu navigieren. Nur eine einzige Testperson fand nicht heraus, dass man die Leiste schieben kann und navigierte bis zum Schluss nur über die Schaltflächen.

Fast jeder hatte zumindest ein Mal einen Tag in der Leiste angeklickt. Diese Funktion wurde nach dem ersten Versuch aber eher selten eingesetzt. Eine Person meinte dazu nach dem Test, dass es schwierig zu verstehen sei, was genau dieser Klick bewirkt – sie hatte im konkreten Fall erwartet, dass die entsprechende Kalenderwoche des angeklickten Tages in den Fokus geladen wird.

Die einzige Navigationsmethode, die nur von einer Testperson angewandt wurde, war das Anklicken einer Tageskopfzelle in der Detailansicht, um den entsprechenden Tag in den Fokus zu rücken.

• Obwohl alle Testpersonen über die Schaltflächen durch die Navigationsleiste scrollten, bemerkte offenbar nur eine einzige Person, dass man diese Schaltflächen auch gedrückt lassen kann, um weiter in eine Richtung zu

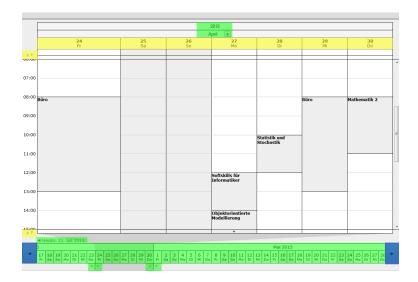


Abbildung 37: Nutzung der Navigations- und Anpassungselemente in TaBaKa: Grün gekennzeichnete Elemente wurden immer oder fast immer genutzt, gelb gekennzeichnete nur selten. Einen Sonderfall bilden die beiden Navigationsschaltflächen (blau): Diese wurden zwar von allen angeklickt, aber nur eine Person bemerkte, dass man sie auch länger drücken kann, um vor- oder zurückzuscrollen.

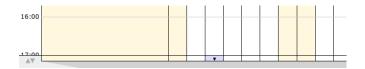


Abbildung 38: Die Markierungen, die in TaBaKa auf Termine außerhalb des angezeigten Zeitintervalls hinweisen sollen, wurden in der Regel übersehen.

scrollen. Die meisten Personen klickten immer nur kurz auf diese Schaltflächen, um jeweils einen Tag weiterzuscrollen. Wenn sie mehrere Tage überspringen wollten, klickten sie wiederholt.

- TaBaKa enthielt ursprünglich einen Bug, aufgrund dessen die Buttons nicht reagierten, wenn man sie zu schnell anklickte. Dieser Fehler trat erst in Test 2 auf, weil weder ich noch die Testperson im Pilottest so schnell auf die Schaltflächen klickten wie es Testperson 2 tat. Damit nachfolgende Testpersonen TaBaKa nicht aufgrund dieses Bugs schlechter bewerten statt sich auf das Konzept zu konzentrieren, wurde dieser Fehler nach Test 2 behoben.
- Nachdem in Folge des Pilottests die Angabe so umgeschrieben worden war, dass explizit auf die Möglichkeit der Anpassung des Tagintervalles hingewiesen wurde, wurde diese Funktion auch von allen Testpersonen in Anspruch angenommen und im anschließenden Gespräch mehrmals als

vorteilhaft angeführt. Gelegentlich war das Verschieben der Intervallgrenzen anfangs nicht ganz klar, was dazu führte, dass die Testpersonen erst nur kurz auf eine Grenze klickten und dann erst später wieder auf dieses Element zurückkamen und entdeckten, wie es tatsächlich funktioniert.

- Die zum Tagintervall recht ähnliche Möglichkeit zur Anpassung des angezeigten Uhrzeit-Bereiches wurde nur von zwei Personen genutzt.
- Wie schon an anderer Stelle erwähnt wurden die Markierungen für Termine außerhalb des angezeigten Zeitintervalls (siehe Abbildung 38) weitgehend übersehen. Eine Person entdeckte sie zwar während des Tests, verstand aber ihre Bedeutung nicht und hielt auch nie die Maus lange genug darüber, um den erklärenden Tooltip zu sehen. Eine Person entdeckte die Markierungen erst als sie nach dem Test noch ein wenig in TaBaKa herumprobierte. Dieser zweiten Person war die Bedeutung dann aber auf Anhieb klar.
- Da von den meisten Testpersonen weder die Markierungen für Termine außerhalb des angezeigten Zeitintervalls noch die Möglichkeit zur Anpassung des Zeitintervalls erkannt wurden, agierten die meisten ähnlich wie in der Wochenansicht in Lightning und scrollten ein wenig auf und ab, um sich zu versichern, dass keine Termine außerhalb liegen. Wie in Lightning scrollte auch hier niemand ganz nach oben und ganz nach unten und diese Kontrolle wurde eher stichprobenartig statt konsequent durchgeführt.
- Zwei Personen bezeichneten das Eintippen der Jahreszahl als mühsam. Da der Test auf einem Laptop ohne Nummernblock stattfand, war das auch nach meinem Empfinden deutlich unintuitiver als auf dem PC, auf dem ich TaBaKa entwickelt hatte. Im Gegenzug gab es aber auch eine Testperson, die erst TaBaKa vorgesetzt bekam und es danach in Lightning als mühsam empfand die Jahreszahl nicht eintippen zu können. Im Optimalfall sollten also wohl beide Optionen zur Verfügung stehen.
- Eine Person tippte die Jahreszahl ein, drückte danach aber nicht »Enter«, sondern klickte den Monat zum Wechseln an. Weil ich diesen Fall in der Implementierung nicht berücksichtigt hatte, wurde so nur der Monat, nicht aber das Jahr gewechselt.
- Etwa die Hälfte der Testpersonen machte durch direkte Äußerungen verständlich, dass sie in TaBaKa klassische Ansichten suchen. Zwei Personen äußerten, dass TaBaKa offenbar »nur eine Wochensicht« habe, bevor sie die Anpassung des Tagintervalls entdeckten. Eine Person kritisierte aber auch nach dem Test und trotz vorgenommener Intervallanpassung das Fehlen einer Monatsansicht.
- Drei Personen waren sich anfangs etwas unklar darüber, welcher Tag der aktuelle ist. Zwei dieser Personen bemerkten schließlich die rote Markierung in der Navigationsleiste, die dritte konnte an der grauen Markierung in der Detailansicht den aktuellen Tag eruieren.
- Eine Person suchte im Navigationskalender eine Markierung, die auf Termine hindeutet ähnlich den fettgeschriebenen Zahlen im Navigationskalender von Lightning.

- Eine Testperson wies mich bezüglich des Namens »TaBaKa« darauf hin, dass »Baka« japanisch sei und übersetzt »Idiot« bedeute.
- Zwei Personen meinten im abschließenden Gespräch, dass TaBaKa an sich gut sei, dass man sich allerdings etwas damit beschäftigen müsse beziehungsweise eine Anleitung benötigt, um leichter damit zurecht zu kommen.

# 13 Anknüpfpunkte für aufbauende Arbeiten

Wie der Usability-Test gezeigt hat, ist trotz einer gewissen Skepsis ein gewisses Potential in TaBaKa vorhanden, auf das man aufbauen kann.

Als erster Schritt zur Weiterführung der Arbeit sollten die bestehenden Fehler behoben werden:

- Das Verschieben der Uhrzeitgrenzen muss so so eingeschränkt werden, dass diese nicht mehr übereinander geschoben werden können.
- Die Tagintervallgrenzen sollen so weit zusammengeschoben werden können, dass nur noch ein einziger Tag im Fokus liegt.
- Wenn man eine Jahreszahl eingibt und danach das Feld verlässt ohne »Enter« zu drücken, soll trotzdem das eingegebene Jahr geladen werden.

Danach sollten Verbesserungen im Sinne der Beobachtungen im Usability-Test vorgenommen werden:

- Die Markierungen für Termine außerhalb des angezeigten Uhrzeitintervalls müssen deutlicher hervorgehoben werden, damit Benutzer sie wahrnehmen. Denkbar wäre etwa ein kräftigerer Farbton oder ein mehrmaliges Aufblinken dieser Elemente, sobald sie in die aktuelle Detailansicht gelangen. (Von einem dauerhaften Blinken ist natürlich abzuraten, weil dieses von Aufgaben innerhalb des aktuellen Zeitintervalls zu stark ablenken würde.)
- Die Jahreszahl soll nicht nur durch Eintippen von Ziffern änderbar sein, sondern auch durch das Auswählen einer Zahl aus einer Liste. Aufgrund der räumlichen Aufteilung würde es sich hier anbieten, die Breite der Zelle zu nutzen, in der sich die Jahreszahl befindet. Man könnte etwa links der aktuellen Jahreszahl vergangene Jahre und rechts davon zukünftige Jahre auflisten, sobald man die Maus über die Zelle bewegt oder die Jahreszahl anklickt.
- Um den Ein- beziehungsweise Umstieg zu erleichtern könnte die Statuszeile des Browsers genutzt werden, um immer aktuelle Informationen über das Element anzuzeigen, über dem sich die Maus gerade befindet. Damit lassen sich auch Interaktionsmöglichkeiten kommunizieren, die nicht direkt sichtbar sind etwa das längere Drücken einer Schaltfläche. Gegenüber den bisher teilweise eingesetzten Tooltips hätten Statuszeilentexte auch den Vorteil, dass sie ohne Verzögerung angezeigt werden, sodass man sofort sieht, ob es zu einem Element zusätzliche Informationen gibt.

Alternativ könnte man auch Tooltips statt wie bisher über das Title-Attribut in HTML durch eine flexiblere JavaScript-Implementierung umsetzen, sodass auch diese ohne Verzögerungen angezeigt werden. In diesem Fall müsste man aber darauf achten, dass diese Hinweise nicht überhand nehmen, da sie immer andere Elemente überdecken können und dadurch mitunter auch störend wirken.

- Um das Verständnis zu erhöhen, was genau bei einem Klick auf einen Tag in der Navigationsleiste passiert, wäre es vorteilhaft, den Übergang zu diesem Tag animiert darzustellen. Wichtig wäre dabei aber, dass dieser Übergang nicht zu lange dauert, um die Handhabung dadurch nicht behäbig werden zu lassen.
- Der aktuelle Tag sollte unter Umständen noch deutlicher hervorgehoben werden, da er von mehreren Personen anfangs erst gesucht werden musste.
- Der Kalender sollte optisch etwas überarbeitet werden. Da die Optik von Lightning als deutlich ansprechender empfunden wurde, könnte man sich diesen Kalender als Vorbild nehmen. In Lightning sind beispielsweise die Rahmenlinien in helleren Farben gehalten und die Termine an den Ecken etwas abgerundet, wogegen TaBaKa sicher relativ kantig wirkt.

Daneben sollten auch technische Verbesserungen des aktuellen Status vorgenommen werden. Insbesondere wäre eine Überarbeitung bezüglich Browserkompatibilität notwendig, da der aktuelle Prototyp aus Zeitgründen ausschließlich für die aktuelle Firefox-Version entwickelt wurde. In der derzeitigen Implementierung wird außerdem in vielen Situationen die komplette Seite neu geladen, was programmiertechnisch eher unelegant ist und unter bestimmten Umständen zu Verzögerungen führen könnte.

Abschließend könnte man damit beginnen die restliche Funktionalität des Low-Fidelity-Prototypen zu implementieren. Beginnen sollte man hierbei am besten mit den Funktionen zum Erstellen und nachträglichen Bearbeiten von Terminen, sodass TaBaKa grundsätzlich tauglich für den Praxiseinsatz wird.

Als weitere Usability-Maßnahmen empfehlen sich Settings, die in dieser Arbeit noch nicht abgedeckt wurden. Dazu gehören insbesondere der Test auf Smartphone-Displays sowie der Test von eng verbuchten Terminkalendern. Da klassische Kalender einen großen Bekanntheitsvorteil haben und eine Einarbeitung in TaBaKa notwendig ist, wäre auch eine Langzeitstudie interessant, um zu beobachten, ob die Testergebnisse und vor allem die Bewertungen nach einer längeren Gewöhnungsphase besser werden.

In einem fortgeschrittenen Stand der Entwicklung könnte es sich auch lohnen über die Erweiterung des Systems um eine Aufgabenliste nachzudenken, da eine solche auch von vielen Konkurrenzprodukten mitgeliefert wird.

#### 14 Fazit

Die klassische Darstellung von Kalendern, in denen unterschiedliche Ansichten auf die selben Daten bereitgestellt werden, bringt Nachteile mit sich, die im Usability-Test nachgewiesen werden konnten. Insbesondere fällt oft die Entscheidung schwer, welche Ansicht die geeignetste für die aktuell zu lösende Aufgabe

ist und auch das Mapping von Navigationselementen kann je nach gewählter Ansicht verwirrend sein. Für aufwändigere Aufgaben ist es in solchen Kalendern oft notwendig, auf zusätzliche Medien als Gedächtnisstützen zurückzugreifen – etwa, um sich bei einer Zählung Zwischenergebnisse zu notieren.

Der alternative Ansatz, der in dieser Arbeit in Form von TaBaKa als Prototyp umgesetzt wurde, zeigt eine klare Erleichterung bei der Lösung solcher aufwändiger Aufgaben und die Testpersonen wussten vor allem die freie Anpassbarkeit des angezeigten Zeitraumes und die effiziente Platznutzung zu schätzen.

Nichtsdestotrotz ist es wie erwartet schwierig Benutzer von etwas Neuem zu überzeugen, wenn die Alternative so bekannt und bewährt ist, dass sie schon als selbstverständlich gilt. Trotz ein paar guter Testergebnisse mit TaBaKa bevorzugt die Mehrheit der Testpersonen klassische Alternativen und an gemessenen Zeiten und Rückmeldungen ist klar ersichtlich, dass man sich in TaBaKa aufgrund seiner Andersartigkeit erst einmal einarbeiten muss, um damit wirklich effektiv arbeiten zu können.

Leider stellte sich auch heraus, dass die Gegenmaßnahmen, die in TaBaKa gegen das Übersehen sehr später oder sehr früher Termine eingebunden sind, nicht wahrgenommen werden. Da sich die Benutzer des grundlegenden Problems aber offensichtlich bewusst sind, ist mitunter nur eine deutlichere Hervorhebung der entsprechenden Elemente notwendig, um potentielle Anwender zumindest in dieser Hinsicht von TaBaKa zu überzeugen.

Mit Abschluss dieser Arbeit liegt wie ursprünglich geplant ein Prototyp vor, der in weiterer Folge zu einem vollwertigen Kalender ausgebaut werden kann und die Ergebnisse des Usability-Tests geben die Richtung vor, in die Veränderungen vorgenommen werden sollten. Dazu zählen neben der Hervorhebung der schon genannten Funktionalitäten als wichtigstem Punkt auch das deutlichere Herausarbeiten von Navigationsalternativen, die keinen direkten Aufforderungscharakter haben, eine prägnantere Kennzeichnung des aktuellen Tages und die Einführung einer Möglichkeit zum Jahreswechsel ohne Ziffern eintippen zu müssen.

# 15 Literatur

- [1] Perlin, K., & Fox, D. (1993). Pad: An Alternative Approach to the Computer Interface. ACM SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, 57-64
- [2] Furnas, G. W., (1986). Generalized Fisheye Views. ACM CHI '86: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 16-23
- [3] Bederson, B. B., & Clamage, A, & Czerwinski, M. P., & Robertson, G. G. (2003). A Fisheye Calendar Interface for PDAs: Providing Overviews for Small Displays. ACM CHI EA '03: CHI '03 Extended Abstracts on Human Factors in Computing Systems, 618-619
- [4] Bederson, B. B., & Clamage, A, & Czerwinski, M. P., & Robertson, G. G. (2004). DateLens: A Fisheye Calendar Interface for PDAs. ACM Transactions on Computer-Human Interaction (TOCHI), 11, 90-119
- [5] Mackinlay, J. D., & Robertson, G. G., & DeLine, R. (1994). Developing Calendar Visualizers for the Information Visualizer. UIST '94 Proceedings of the 7th annual ACM symposium on User interface software and technology, 109-118
- [6] Kincaid, C. M., & Dupont, P. B., & Kaye. A. R. (1985). Electronic calendars in the office: An assessment of user needs and current technology. ACM Transactions on Information Systems (TOIS), 3(1), 89-102
- [7] Luz, S., & Masoodian, M. (2010). Improving focus and context awareness in interactive visualization of time lines. BCS '10 Proceedings of the 24th BCS Interaction Specialist Group Conference, 72-80
- [8] http://www.marco.org/2010/03/28/more-ideas-than-time-logarithmic-calendar-view (Juni 2013)
- [9] http://brooksreview.net/2010/09/sucky-calendars/ (Juni 2013)
- [10] http://www.mozilla.org/projects/calendar/ (Juni 2013)
- [11] https://addons.mozilla.org/de/thunderbird/addon/lightning/ (Juni 2013)
- [12] http://blog.lizardwrangler.com/2012/07/06/thunderbird-stability-and-community-innovation/ (Juni 2013)
- [13] http://eagereyes.org/blog/2008/fisheye-calendar-at-yahoo (Juni 2013)
- [14] http://mashable.com/2008/10/08/yahoo-calendar-beta/ (Juni 2013)
- [15] http://lifehacker.com/5061885/yahoo-calendar-beta-on-track-to-challenge-google-cal (Juni 2013)

# 16 Anhang

# 16.1 Beispielangabe für den Usability-Test

Auf den nachfolgenden Seiten befindet sich ein Angabetext wie er den Testpersonen ausgehändigt wurde. Im konkreten Beispiel handelt es sich um die Version, welche Testperson 3 vorgelegt wurde.

#### 16.1.1 Allgemeines

Vielen Dank für Ihre Bereitschaft an diesem Usability-Test teilzunehmen! Dieser Test soll dazu dienen, zwei verschiedene Kalenderdarstellungen miteinander zu vergleichen. Es geht dabei nicht darum Ihre Kompetenz als Benutzer zu testen, sondern darum die Tauglichkeit der beiden Kalender für alltägliche Aufgaben festzustellen. Nicht Sie werden getestet, sondern die Programme.

### 16.1.2 Demographische Daten

Bitte geben Sie vor weiteren Schritten hier ein paar allgemeine Daten zu Ihrer Person an! Alle Informationen werden vertraulich behandelt und nur in anonymisierter Form zu statistischen Zwecken weiterverarbeitet.

Geschlecht (bitte ankreuzen):

- o männlich
- o weiblich
- o keine Angabe

Alter:

Berufsstand (bitte ankreuzen, Mehrfachangabe erlaubt):

- o Schüler
- o Student
- o unselbstständig berufstätig
- o selbstständig berufstätig
- o nicht berufstätig

Falls Sie Schüler oder Student sind: Welche Fachrichtung studieren Sie (z.B. Medieninformatik) bzw. welche Schulform besuchen Sie (z.B. HTL Elektrotechnik)?

Falls Sie berufstätig sind: Als was sind Sie tätig (z.B. Programmierer)?

#### 16.1.3 Vorkenntnisse

Die nachfolgenden Fragen sollen dazu dienen die Ergebnisse des Tests anhand Ihrer Vorerfahrung besser beurteilen zu können.

Benutzen Sie Kalendersoftware?

- o Ja
- o Nein

Benutzen Sie Kalender auf Papier? (Taschenkalender, Wandkalender o.ä.)

- o Ja
- o Nein

Wenn Sie Kalendersoftware verwenden: Auf welchen Geräten nutzen Sie diese?

Gerät	nie	selten	manchmal	oft
PC/Mac	0	О	0	0
Laptop	0	0	0	0
Tablet	0	0	0	0
Smartphone	О	0	0	0

Wie vertraut sind Sie mit den folgenden Anwendungen?

Anwendung	unbekannt	namentlich bekannt	schon benutzt	regelmäßige Nutzung
Microsoft Outlook	О	O	О	О
Apple iCal / Apple Kalender	О	0	0	0
Mozilla Thunder- bird mit Lightning- Plugin	О	О	0	O
Mozilla Sunbird	0	0	О	0
Evolution	О	О	О	0
Kontact	О	0	О	О
TaBaKa	О	O	0	0

Wenn Sie Kalendersoftware mit Aufteilung in Tages-, Wochen- und Monatsansicht verwenden: Welche Ansicht benutzen Sie am Häufigsten ?

- o Tagesansicht
- o Wochenansicht
- o Monatsansicht

#### 16.1.4 Vorgehensweise

Auf den nachfolgenden Seiten werden Ihnen nun verschiedene Aufgaben gestellt, die unter Zuhilfenahme zweier unterschiedlicher Kalendersysteme am PC zu lösen sind. Es sind dabei zuerst Aufgaben in einem und dann in dem anderen System zu lösen. Die Aufgaben in beiden Kalendern sind dabei ähnlich, aber nicht identisch.

Alle gestellten Aufgaben sind jeweils unterhalb der Frage auf Papier zu beantworten. Bitte ändern Sie nichts am Datenbestand! Es sind für diesen Test keine Termine einzutragen oder zu bearbeiten.

Gehen Sie bitte alle Aufgabenstellungen der Reihe nach durch und blättern Sie erst weiter, wenn ein Blatt vollständig abgearbeitet ist!

Versetzen Sie sich bitte für den gesamten Test in folgendes Szenario:

Heute ist Mittwoch, der 29. April 2015. (Die Systemzeit des Testcomputers ist entsprechend eingestellt – Sie werden sich also in beiden Kalendern automatisch im richtigen Zeitintervall wiederfinden und müssen sich das Datum nicht merken, weil es im Kalender direkt abrufbar ist.)

Sie studieren Informatik und arbeiten nebenbei als Programmierer in einem Büro. In Ihrem Kalender befinden sich dadurch drei Arten von Terminen:

- Lehrveranstaltungen
- Prüfungen
- Bürozeiten

# 16.1.5 Kalender 1: Lightning (Thunderbird-Plugin)

Bei dem ersten zu testenden Kalender handelt es sich um Lightning, ein Plugin für den E-Mail-Client Thunderbird.

Sie haben hier die bekannten Ansichten für einen Tag, eine Woche und einen Monat zur Verfügung. Zusätzlich gibt es ein eine Ansicht, die sich »Mehrere Wochen« nennt. Diese sieht der Monatsansicht ähnlich, zeigt aber auch Zeiträume an, die über Monatsgrenzen hinausgehen.

Links oben befindet sich eine kleine Monatsübersicht, die in allen Ansichten erhalten bleibt.

Sie können selbst entscheiden, welche Ansichten Sie für die Lösung der Aufgaben nutzen und dürfen jederzeit zwischen diesen wechseln.

# Kalender 1 – Aufgabe 1: Orientierung

Beantworten Sie bitte die folgenden Fragen, indem Sie die entsprechenden Tage im Kalender nachschlagen:

- 1. Welches konkrete Datum hat der kommende Montag?
- 2. Welches konkrete Datum hatte der Donnerstag vorletzter Woche?
- 3. Welcher Wochentag war am ersten Jänner des laufenden Jahres?
- 4. Welches konkrete Datum hatte der letzte Freitag im Februar?
- 5. Welcher Wochentag war am 21.10.1992?

Springen Sie bitte abschließend wieder zum aktuellen Tag!

#### Kalender 1 – Aufgabe 2: Terminplanung

In den folgenden Fragen sollen Sie Termine planen. Tragen Sie diese aber bitte nicht in den Kalender ein, sondern schreiben Sie nur wie angegeben den jeweils ausgewählten Tag unter die Angabe!

- 1. Ihr Arbeitgeber benötigt an einem der nächsten zehn Tage (also bis zum 9. Mai) eine Sonderschicht von Ihnen innerhalb der Bürozeiten (Mo. bis Fr. von 08:00 bis 17:00). An welchem dieser Tage hätten Sie grundsätzlich noch die meiste unverplante Zeit innerhalb der Bürozeiten zu Verfügung? (Muss nicht am Stück sein.) Bitte beachten Sie dabei, dass der 1. Mai ein Feiertag ist! (Feiertage sind aus technischen Gründen in diesem Test nicht im Kalender eingetragen.)
- 2. Im August wollen Sie für fünf Tage zu einer großen Veranstaltung nach Deutschland fahren. Gehen Sie davon aus, dass Sie frühmorgens losfahren und spätabends zurück kommen diese fünf Tage müssen also komplett frei sein! Wann wäre der frühest mögliche Termin für die Abfahrt, sodass es zu keinen Überschneidungen mit anderen Terminen kommt?

Springen Sie bitte abschließend wieder zum aktuellen Tag!

# Kalender 1 – Aufgabe 3: Terminanalyse

Für eine Gehaltsverhandlung mit Ihrem Arbeitgeber wollen Sie sich einen Überblick verschaffen, wie oft Sie im ersten Quartal (= Jänner bis März) des Jahres von Ihren üblichen Arbeitszeiten abgewichen sind. Üblicherweise planen Sie immer Arbeitszeiten von 08:00 bis 13:00. Zählen Sie alle Tage, an denen es Abweichungen von diesem Zeitrahmen gegeben hat und notieren Sie hier die Gesamtsumme!

Der Test des ersten Kalenders ist nun abgeschlossen.

#### 16.1.6 Kalender 2: TaBaKa

Bei dem zweiten zu testenden Kalender handelt es sich um TaBaKa, eine webbasierte Anwendung.

Alle Tage werden hier auf einer fortlaufenden Leiste dargestellt, die Sie im unteren Fensterbereich sehen. Oberhalb wird jener Bereich im Detail angezeigt, der aktuell in der Leiste ausgewählt ist. Der erste im Detailbereich angezeigte Tag verfügt dabei immer über eine besonders große Spaltenbreite.

Sie können in der Leiste jederzeit ganz unten die beiden Grenzen des Fokus-Bereiches verschieben, um einen größeren oder kleineren Zeitraum im Detail zu betrachten.

# Kalender 2 – Aufgabe 1: Orientierung

Beantworten Sie bitte die folgenden Fragen, indem Sie die entsprechenden Tage im Kalender nachschlagen:

- 1. Welches konkrete Datum hat der kommende Dienstag?
- 2. Welches konkrete Datum hatte der Mittwoch vorletzter Woche?
- 3. Welcher Wochentag war am 31. Jänner des laufenden Jahres?
- 4. Welches konkrete Datum hatte der erste Montag im März?
- 5. Welcher Wochentag war am 14.03.1987?

Springen Sie bitte abschließend wieder zum aktuellen Tag!

#### Kalender 2 – Aufgabe 2: Terminplanung

In den folgenden Fragen sollen Sie Termine planen. Tragen Sie diese aber bitte nicht in den Kalender ein, sondern schreiben Sie nur wie angegeben den jeweils ausgewählten Tag unter die Angabe!

- 1. Für eine Lehrveranstaltung müssen Sie gemeinsam mit mehreren Kollegen in den nächsten zehn Tagen eine Laborübung absolvieren. Das Labor ist jeweils Montag bis Freitag von 08:00 bis 17:00 Uhr geöffnet. Sie wollen Ihren Kollegen Terminvorschläge machen, aber dabei auf jeden Fall den Tag ausschließen, der bei Ihnen ohnehin schon am dichtesten verplant ist. An welchem der zehn Tage haben Sie am wenigsten freie Zeit zu Verfügung?
- 2. Im Juli wollen Sie für vier Tage mit einem Bekannten nach Salzburg fahren. Gehen Sie davon aus, dass Sie frühmorgens losfahren und spätabends zurück kommen diese vier Tage müssen also komplett frei sein! Wann wäre der frühest mögliche Termin für die Abfahrt, sodass es zu keinen Überschneidungen mit anderen Terminen kommt?

Springen Sie bitte abschließend wieder zum aktuellen Tag!

# Kalender 2 – Aufgabe 3: Terminanalyse

Im Büro wird Ihnen Ihre Unflexibilität vorgeworfen, weil Sie immer nur Arbeitszeiten von 08:00 bis 13:00 Uhr wahrnehmen. Sie wissen aber, dass diese Anschuldigungen nicht stimmen und Sie vor allem am Ende der letzten Sommerferien und am Beginn des darauf folgenden Semesters oft von Ihren üblichen Zeiten abgewichen sind. Zählen Sie alle Tage, an denen es in den Monaten September bis November des Vorjahres Abweichungen von diesem Zeitrahmen gegeben hat und notieren Sie hier die Gesamtsumme!

Der Test des zweiten Kalenders ist nun abgeschlossen.

# 16.1.7 Vergleich der beiden Kalender

Sie haben nun in zwei unterschiedlichen Kalendersystemen sehr ähnliche Aufgaben gelöst. Geben Sie zu den nachfolgenden Aspekten bitte an, in welchem Kalender Sie diese als besser gelöst empfinden:

Eigenschaft	Lightning	beide gleich	TaBaKa
Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	О	0	О
Alltagstauglichkeit	О	0	О
Navigation	0	0	О
Verständlichkeit	О	0	О
Übersichtlichkeit	О	0	О
Einfachheit	О	0	О
Optik/Ästhetik	0	O	О
Zentrierung auf das Wesentliche	О	0	О
Effiziente Nutzung des vorhandenen Platzes	О	O	О
Bewahrung vor Fehlern	О	О	О
Flexibilität	О	0	О
Kennzeichnung unterschiedlicher Terminkategorien	О	О	0

Mit diesen Angaben ist der Test abgeschlossen. Vielen Dank für Ihre Teilnahme!

# 16.2 Detailergebnisse des Usability-Tests

Auf den nachfolgenden Seite sind die detaillierten Ergebnisse und Notizen des Usability-Tests einsehbar. Die demographischen Daten sind dabei bewusst nicht angegeben, weil diese aufgrund der Rekrutierung in meinem Bekanntenkreis dazu beitragen könnten, dass einzelne Ergebnisse zu konkreten Personen zurückverfolgt werden könnten, was der im Test zugesicherten Anonymität widersprechen würde.

#### 16.2.1 Erwartete Antworten

Auf die einzelnen Übungen wurden folgende Antworten erwartet:

Aufgabe	Antwort
1.1A.	05. Mai
1.1B.	04. Mai
1.2A.	16. April
1.2B.	15. April
1.3A.	02. März
1.3B.	27. Februar
1.4A.	Donnerstag
1.4B.	Samstag
1.5A.	Samstag
1.5B.	Mittwoch
2.1A.	05. Mai
2.1B.	06. Mai
2.2A.	11. Juli
2.2B.	05. August
3.1A.	8
3.1B.	8

# 16.2.2 Testperson 1 (Pilottest)

Vorkenntnisse

Nutzt Kalendersoftware	Ja
Nutzt Papierkalender	Ja
PC/Mac	oft genutzt
Laptop	oft genutzt
Tablet	nie genutzt
Smartphone	oft genutzt
Microsoft Outlook	schon benutzt
Apple iCal/Kalender	schon benutzt
Mozilla Thunderbird + Lightning	namentlich bekannt
Mozilla Sunbird	namentlich bekannt
Evolution	unbekannt
Kontact	unbekannt
TaBaKa	unbekannt
Meistgenutzte Ansicht	Wochenansicht

Antworten

Kalender	Aufgabe	Antwort	Korrektheit
Lightning	1.1A.	28. Apr. 2015	falsch
Lightning	1.2A.	23. Apr. 2015	falsch
Lightning	1.3A.	2. März 2015	richtig
Lightning	1.4A.	Donnerstag	richtig
Lightning	1.5A.	Samstag	richtig
Lightning	2.1A.	Dienstag 5. Mai	richtig
Lightning	2.2A.	11-14. Juli 2015	richtig
Lightning	3.1A.	7	falsch
TaBaKa	1.1B.	04. Mai 2015	richtig
TaBaKa	1.2B.	15. April 2015	richtig
TaBaKa	1.3B.	27. Februar	richtig
TaBaKa	1.4B.	Samstag	richtig
TaBaKa	1.5B.	Mittwoch	richtig
TaBaKa	2.1B.	Mittwoch	richtig
TaBaKa	2.2B.	5. August	richtig
TaBaKa	3.1B.	8	richtig

# Vergleich der beiden Kalender

Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	beide gleich
Alltagstauglichkeit	Lightning
Navigation	TaBaKa
Verständlichkeit	Lightning
Übersichtlichkeit	beide gleich
Einfachheit	beide gleich
Optik/Ästhetik	beide gleich
Zentrierung auf das Wesentliche	beide gleich
Effiziente Nutzung des vorhandenen Platzes	TaBaKa
Bewahrung vor Fehlern	beide gleich
Flexibilität	beide gleich
Kennzeichnung unterschiedlicher Terminkategorien	beide gleich

# Allgemeine Notizen zum Test:

Obwohl ich absichtlich nie gesagt hatte, dass einer der beiden Kalender von mir ist, kam irgendwann die Frage »Was ist eigentlich deins?«, die ich nicht beantwortete.

Notizen und gemessene Zeiten für Kalender 1

Aufgabe	Notizen	benötigte Minuten
Lightning 1.1A.	direkt über Navigationskalender abgelesen, falscher Tag (davor statt danach)	
Lightning 1.2A.	direkt über Navigationskalender abgelesen	3
Lightning 1.3A.	blättert über Navigationskalender zurück	
Lightning 1.4A.	blättert über Navigationskalender zurück; Nachfrage, was mit »Wochentag« gemeint ist	
Lightning 1.5A.	erst Aussage »Bäh!«, dann wie vorgesehen über Navigationskalender zurücknavigiert; am Ende aktuellen Tag gesucht – war wegen 2015 verwirrt	
Lightning 2.1A.	blättert vor und zurück; »Ich rate mal«; Unklarheit, ob VO gestrichen werden soll; alle 10 Tage?	6
Lightning 2.2A.	Mehrere Wochen, blättert 1-2 Mal vor, dann doch über Navigationskalender	
Lightning 3.1A.	Monatsansicht, über Navigationskalender zurückgeblättert, Wechsel zu Wochenansicht; hat bei einem Termin gemerkt, dass es unterhalb noch weitergeht und scrollte danach in mehreren Wochen ein bisschen auf und ab, hörte aber nach ein paar Wochen wieder damit auf; zählt mit Hilfsstrichen auf Papier	3

Für alle Aufgaben aus Abschnitt 1 wurde nur der Navigationskalender genutzt.

Notizen und gemessene Zeiten für Kalender 2

Aufgabe	Notizen	benötigte Minuten
TaBaKa 1.1B.	direkt abgelesen	
TaBaKa 1.2B.	Formulierung »vorletzter Woche« unklar; erst Klick auf Button, dann Schieben, dann wieder Klicken	3
TaBaKa 1.3B.	durch Verschieben der Leiste gefunden	
TaBaKa 1.4B.	durch Verschieben der Leiste gefunden	
TaBaKa 1.5B.	geht zurück zu Heute; »Kann man irgendwo das Jahr manipulieren?«; sucht erst unten, findet es dann aber gleich oben	
TaBaKa 2.1B.	fragt »Wo ist heute?«, obwohl die Funktion davor schon genutzt wurde; schiebt nur, aber vergrößert Rahmen nicht; versucht zu zoomen (Strg + Mausrad), um das angezeigte Intervall zu ändern; findet Einschränkung der Uhrzeit und setzt diese auf 05:00 bis 20:00; Unklarheit, was genau verlangt ist; nutzt Tooltips; »Tag ausschließen« – dann ist Mittwoch klar	8
TaBaKa 2.2B.	Wechselt über Monat; verwechselt Angabe erst mit der von Kalender 1 (4 Tage); »Wo kann ich mir den Monat anzeigen lassen?«; sucht Monatsansicht; blättert & scrollt vor & zurück; »Gibt nur Wochenansicht«	
TaBaKa 3.1B.	über Monat + Klick zum 01.09., dann Klick jeweils auf den Tag nach der Markierung; war erst im falschen Jahr, bemerkte Fehler aber selbst; Strichliste	4

Testperson 1 bezeichnete TaBaKa in Bezug auf die Navigation als intuitiver, Zitat: »Man klickt da drauf, was man verändern will.«

Das Springen per Klick in der Navigationsleiste wurde als schwer zu verstehen empfunden. Ursprünglich glaubte die Testperson dadurch jeweils eine Kalenderwoche in den Fokus zu laden.

# 16.2.3 Testperson 2

Vorkenntnisse

Vorkenntnisse	
Nutzt Kalendersoftware	Ja
Nutzt Papierkalender	Nein
PC/Mac	oft
Laptop	oft
Tablet	nie
Smartphone	oft
Microsoft Outlook	schon benutzt
Apple iCal/Kalender	namentlich bekannt
Mozilla Thunderbird + Lightning	schon benutzt
Mozilla Sunbird	namentlich bekannt
Evolution	unbekannt
Kontact	unbekannt
TaBaKa	unbekannt
Meistgenutzte Ansicht	Wochenansicht

Antworten

Kalender	Aufgabe	Antwort	Korrektheit
TaBaKa	1.2B.	15. April 2015	richtig
TaBaKa	1.1A.	5. Mai 2015	richtig
TaBaKa	1.3A.	2. März 2015	richtig
TaBaKa	1.4B.	Samstag	richtig
TaBaKa	1.5A.	Samstag	richtig
TaBaKa	2.2A.	4. Juli 2015	falsch
TaBaKa	2.1B.	6. Mai 2015	richtig
TaBaKa	3.1B.	6	falsch
Lightning	1.2A.	16. April 2015	richtig
Lightning	1.1B.	4. Mai 2015	richtig
Lightning	1.3B.	27. Feb. 2015	richtig
Lightning	1.4A.	Donnerstag	richtig
Lightning	1.5B.	Mittwoch	richtig
Lightning	2.2B.	5. August 2015	richtig
Lightning	2.1A.	5. Mai 2015	richtig
Lightning	3.1A.	8	richtig

# Vergleich der beiden Kalender

Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	Lightning
Alltagstauglichkeit	Lightning
Navigation	Lightning
Verständlichkeit	Lightning
Übersichtlichkeit	Lightning
Einfachheit	Lightning
Optik/Ästhetik	Lightning
Zentrierung auf das Wesentliche	beide gleich
Effiziente Nutzung des vorhandenen Platzes	beide gleich
Bewahrung vor Fehlern	beide gleich
Flexibilität	beide gleich
Kennzeichnung unterschiedlicher Terminkategorien	beide gleich

Notizen und gemessene Zeiten für Kalender  $1\,$ 

Aufgabe	Notizen	benötigte Minuten
TaBaKa 1.2B.	sucht Button links, verschiebt linke Grenze bis an Rand	
TaBaKa 1.1A.	direkt abgelesen	5
TaBaKa 1.3A.	Monat umgestellt und in Einzelschritten über Button zurück geklickt	
TaBaKa 1.4B.	Über Monat auf Februar, dann über Button um einen Tag zurück	
TaBaKa 1.5A.	ОК	
TaBaKa 2.2A.	Markierung für späten Termin übersehen	2
TaBaKa 2.1B.	ОК	
TaBaKa 3.1B.	Ansteuerung OK, klickt sich über Button einzeln vor; vergrößert Bereich; klickt am Ende auf 29.10. in der Leiste	3

Navigierte in Ta<br/>Ba Ka fast ausschließlich über die Buttons, die aber wegen eines Bugs nicht immer funktionierten.

Notizen und gemessene Zeiten für Kalender 2

Aufgabe	Notizen	benötigte Minuten
Lightning 1.2A.	direkt in Navigationskalender (hält Maus darüber)	
Lightning 1.1B.	direkt in Navigationskalender (ohne Maus)	$oxed{2}$
Lightning 1.3B.	über Navigationskalender zurückgeblättert	
Lightning 1.4A.	über Navigationskalender vorgeblättert	
Lightning 1.5B.	direkt über Navigationskalender	
Lightning 2.2B.	über Navigationskalender; fetten Tag angeklickt und so Termin außerhalb gefunden	3
Lightning 2.1A.	Mehrere Wochen; hält Maus für Details über Termine; nach Aufschreiben der Lösung noch einmal kontrolliert	
Lightning 3.1A.	Monatsansicht; Navigation über Navigationskalender; Kontrolle der Termine per Hover	2

Sehr routinierter Umgang mit Lightning.

# 16.2.4 Testperson 3

Vorkenntnisse

Vorkenntnisse	
Nutzt Kalendersoftware	Ja
Nutzt Papierkalender	Nein
PC/Mac	nie
Laptop	manchmal
Tablet	manchmal
Smartphone	oft
Microsoft Outlook	namentlich bekannt
Apple iCal/Kalender	namentlich bekannt
Mozilla Thunderbird + Lightning	namentlich bekannt
Mozilla Sunbird	unbekannt
Evolution	namentlich bekannt
Kontact	unbekannt
TaBaKa	unbekannt
Meistgenutzte Ansicht	Wochenansicht

Antworten

Kalender	Aufgabe	Antwort	Korrektheit
Lightning	1.1B.	4.5.2015	richtig
Lightning	1.2A.	16.4.2015	richtig
Lightning	1.4A.	Do.	richtig
Lightning	1.3B.	27.2.2015	richtig
Lightning	1.5B.	Mi	richtig
Lightning	2.1A.	5.5.2015	richtig
Lightning	2.2B.	5.8.2015	richtig
Lightning	3.1A.	8 7	falsch
TaBaKa	1.1A.	5.5.2015	richtig
TaBaKa	1.2B.	15.4.2015	richtig
TaBaKa	1.4B.	Sa	richtig
TaBaKa	1.3A.	2.3.2015	richtig
TaBaKa	1.5A.	Мо	falsch
TaBaKa	2.1B.	6.5.2015	richtig
TaBaKa	2.2A.	4.7.2015	falsch
TaBaKa	3.1B.	8	richtig

### Vergleich der beiden Kalender

Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	beide gleich
Alltagstauglichkeit	TaBaKa
Navigation	Lightning
Verständlichkeit	Lightning
Übersichtlichkeit	Lightning
Einfachheit	Lightning
Optik/Ästhetik	beide gleich
Zentrierung auf das Wesentliche	TaBaKa
Effiziente Nutzung des vorhandenen Platzes	beide gleich
Bewahrung vor Fehlern	TaBaKa
Flexibilität	Lightning
Kennzeichnung unterschiedlicher Terminkategorien	TaBaKa

### Allgemeine Notizen zum Test:

Allgemeine Tauglichkeit kommt auf Aufgabe an – manche in TaBaKa, andere in Lightning besser zu lösen; bzgl. Optik: Sliden/Animation wäre in beiden Kalendern gut;

Notizen und gemessene Zeiten für Kalender 1

Aufgabe	Notizen	benötigte Minuten
Lightning 1.1B.	Monatsansicht; beschwert sich, weil in Monatsansicht beim Scrollen immer gleich um eine ganze Seite ge- scrollt wird	
Lightning 1.2A.	Wechselt zu »Mehrere Wochen«	3
Lightning 1.4A.	Klickte erst auf Monat in Detailansicht; war sich unsicher, worauf sich die Pfeile zum Blättern beziehen	
Lightning 1.3B.	OK	
Lightning 1.5B.	Jahr über Navigationskalender geändert (Maus davor kurz darüber gehalte und Scrollrad versucht)	
Lightning 2.1A.	Wechselt zwischen Wochen- und Monatsansicht; unsicher, welche Ansicht besser; geht letztolich in Wochenansicht durch und scrollt immer ein bisschen hinauf und hinunter	4
Lightning 2.2B.	erst über Navigationskalender etwas nach vorne geblättert, dann direkt Monat geändert	
Lightning 3.1A.	Blättert durch Navigationskalender; sucht Suchfunktion; findet in Suchfunktion keinen passenden Zeitrahmen; wechselt zu Monatsansicht und hält Maus für Details über einzelne Termine; blättert mit Scrollrad vor; wechselt gegen Ende noch einmal in Wochenansicht und kommt letztendlich auf 8 Termine, wiederholt die Zählung dann aber nur in der Wochenansicht und kommt nur noch auf 7	6

Blättern in Lightning teilweise irreführend.

Notizen und gemessene Zeiten für Kalender 2

Aufgabe	Notizen	benötigte Minuten
TaBaKa 1.1A.	direkt abgelesen	
TaBaKa 1.2B.	schiebt Balken, bis Tag sichtbar wird	$oxed{2}$
TaBaKa 1.4B.	wechselt Monat, schiebt und klickt	
TaBaKa 1.3A.	klickt in der Detailansicht auf den Tag	
TaBaKa 1.5A.	OK (musste nur Jahr eingeben), Heute-Link sofort gefunden	
TaBaKa 2.1B.	vergrößert angezeigten Zeitraum; scrollt etwas hinauf und hinunter und sagt: »Ich hoffe, ich sehe da alles«; meint, dass mehrere Tage gleich wären	4
TaBaKa 2.2A.	ändert Monat; ändert angezeigten Bereich auf 1 Monat; Markierung für späten Termin übersehen	
TaBaKa 3.1B.	ändert erst Monat, dann Jahr; passt Rahmen neu an Monat an; schiebt jeweils zum nächsten Monat und passt angezeigten Zeitraum an	2

Probierte schon vor dem Lesen der ersten Übungsangabe für Ta<br/>Baka die Anpassung des Zeitintervalls.

Stört sich daran, dass in TaBaKa keine Monatsansicht – meint aber auch, dass diese im Alltag wahrscheinlich nicht so wichtig ist; Eintippen der Jahreszahl in TaBaKa ist mühsam; TaBaKa ist ohne Anleitung schwieriger verständlich; hat erst nach dem Test beim Herumprobieren die Markierung für frühere und spätere Termine entdeckt

# 16.2.5 Testperson 4

Vorkenntnisse

Nutzt Kalendersoftware	Nein
Nutzt Papierkalender	Ja
PC/Mac	-
Laptop	-
Tablet	-
Smartphone	-
Microsoft Outlook	namentlich bekannt
Apple iCal/Kalender	namentlich bekannt
Mozilla Thunderbird + Lightning	namentlich bekannt
Mozilla Sunbird	unbekannt
Evolution	unbekannt
Kontact	unbekannt
TaBaKa	unbekannt
Meistgenutzte Ansicht	-

Kalender	Aufgabe	Antwort	Korrektheit
TaBaKa	1.2A.	16. April 2015	richtig
TaBaKa	1.1A.	5. Mai 2015	richtig
TaBaKa	1.3A.	2. März 2015	richtig
TaBaKa	1.4A.	Donnerstag	richtig
TaBaKa	1.5B.	Mittwoch	richtig
TaBaKa	2.2B.	Mi 5 Mo 10. August 2015	richtig
TaBaKa	2.1B.	Mittwoch 6.	richtig
TaBaKa	3.1B.	8 Tage	richtig
Lightning	1.2B.	15. April 2015	richtig
Lightning	1.1B.	4. Mai 2015	richtig
Lightning	1.3B.	27. Februar 2015	richtig
Lightning	1.4B.	Samstag	richtig
Lightning	1.5A.	Samstag	richtig
Lightning	2.2A.	11 14. Juli 2015	richtig
Lightning	2.1A.	Dienstag 5. Mai	richtig
Lightning	3.1A.	8 Tage	richtig

# Vergleich der beiden Kalender

Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	beide gleich
Alltagstauglichkeit	beide gleich
Navigation	beide gleich
Verständlichkeit	Lightning
Übersichtlichkeit	beide gleich
Einfachheit	Lightning
Optik/Ästhetik	beide gleich
Zentrierung auf das Wesentliche	beide gleich
Effiziente Nutzung des vorhandenen Platzes	beide gleich
Bewahrung vor Fehlern	beide gleich
Flexibilität	TaBaKa
Kennzeichnung unterschiedlicher Terminkategorien	beide gleich

Notizen und gemessene Zeiten für Kalender  $1\,$ 

Aufgabe	Notizen	benötigte Minuten
TaBaKa 1.2A.	Fragt erst, welcher Tag heute ist – sieht dann oben in der Deatailansicht, dass ein Tag grau markiert ist; Klickt oben auf Tage; Klickt erst auf linke Zeitraumgrenze, geht dann über den Button nach links zurück	5
TaBaKa 1.1A.	Klickt Tag in Leiste an	
TaBaKa 1.3A.	navigiert erst über Button, meint dann aber, dass das auch schneller gehen muss und findet die Möglichkeit zum Umstellen des Monats	
TaBaKa 1.4A.	Monatswechsel + Klick auf den 1. in der Leiste	
TaBaKa 1.5B.	Ändert Jahr; überlegt; ändert Monat und sucht Tag in der Leiste	
TaBaKa 2.2B.	Stellt rechten Monat auf August um und geht einen Schritt vor; scrollt etwas hinauf und hinunter und findet dabei den Termin außerhalb	4
TaBaKa 2.1B.	Heute; Bemerkt Möglichkeiten zum Ziehen des Bal- kens und zum Ändern des Zeitbereiches; lässt sich richtigen Bereich anzeigen; scrollt hinauf und hinun- ter	
TaBaKa 3.1B.	Wechselt ins Jahr 2014; zieht Balken bis Anfang September; Klickt sich über Button vor; Setzt Zeitbereich auf ganzen September; zieht danach jeweils Balken zum nächsten Monat vor und passt Zeitraum an	2

TaBaKa sei gut, aber man müsse sich ein bisschen damit spielen; Hinweis zum Namen TaBaKa: »Baka« ist japanisch und bedeutet »Idiot«

Notizen und gemessene Zeiten für Kalender 2

Aufgabe	Notizen	benötigte Minuten
Lightning 1.2B.	direkt in Navigationskalender (hält Maus über Tag)	
Lightning 1.1B.	direkt in Navigationskalender (ohne Maus)	$oxed{2}$
Lightning 1.3B.	sucht Monat erst in Detailansicht	
Lightning 1.4B.	blättert über Navigationskalender zurück	
Lightning 1.5A.	ändert erst Jahreszahl, dann Monat im Navigationskalender; klickt auf Tag; zurück zum aktuellen Tag mit Heute-Button	
Lightning 2.2A.	Monatswechsel im Navigationskalender; Monatsansicht; ist erst verwundert, weil Detailansicht nicht automatisch zum geänderten Zeitraum des Navigationskalenders gewechselt hat – probiert herum, um die Detailansicht zu ändern und entfernt dabei auch kurzfristig den Haken bei »Privat« – findet letztendlich heraus, dass man einen Tag im Navigationskalender anklicken muss	4
Lightning 2.1A.	Zurück über Heute-Schaltfläche; geht erst zur Wochenansicht, dann zu Mehrere Wochen, nutzt Mouse-Over-Informationen	
Lightning 3.1A.	Wechselt Monat über Navigationskalender; blättert durch Navigationskalender; klickt einen Termin an und sagt kurz danach »Ah, nein«; sucht zusätzliche Einstellungen in Menüs, schaltet Tagesplan ein und wieder aus, deaktiviert und reaktiviert »Privat«; geht zu Wochenansicht, dann zu Mehrere Wochen und letztlich wieder zur Wochenansicht; geht Woche für Woche durch und klickt jeweils im Navigationskalender in die nächste Zeile, um eine Woche weiter zu kommen	3

In Lightning sei das einzelne Durchschauen mühsam.

# 16.2.6 Testperson 5

Vorkenntnisse

Vorkenntnisse	T
Nutzt Kalendersoftware	Ja
Nutzt Papierkalender	Nein
PC/Mac	oft
Laptop	manchmal
Tablet	nie
Smartphone	oft
Microsoft Outlook	schon benutzt
Apple iCal/Kalender	unbekannt
Mozilla Thunderbird + Lightning	schon benutzt
Mozilla Sunbird	ubekannt
Evolution	unbekannt
Kontact	unbekannt
TaBaKa	unbekannt
Meistgenutzte Ansicht	Monatsansicht

Antworten

Kalender	Aufgabe	Antwort	Korrektheit
Lightning	1.1B.	4.5.2010	richtig
Lightning	1.2B.	15.4.15	richtig
Lightning	1.4B.	Sa	richtig
Lightning	1.3B.	27	richtig
Lightning	1.5A.	Sa	richtig
Lightning	2.1A.	Di 5.5.15	richtig
Lightning	2.2A.	11.7.	richtig
Lightning	3.1A.	7	falsch
TaBaKa	1.1A.	5.5.15	richtig
ТаВаКа	1.2A.	16.5.15	richtig (Tag richtig, Monat falsch)
TaBaKa	1.4A.	Do	richtig
TaBaKa	1.3A.	2.3.15	richtig
TaBaKa	1.5B.	Mi	richtig
TaBaKa	2.1B.	6.5.15	richtig
TaBaKa	2.2B.	5.8 - 9.8	richtig
TaBaKa	3.1B.	8	richtig

# Vergleich der beiden Kalender

Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	Lightning
Alltagstauglichkeit	Lightning
Navigation	beide gleich
Verständlichkeit	beide gleich
Übersichtlichkeit	beide gleich
Einfachheit	TaBaKa
Optik/Ästhetik	Lightning
Zentrierung auf das Wesentliche	TaBaKa
Effiziente Nutzung des vorhandenen Platzes	TaBaKa
Bewahrung vor Fehlern	beide gleich
Flexibilität	Lightning
Kennzeichnung unterschiedlicher Terminkategorien	Lightning

# Notizen und gemessene Zeiten für Kalender 1

Aufgabe	Notizen	benötigte Minuten
Lightning 1.1B.	direkt abgelesen	
Lightning 1.2B.	direkt abgelesen	3
Lightning 1.4B.	über Navigationskalender zurückgeblättert	
Lightning 1.3B.	über Navigationskalender vorgeblättert	
Lightning 1.5A.	Jahr im Navigationskalender mit etwas nervösem Scrollen geändert; Monat im Navigationskalender geändert; wechselt über Kreissymbol zum aktuellen Tag und äußert dabei die Vermutung, dass dieses Symbol dazu da ist	
Lightning 2.1A.	klickt 8. im Navigationskalender an; Mehrere Wochen; stellt fest, dass der Kalender keine freie Wahl des Zeitintervalls hat; Wochenansicht; wechselt zwischen beiden Wochen und scrollt immer ein kleines Stück hinaus und hinunter; zählt freie Stunden	4
Lightning 2.2A.	Monatsansicht; blättert in Detailansicht vor	
Lightning 3.1A.	blättert in Detailansicht zurück; wechselt zu Wochenansicht und bezeichnet diese als »gescheiter«, weil man das Wesentliche dort auf einen Blick sieht; macht Strichliste	4

Notizen und gemessene Zeiten für Kalender 2

Aufgabe	Notizen	benötigte Minuten
TaBaKa 1.1A.	sagt »Der Strich wird heute markieren«	
TaBaKa 1.2A.	über Button zurück	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$
TaBaKa 1.4A.	ändert Monat, über Button zurück	
TaBaKa 1.3A.	OK; stellt zwischendurch Frage, ob der Kalender nur eine Wochenansicht hat	
TaBaKa 1.5B.	Klickt auf Jahreszahl und sagt »Bitte lass' dich ändern!« – drückt danach nicht Enter, sondern ändert gleich den Monat - dadurch erst im falschen Jahr – bemerkt und korrigiert es aber gleich	
TaBaKa 2.1B.	Klickt erst auf Grenze von Intervall, danach auf 6.	3
TaBaKa 2.2B.	Wechselt Monat; Klick auf 1.; sagt erst »eh keine Termine«, scrollt dann aber noch hinauf und hinunter und entdeckt dabei den Spättermin (Hinweis gesehen?); entdeckt auch Anpassung des Zeitraumes	
TaBaKa 3.1B.	ändert Jahr (diesmal mit Enter); ändert rechten Monat; setzt Zeitraum auf September und danach jeweils auf nächsten Monat; schreibt sich zur Hilfe Zahl je Monat zum Addieren auf	2

Bemerkte beim ersten Blick auf TaBaKa, dass es merkwürdig sei, dass der Kalender in der Angabe als webbasierend beschrieben wird, aber lokal läuft; erprobt Navigation vor Abarbeitung der Angabe mit einem Klick auf einen Tag.

Die Zentrierung auf das Wesentliche ist zwar bei TaBaKa stärker, aber eher im Negativen zu verstehen (zu reduziert); Die Kennzeichnung der Terminkategorien sei in Lightning besser, weil dort auch unterschiedliche Kalender unterschiedbar sind; An TaBaKa ist vor allem die freie Zeitraumwahl gut, sonst ist er eher nicht so gut.

# 16.2.7 Testperson 6

Vorkenntnisse

Vorkenntnisse	
Nutzt Kalendersoftware	Ja
Nutzt Papierkalender	Nein
PC/Mac	oft
Laptop	oft
Tablet	nie
Smartphone	oft
Microsoft Outlook	regelmäßige Nutzung
Apple iCal/Kalender	namentlich bekannt
Mozilla Thunderbird + Lightning	unbekannt
Mozilla Sunbird	unbekannt
Evolution	unbekannt
Kontact	unbekannt
TaBaKa	unbekannt
Meistgenutzte Ansicht	Wochenansicht

Antworten

Kalender	Aufgabe	Antwort	Korrektheit
TaBaKa	1.1A.	5.5.	richtig
TaBaKa	1.2A.	16.4.	richtig
TaBaKa	1.3B.	27.2	richtig
TaBaKa	1.4B.	Donnerstag	falsch
TaBaKa	1.5A.	Samstag	richtig
TaBaKa	2.2B.	3.8 7.8.	falsch
TaBaKa	2.1A.	4.5.	falsch
TaBaKa	3.1B.	3	falsch
Lightning	1.1B.	01.05	falsch
Lightning	1.2B.	15.04	richtig
Lightning	1.3A.	02.03	richtig
Lightning	1.4A.	Donnerstag	richtig
Lightning	1.5B.	Mittwoch	richtig
Lightning	2.2A.	11 14.07	richtig
Lightning	2.1B.	04.05	falsch
Lightning	3.1A.	(nicht beendet)	(nicht beendet)

### Vergleich der beiden Kalender

Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	TaBaKa
Alltagstauglichkeit	TaBaKa
Navigation	TaBaKa
Verständlichkeit	TaBaKa
Übersichtlichkeit	TaBaKa
Einfachheit	TaBaKa
Optik/Ästhetik	Lightning
Zentrierung auf das Wesentliche	Lightning
Effiziente Nutzung des vorhandenen Platzes	TaBaKa
Bewahrung vor Fehlern	beide gleich
Flexibilität	beide gleich
Kennzeichnung unterschiedlicher Terminkategorien	TaBaKa

### Allgemeine Notizen zum Test:

Die Testperson war am Vorabend lange fort, musste am Tag des Tests früh auf und nahm erst am späten Nachmittag am Test teil. Eine entsprechende Konzentrationsschwäche war deutlich sichtbar.

Notizen und gemessene Zeiten für Kalender 1

Aufgabe	Notizen	benötigte Minuten
TaBaKa 1.1A.	erst unklar, welcher Tag heute ist, erkennt dann aber roten Pfeil als Indikator; ändert angezeigtes Tagintervall; erst nicht bewusst, dass wir im April sind und hält deshalb vorerst das Ergebnis für falsch	4
TaBaKa 1.2A.	schnelle Klicks auf Button	
TaBaKa 1.3B.	OK (nur Monat umgestellt und dann direkt abgelesen)	
TaBaKa 1.4B.	wechselt auf Monat Dezember (2015)	
TaBaKa 1.5A.	ОК	
TaBaKa 2.2B.	wechselt über rechten Monat; sucht farbliche Kennzeichnung im Navigationskalender, die angibt, wann Termine sind; passt Tagintervall an; übersieht Markierung; wechselt über Heute-Link zurück zum heutigen Tag	4
TaBaKa 2.1A.	hat noch einen ganzen Monat im Bereich und kann die Termine daher direkt ablesen	
TaBaKa 3.1B.	stellt angezeigten Bereich wieder auf 7 Tage; ändert erst Jahr und dann rechten Monat; stellt Tagintervall so um, dass fast die ganze Navigationsleiste abgedeckt wird; klickt auch länger auf Button und zieht Leiste, um vor zu scrollen; merkt, dass hintere Markierung beim Scrollen mitgeht, aber nicht, dass am anderen Ende Tage wieder aus der Markierung verschwinden und zählt deshalb am Ende nur im Bereich vom 17.10 bis zum 29.11.	1

## Notizen und gemessene Zeiten für Kalender 2

Aufgabe	Notizen	benötigte Minuten
Lightning 1.1B.	direkt abgelesen	
Lightning 1.2B.	direkt abgelesen	$oxed{2}$
Lightning 1.3A.	über Navigationskalender 1 Mal zurückgeblättert	
Lightning 1.4A.	Monat geändert und direkt abgelesen	
Lightning 1.5B.	Klickt auf Jahr und will tippen, stöhnt »ernsthaft?« als sie sieht, dass sie scrollen muss; wechselt Monat; zurück zu Heute über Kreis (wartet erst Tooltip ab)	
Lightning 2.2A.	Mehrere Wochen; Monatsansicht; scrollt mit Mausrad und ist erst verwirrt wegen des Scrollverhaltens	3
Lightning 2.1B.	Klickt erst auf Monat in Details, dann auf Heute; immer noch in Monatsansicht, muss erst den aktuellen Tag (ganz unten) suchen	
Lightning 3.1A.	wechselt Monat; Mehrere Wochen; Monat; sucht zusätzliche Einstellungen; bemerkt Kategoriefarbe; glaubt, dass jeder Termin einzeln geöffnet werden müsste und sucht weiter Zusatzfunktionen, um diesen Aufwand zu umgehen; gelangt durch die Suche in Aufgabenansicht und findet Kalender erst nach längerer Suche wieder; kapituliert schließlich, weil es zu mühsam wäre jeden Termin einzeln zu öffnen	6

# 16.2.8 Testperson 7

Vorkenntnisse

Nutzt Kalendersoftware	Nein
Nutzt Papierkalender	Ja
PC/Mac	-
Laptop	-
Tablet	-
Smartphone	-
Microsoft Outlook	schon benutzt
Apple iCal/Kalender	namentlich bekannt
Mozilla Thunderbird + Lightning	namentlich bekannt
Mozilla Sunbird	namentlich bekannt
Evolution	unbekannt
Kontact	unbekannt
TaBaKa	unbekannt
Meistgenutzte Ansicht	-

Antworten

Kalender	Aufgabe	Antwort	Korrektheit
Lightning	1.2B.	15. April 2015	richtig
Lightning	1.1B.	4. Mai 2015	richtig
Lightning	1.3A.	2. März 2015	richtig
Lightning	1.4A.	Donnerstag	richtig
Lightning	1.5B.	Mittwoch	richtig
Lightning	2.1B.	Montag, 4. Mai 2015	falsch
Lightning	2.2A.	SA 11. Juli - DI 14. Juli 2015	richtig
Lightning	3.1A.	6	falsch
TaBaKa	1.2A.	16. April 2015	richtig
TaBaKa	1.1A.	5. Mai 2015	richtig
TaBaKa	1.3B.	27. Februar 2015	richtig
TaBaKa	1.4B.	Samstag	richtig
TaBaKa	1.5A.	Samstag	richtig
TaBaKa	2.1A.	DI, 5. Mai 2015	richtig
TaBaKa	2.2B.	von MI 5. August	richtig
TaBaKa	3.1B.	7	falsch

# Vergleich der beiden Kalender

Allgemeine Tauglichkeit zur Lösung der gegebenen Aufgaben	Lightning
Alltagstauglichkeit	Lightning
Navigation	Lightning
Verständlichkeit	Lightning
Übersichtlichkeit	Lightning
Einfachheit	Lightning
Optik/Ästhetik	Lightning
Zentrierung auf das Wesentliche	Lightning
Effiziente Nutzung des vorhandenen Platzes	beide gleich
Bewahrung vor Fehlern	Lightning
Flexibilität	beide gleich
Kennzeichnung unterschiedlicher Terminkategorien	beide gleich

Notizen und gemessene Zeiten für Kalender 1

Aufgabe	Notizen	benötigte Minuten		
Lightning 1.2B.	direkt (Finger als Hilfe)			
Lightning 1.1B.	direkt (Finger als Hilfe) 3			
Lightning 1.3A.	über Navigationskalender 1 Seite zurück			
Lightning 1.4A.	über Navigationskalender zurückgeblättert			
Lightning 1.5B.	sucht; wechselt wieder zu Heute; wechselt Ansichten; wechselt Jahr und Monat in Navigationskalender; hält Maus über Datum in Navigationskalender; Zurück zu Heute über Kreis (hält Maus darüber, bis Tooltip erscheint)			
Lightning 2.1B.	Mehrere Wochen, »Kann man das irgendwie verschieben?« – »Nein?« – »Das ist dämlich«; scrollt und blättert in Detailansicht; ohne Tooltips, nur auf Anzahl der Termine (nicht auf Länge) geachtet	4		
Lightning 2.2A.	Monatswechsel; Monatsansicht; Heute-Link; blättert wieder vor; wechselt wieder ein Wochenansicht			
Lightning 3.1A.	Mehrere Wochen; Rechtsklick auf Datum in Details; Monatsansicht; blättert über Detailansicht zurück; Mehrere Wochen; überlegt; Nachfrage zum Verständnis der Angabe; Nutzt Tooltips; schreibt Rechnung auf; merkt nach 1 Mal scrollen und weiterzählen, dass bisherige Lösung falsch, weil nicht um einen Monat weitergescrollt; wechselt in Monatsansicht und beginnt neu	5		

Notizen und gemessene Zeiten für Kalender 2

Aufgabe	Notizen	benötigte Minuten
TaBaKa 1.2A.	schiebt Leiste	
TaBaKa 1.1A.	direkt abgelesen (Maus darüber gehalten)	$oxed{3}$
TaBaKa 1.3B.	Monatswechsel, schieben der Leiste, Klick auf Tag	
TaBaKa 1.4B.	linken Monat geändert	
TaBaKa 1.5A.	verärgert, weil man das Jahr händisch eintippen muss; ändert linken Monat; lobt Umsetzung der Heute-Funktion	
TaBaKa 2.1A.	passt Tagintervall an; bemerkt Verschieben des Zeitintervalls – führt aber zu bekanntem Bug	7
TaBaKa 2.2B.	ändert erst rechten Monat, dann probeweise linken Monat; zieht 1. in Fokus; markiert Monat; scrollt etwas auf und ab – bemerkt dabei die Markierungen für auswärtige Termine, versteht aber die Bedeutung nicht und hält die Maus immer zu kurz darüber, um den Tooltip zu sehen; ändert Zeitintervall	
TaBaKa 3.1B.	ändert erst Jahr, dann Monat; will Markierung weiter ziehen; ändert Zeitintervall; ändert Tagintervall auf mehr als 1 Monat, scrollt jeweils bis zum nächsten noch nicht enthaltenen Tag vor und reduziert Ende Nov. das Tagintervall wieder, sodass nur verbliebene Tage angezeigt	(nicht erfasst)

Kommentiert TaBaKa gleich auf den ersten Blick mit »Whoa«, »What The Fuck« und »Das ist ja komisch«. TaBaKa sei nicht so übersichtlich – bei dem anderen Kalender sei »alles auf einer Seite«.

Probiert schon vor der ersten Aufgabe ein wenig herum und bemerkt dabei Klick auf Button und Schieben der Leiste. Freie Intervallwahl gelobt.

# 16.3 Termine im Usability-Test

 $\operatorname{Im}$  Usability-Test wurden beide Kalender mit den nachfolgenden Terminen befüllt.

Datum	von	bis	Kategorie	Titel
03.09.2014	08:00	13:00	Arbeit	Büro
05.09.2014	08:00	13:00	Arbeit	Büro
09.09.2014	08:45	13:45	Arbeit	Büro
12.09.2014	08:00	13:00	Arbeit	Büro
17.09.2014	08:00	13:00	Arbeit	Büro
19.09.2014	09:00	14:00	Arbeit	Büro
22.09.2014	09:00	14:00	Arbeit	Büro
24.09.2014	09:00	14:00	Arbeit	Büro
24.09.2014	15:00	16:00	Arbeit	Büro
25.09.2014	08:00	13:00	Arbeit	Büro
01.10.2014	08:00	13:00	Arbeit	Büro
03.10.2014	08:00	13:00	Arbeit	Büro
08.10.2014	08:00	13:00	Arbeit	Büro
09.10.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
10.10.2014	08:00	13:00	Arbeit	Büro
13.10.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
15.10.2014	12:00	17:00	Arbeit	Büro
16.10.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
16.10.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
17.10.2014	08:00	13:00	Arbeit	Büro
20.10.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
21.10.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
22.10.2014	08:00	13:00	Arbeit	Büro
23.10.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
23.10.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
24.10.2014	08:00	14:00	Arbeit	Büro
27.10.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
28.10.2014	08:00	09:00	Prüfung	Zwischentest Mathematik
28.10.2014	09:00	12:00	Lehrveranstaltung	Mathematik 1
29.10.2014	08:00	13:00	Arbeit	Büro
30.10.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
30.10.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
31.10.2014	08:00	13:00	Arbeit	Büro
03.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
04.11.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
05.11.2014	08:00	13:00	Arbeit	Büro
06.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
06.11.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
07.11.2014	08:00	13:00	Arbeit	Büro
10.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
10.11.2014	14:00	15:00	Prüfung	Abgabegespräch Einführung in die technische Informatik
11.11.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
12.11.2014	07:30	15:00	Arbeit	Büro
13.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
13.11.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
14.11.2014	08:00	13:00	Arbeit	Büro
17.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
18.11.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
18.11.2014	12:00	16:00	Arbeit	Büro
19.11.2014	08:00	13:00	Arbeit	Büro

20 11 2014	11.00	14.00	T 1	T3: 6:1
20.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
20.11.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
21.11.2014	08:00	13:00	Arbeit	Büro
24.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
25.11.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
26.11.2014	08:00	13:00	Arbeit	Büro
27.11.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
27.11.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
28.11.2014	08:00	13:00	Arbeit	Büro
01.12.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
02.12.2014	08:00	09:00	Prüfung	Zwischentest Mathematik
02.12.2014	09:00	11:00	Lehrveranstaltung	Mathematik 1
03.12.2014	08:00	13:00	Arbeit	Büro
04.12.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
04.12.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
05.12.2014	08:00	13:00	Arbeit	Büro
09.12.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
10.12.2014	08:00	13:00	Arbeit	Büro
10.12.2014	14:00	15:00	Prüfung	Abgabegespräch Einführung in die technische Informatik
11.12.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
11.12.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
12.12.2014	08:00	13:00	Arbeit	Вйго
15.12.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
16.12.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
17.12.2014	08:00	13:00	Arbeit	Büro
18.12.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
18.12.2014	15:00	16:00	Lehrveranstaltung	Grundlagen methodischen Arbeitens
19.12.2014	08:00	13:00	Arbeit	Büro
22.12.2014	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
23.12.2014	08:00	11:00	Lehrveranstaltung	Mathematik 1
25.12.2014	15:00	16:30	Prüfung	Prüfung Grundlagen methodischen Arbeitens
02.01.2015	08:00	15:00	Arbeit	Büro
05.01.2015	08:00	13:00	Arbeit	Büro
07.01.2015	08:00	13:00	Arbeit	Вйго
08.01.2015	11:00	14:00	Lehrveranstaltung	Einführung in die technische Informatik
09.01.2015	08:00	13:00	Arbeit	Büro
13.01.2015	08:00	11:00	Lehrveranstaltung	Mathematik 1
14.01.2015	08:00	13:00	Arbeit	Вёго
15.01.2015	08:00	13:00	Arbeit	Büro
16.01.2015	08:00	13:00	Arbeit	Büro
19.01.2015	11:00	14:00	Prüfung	Prüfung Einführung in die technische Informatik
19.01.2015	15:00	18:00	Arbeit	Büro
20.01.2015	08:00	11:00	Lehrveranstaltung	Mathematik 1
21.01.2015	08:00	13:00	Arbeit	Büro
23.01.2015	08:00	13:00	Arbeit	Büro
26.01.2015	08:00	16:00	Arbeit	Büro
28.01.2015	08:00	13:00	Arbeit	Büro
30.01.2015	07:30	12:30	Arbeit	Büro
04.02.2015	08:00	13:00	Arbeit	Büro
04.02.2015	14:00	17:00	Prüfung	Prüfung Mathematik 1
06.02.2015	08:00	13:00	Arbeit	Büro
11.02.2015	08:00	13:00	Arbeit	Büro
11.02.2015	14:00	18:00	Arbeit	Büro
13.02.2015	08:00	13:00	Arbeit	Büro
17.02.2015	08:00	13:00	Arbeit	Büro
11.02.2013	00.00	10.00	1110010	24.0

20.02.2015	08:00	13:00	Arbeit	Büro
25.02.2015	08:00	13:00	Arbeit	Büro
27.02.2015	08:00	13:00	Arbeit	Büro
04.03.2015	08:00	13:00	Arbeit	Büro
06.03.2015	08:00	13:00	Arbeit	Büro
09.03.2015	08:00	13:00	Arbeit	Büro
10.03.2015	08:00	13:00	Arbeit	Büro
11.03.2015	08:00	14:30	Arbeit	Büro
12.03.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
13.03.2015	08:00	13:00	Arbeit	Büro
16.03.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
16.03.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
17.03.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
17.03.2015	13:00	15:00	Arbeit	Büro
18.03.2015	08:00	13:00	Arbeit	Büro
19.03.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
20.03.2015	08:00	13:00	Arbeit	Büro
23.03.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
23.03.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
24.03.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
24.03.2015	13:00	15:00	Arbeit	Büro
25.03.2015	08:00	13:00	Arbeit	Büro
26.03.2015	08:00	09:00	Prüfung	Zwischenprüfung Mathematik 2
26.03.2015	09:00	11:00	Lehrveranstaltung	Mathematik 2
27.03.2015	08:00	13:00	Arbeit	Büro
30.03.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
	14:00			
30.03.2015 31.03.2015	10:00	16:00 12:00	Lehrveranstaltung  Lehrveranstaltung	Objektorientierte Modellierung Statistik und Stochastik
01.04.2015	08:00	13:00	Arbeit	Büro
02.04.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
08.04.2015	08:00	13:00	Arbeit	Büro
09.04.2015	08:00	13:00	Arbeit	Büro
10.04.2015	08:00	13:00	Arbeit	Büro
13.04.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
13.04.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
14.04.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
14.04.2015	17:20	17:40	Prüfung	Abgabegespräch Objektorientierte Modellierung
15.04.2015	08:00	13:30	Arbeit	Büro
16.04.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
17.04.2015	08:00	13:30	Arbeit	Büro
20.04.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
20.04.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
21.04.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
22.04.2015	08:00	13:00	Arbeit	Büro
23.04.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
24.04.2015	08:00	13:00	Arbeit	Büro
27.04.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
27.04.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
28.04.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
29.04.2015	08:00	13:00	Arbeit	Büro
30.04.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
04.05.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
04.05.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
05.05.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
06.05.2015	08:00	13:30	Arbeit	Büro
				ı

07.05.2015	08:00	11:00	Lohmonanataltung	Mathematik 2
08.05.2015	08:00	13:00	Lehrveranstaltung Arbeit	Büro
11.05.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
11.05.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
13.05.2015	08:00	13:00	Arbeit Arbeit	Büro
15.05.2015	08:00	13:00		
18.05.2015	12:00	14:00	Lehrveranstaltung	Softskills für Informatiker
18.05.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
19.05.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
20.05.2015	08:00	13:00	Arbeit	Büro
21.05.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
22.05.2015	08:00	13:00	Arbeit	Büro
26.05.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
27.05.2015	08:00	13:00	Arbeit	Büro
28.05.2015	08:00	09:00	Prüfung	Zwischenprüfung Mathematik 2
28.05.2015	09:00	12:00	Lehrveranstaltung	Mathematik 2
29.05.2015	08:00	13:00	Arbeit	Büro
01.06.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
02.06.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
03.06.2015	08:00	13:00	Arbeit	Büro
05.06.2015	08:00	13:00	Arbeit	Büro
08.06.2015	14:00	16:00	Lehrveranstaltung	Objektorientierte Modellierung
09.06.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
10.06.2015	08:00	13:00	Arbeit	Büro
11.06.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
12.06.2015	08:00	13:00	Arbeit	Büro
16.06.2015	10:00	12:00	Lehrveranstaltung	Statistik und Stochastik
16.06.2015	15:30	15:50	Prüfung	Abgabegespräch Objektorientierte Modellierung
17.06.2015	08:00	13:00	Arbeit	Büro
18.06.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
19.06.2015	08:00	13:00	Arbeit	Büro
24.06.2015	08:00	13:00	Arbeit	Büro
25.06.2015	08:00	11:00	Lehrveranstaltung	Mathematik 2
26.06.2015	08:00	13:00	Arbeit	Büro
29.06.2015	14:00	16:00	Prüfung	Objektorientierte Modellierung
30.06.2015	10:00	11:00	Prüfung	Prüfung Statistik und Stochastik
01.07.2015	08:00	13:00	Arbeit	Büro
03.07.2015	08:00	13:00	Arbeit	Büro
06.07.2015	18:00	20:00	Prüfung	Prüfung Mathematik 2
08.07.2015	08:00	13:00	Arbeit	Büro
10.07.2015	08:00	13:00	Arbeit	Büro
15.07.2015	08:00	13:00	Arbeit	Büro
17.07.2015	08:00	13:00	Arbeit	Büro
22.07.2015				
22.01.2010	08:00	13:00	Arbeit	Büro
24.07.2015	08:00	13:00 13:00	Arbeit Arbeit	Büro Büro
24.07.2015	08:00	13:00	Arbeit	Büro
24.07.2015 29.07.2015	08:00 08:00	13:00 13:00	Arbeit Arbeit	Büro
24.07.2015 29.07.2015 31.07.2015	08:00 08:00 08:00	13:00 13:00 13:00	Arbeit Arbeit Arbeit	Büro Büro
24.07.2015 29.07.2015 31.07.2015 04.08.2015	08:00 08:00 08:00 18:00	13:00 13:00 13:00 20:30	Arbeit Arbeit Arbeit Arbeit	Büro Büro Büro
24.07.2015 29.07.2015 31.07.2015 04.08.2015 12.08.2015	08:00 08:00 08:00 18:00 08:00	13:00 13:00 13:00 20:30 13:00	Arbeit Arbeit Arbeit Arbeit Arbeit	Büro Büro Büro Büro
24.07.2015 29.07.2015 31.07.2015 04.08.2015 12.08.2015	08:00 08:00 08:00 18:00 08:00 08:00	13:00 13:00 13:00 20:30 13:00	Arbeit Arbeit Arbeit Arbeit Arbeit Arbeit	Büro Büro Büro Büro
24.07.2015 29.07.2015 31.07.2015 04.08.2015 12.08.2015 14.08.2015 19.08.2015	08:00 08:00 08:00 18:00 08:00 08:00	13:00 13:00 13:00 20:30 13:00 13:00	Arbeit Arbeit Arbeit Arbeit Arbeit Arbeit Arbeit	Büro           Büro           Büro           Büro           Büro           Büro
24.07.2015 29.07.2015 31.07.2015 04.08.2015 12.08.2015 14.08.2015 19.08.2015 21.08.2015	08:00 08:00 08:00 18:00 08:00 08:00 08:00	13:00 13:00 13:00 20:30 13:00 13:00 13:00	Arbeit Arbeit Arbeit Arbeit Arbeit Arbeit Arbeit Arbeit Arbeit	Büro           Büro           Büro           Büro           Büro           Büro           Büro

02.09.2015	08:00	13:00	Arbeit	Büro
04.09.2015	08:00	13:00	Arbeit	Büro
09.09.2015	08:00	13:00	Arbeit	Büro
11.09.2015	08:00	13:00	Arbeit	Büro
16.09.2015	08:00	13:00	Arbeit	Büro
18.09.2015	08:00	13:00	Arbeit	Büro
23.09.2015	08:00	13:00	Arbeit	Büro
25.09.2015	08:00	13:00	Arbeit	Büro
30.09.2015	08:00	13:00	Arbeit	Büro

#### Quelltext des High-Fidelity-Prototypen

#### HTML16.4.1

Listing 1: index.html

```
<head>
                                           <title>TaBaKa</title>
                                         <meta http-equive"ontent-type" content="text/html;_charset=ISO-8859-1" />
k rel="stylesheet" type="text/css" href="screen.css" />
                                         <script type="text/javascript" src="language.js"></script>
<script type="text/javascript" src="appointments.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></s
                                         <script type="text/javascript" src="initializer.js"></script>
<script type="text/javascript" src="format_helpers.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></scrip
 13
14
15
16
17
18
19
                                         <script type="text/javascript" src="main.js"></script>
<script type="text/javascript" src="outside_indicators.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script><
                                         <script type="text/javascript" src="navbar.js"></script>
<script type="text/javascript" src="today_indicator.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></
 20
21
22
23
24
25
                                         <script type="text/javascript" src="scaling.js"></script>
<script type="text/javascript" src="dragging.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script
                                   </head>
26
27
28
                                 <body>
                                       <script type="text/javascript">
initialize();
</script>
 \frac{29}{30}
\frac{31}{32}
\frac{33}{33}
                                         <!-- ====== Menu ======= -->
 \frac{34}{35}
                                         <div id="menu">
 36
37
38
39
40
                                         <!-- ====== Main Area ======= -
                                         <div id="calendararea" onscroll="scroll();">
<script type="text/javascript">
drawCols();
drawCalendarBase();
 42
 43
 44
45
46
47
                                       </script>
</div>
                                       <div id="starttimer" onmousedown="grabStarttimer();_return_false;">&#x25b2;&#
x25bc;</div>
<div id="endtimer" onmousedown="grabEndtimer();_return_false;">&#x25b2;&#x25b2;&#x25b2;&#x25b2;
                                                                                                                                                                                                                                      onmousedown="grabEndtimer(); _return_false; ">▲ ▼
48
                                                                                           </div>
                                         <div id="timerinfo"></div>
 50
51
52
53
54
55
                                           <!-- ====== Navigation Area ====== -->
                                         <div id="today">
<a href="javascript:_focusToday();" id="todaylink">&#x25bc;</a>
                                         <a hre </a>
 56
                                         57
 58
                                       60
61
                                           <div id="navbarcontainer">

  <script type="text/javascript">
  drawNavBar();
 66
                                         </ri>
</script>

</div>
 67
68
 69
70
71
72
73
74
75
76
77
78
79
80
81
                                            <div id="marker"></div>
                                         <canvas id="triangleleft" width="32" height="32"></canvas>
<script type="text/javascript">
var canvas = document.getElementById('triangleleft');
var context = canvas.getContext('2d');
                                                   context.beginPath();
context.moveTo(32, 0);
context.lineTo(32, 32);
context.lineTo(0, 0);
context.closePath();
```

```
context.fillStyle = 'black';
context.fill();
</script>
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
       <canvas id="triangleright" width="32" height="32"></canvas>
<script type="text/javascript">
  var canvas = document.getElementById('triangleright');
  var context = canvas.getContext('2d');
        context.beginPath();
context.moveTo(0, 0);
context.lineTo(0, 32);
context.lineTo(32, 0);
context.losePath();
       context.fillStyle = 'black';
context.fill();
</script>
98
99
100
101
102
103
       104
105
       107
108
        <script type="text/javascript">
109
110
111
112
113
114
115
        document.getElementById("today").title = LANG_CLICK_TODAY;
        scale();
window.onresize = scale;
        document.onmouseup = testReleasing;
116
117
        window.location.href="#time" + hourFirst;
118
119
      </body>
123
     </html>
124
```

#### 16.4.2 CSS

#### Listing 2: screen.css

```
margin-top: 119px;
overflow: auto;
border-width: 1px 0 1px 0;
border-style: solid;
border-color: black;
.hourborder {
width: 100%;
margin: 0;
border-width: 0 0 1px 0;
border-style: solid;
border-color: #ccc;
             #timeline {
  position: absolute;
  top: 0;
  width: 45px;
  background-color: white;
             #timeline .time {
  text-align: center;
}
            .yearheader, .monthheader, .dayheader, .dayappointment, .outsidetop, .outsidebottom {
   position: fixed;
   left: 45px;
   margin: 0;
   padding: 4px 0;
   height: 14px;
   border: 1px solid black;
   text-align: center;
   font-size: 12px;
   font-weight: normal;
   overflow: hidden;
   background-color: white;
}
              .yearheader, .monthheader, .dayheader { background-color: rgb(96%, 96%, 96%);
              .yearheader {
top: 30px;
padding: 2px 0;
height: 18px;
              .monthheader {
                top: 53px;
padding: 2px 0;
height: 18px;
100
101
102
103
104
              .dayheader {
top: 76px;
padding: 2px 0;
float: left;
height: 29px;
105
106
107
108
109
110
111
              .dayheader:first-line {
font-weight: bold;
112
113
114
115
116
117
118
119
               .today {
background-color: #ddd;
              .dayappointment {
  top: 110px;
  padding: 2px 0;
  float: left;
120
121
122
122
123
124
125
126
127
             .outsidetop {
top: 129px;
padding: 0;
float: left;
height: 10px;
font-size: 8px;
128
129
130
131
132
133
             .outsidebottom {
  padding: 0;
  float: left;
  height: 10px;
  font-size: 8px;
}
\frac{134}{135}
136
137
138
139
140
141
142
              .colback, .colline {
   position: absolute;
   height: 100%;
   border-width: 0 1px 0 1px;
   border-style: solid;
   border-color: black;
```

```
z-index: -1;
            .colback {
z-index: -1;
background-color: white;
149
150
151
151
152
153
154
155
156
          .appointment {
  position: absolute;
  border: 1px solid black;
  overflow: hidden;
}
\frac{157}{158}
           .style1 {
background-color: #fdd;
159
160
161
162
163
164
           .style1:hover {
background-color: #fcc;
\begin{array}{c} 165 \\ 166 \end{array}
           .style2 { background-color: #dfd;
\begin{array}{c} 167 \\ 168 \\ 169 \\ 170 \\ 171 \\ 172 \\ 173 \\ 174 \\ 175 \\ 176 \\ 177 \\ 178 \\ 179 \\ \end{array}
           . style2:hover {
 background-color: #cfc;
           .style3 {
  background-color: #ddf;
}
           .style3:hover { background-color: #ccf;
180
181
181
182
183
184
185
186
187
          #starttimer , #endtimer {
  position: absolute;
  width: 45px;
  height: 20px;
  text-align: center;
  color: #aaa;
  z-index: 50;
  cursor: ns-resize;
}
188
189
190
191
192
193
194
          #starttimer:hover, #endtimer:hover {
  color: black;
}
195
196
          #starttimer {
  top: 120px;
  border-bottom: 1px solid black;
197
198
199
200
201
202
203
204
           #endtimer {
  border-top: 1px solid black;
          #timerinfo {
display: none;
position: absolute;
left: 50px;
width: 45px;
height: 18px;
border: 1px solid black;
background-color: white;
text-align: center;
z-index: 50;
}
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
           /* _____ Navigation Area _____ */
         #navbarcontainer {
  overflow: hidden;
  margin: 0 45px;
}
220
221
221
222
223
224
225
226
          #navbar {
   position: relative;
   margin-top: 40px;
   height: 60px;
   border: 1px solid black;
   border-collapse: collapse;
   text-align: center;
   table-layout: fixed;
   overflow: hidden;
   background-color: white;
}
\frac{227}{228}
229
230
231
232
\frac{233}{234}
235
236
237
238
239
240
241
          #navbar th {
  border: 1px solid black;
  font-weight: normal;
}
          #navbar td {
  border: 1px solid black;
```

```
244 | width: 25px;
245 | font-size: 10px;
246 | }
247 | #navbar td:first-
          #navbar td:first-line {
  font-size: 12px;
}
249
          }
250
250
251
252
253
254
255
          #scrollleft, #scrollright {
  position: absolute;
  height: 64px;
  width: 45px;
  z-index: 1;
}
\begin{array}{c} 256 \\ 257 \end{array}
258
258
259
260
261
262
263
           button#scrollright {
  right: 0;
}
         #marker {
   position: absolute;
   left: 219px;
   height: 112px;
   width: 175px;
   background-color: black;
   opacity: 0.12;
}
\frac{264}{265}
266
267
268
269
270
271
272
273
274
275
276
277
278
279
          #triangleleft, #triangleright {
  position: absolute;
  height: 32px;
  opacity: 0.12;
}
         #overlaydragger {
  position: absolute;
  height: 22px;
  width: 100%;
  cursor: ew-resize;
}
280
281
281
282
283
284
285
286
          #overlayclicker {
  position: absolute;
  height: 38px;
  width: 100%;
  cursor: pointer;
}
287
288
289
289
290
291
292
293
294
          #rangeleft, #rangeright {
  position: relative;
  float: left;
  width: 41px;
  height: 20px;
  text-align: center;
  color: #aaa;
  cursor: ew-resize;
}
295
296
297
298
299
300
301
          #rangeleft:hover, #rangeright:hover {
  color: black;
}
302
303
304
305
306
307
308
309
          #rangeleft {
    left: 198px;
}
           #rangeright {
310
311
              left: 332px;
311
312
313
314
315
316
317
          #rangeleft div, #rangeright div {
  width: 20px;
  height: 20px;
  float: left;
}
318
319
          #rangeleft div:first-child, #rangeright div:first-child {
   border-right: 1px solid black;
320
321
322
323
324
325
          #today {
  position: absolute;
  width: 25px;
  z-index: 20;
  text-align: center;
326
327
           #today a {
  color: #c00;
  text-decoration: none;
\frac{332}{333}
334
335
336
337
338
339
340
                                     General Classes ====
              \label{eq:weekend:not(.style1):not(.style2):not(.style3) {background-color: } rgb(100\%, 97\%, 87\%); \\
```

#### 16.4.3 Initialisierungsdaten

#### Listing 3: language.js

Im nachfolgenden Code sind nur zwei Termine angegeben, um das Konzept nachvollziehen zu können. Die Originaldatei aus dem Usability-Test wäre mehrere Seiten lang.

### Listing 4: appointments.js

# Listing 5: initializer.js

```
10 var NAVBARCELLWIDTH = 25;
     var today = new Date();
var mainDay = new Date(today.getTime()); // day in focus
    var days = 7; // days to show at once in the main view var hourFirst = 8; // first hour to show without scrolling var hourLast = 17; // last hour to show without scrolling var shownHours = hourLast — hourFirst; var rangeLeft = 7; // days shown in navbar before main day
19
20
21
    23
    24
\frac{25}{26}
    var navbarXPosition = 0;
var navbarXPositionStart = 0;
var rangeleftXPosition = 0;
var rangeleftXPositionStart = 0;
var rangerightXPosition = 0;
var todayPosition = -999999;
\frac{27}{28}
29
30
31 \\ 32 \\ 33 \\ 34 \\ 35
     /* initialized values to build the calendar */ function initialize() \{
\frac{36}{37}
      \label{eq:check_if_parameter} \begin{subarrate}{ll} // & check & if & parameter & was & given \\ var & parameter & = & location.search.substr(location.search.indexOf("?") + 1); \end{subarrate}
     40
41
42
43
44
45
      } else {
  mainDay.setHours(12);
  mainDay.setMinutes(0);
  mainDay.setSeconds(0);
}
46
47
48
49
50
51
52
        // check for settings in cookies
53
54
55
56
57
58
59
60
61
      cookienames = new Array();
cookievalues = new Array();
      if (document.cookie) {
        cookiestring = document.cookie;
       days = cookiestring.substring(cookiestring.search("=") + 1, cookiestring.search(";
")) * 1;
62
        cookiestring = cookiestring.substring(cookiestring.search(";") + 1);
         \begin{array}{lll} range Left &= cookiestring.substring (cookiestring.search ("=") + 1, cookiestring. \\ &search (";")) * 1; \\ cookiestring &= cookiestring.substring (cookiestring.search (";") + 1); \end{array} 
65
66
         \begin{array}{lll} today Position = cookiestring.substring (cookiestring.search ("=") + 1, cookiestring.\\ search (";")) & * 1;\\ cookiestring = cookiestring.substring (cookiestring.search (";") + 1); \end{array} 
67
68
69
70
        \begin{array}{lll} hourFirst = cookiestring.substring(cookiestring.search("=") + 1, cookiestring.\\ search(";")) * 1; \end{array}
        search (";")) * 1; \\ cookiestring = cookiestring.substring (cookiestring.search (";") + 1); \\
71
72
73
74
75
76
77
78
79
        shown Hours \ = \ cookiestring \, . \, substring \, (\, cookiestring \, . \, search \, ("=" \, ) \ + \ 1) \ * \ 1;
        hourLast = hourFirst + shownHours;
        delse {
  document.cookie = "days=" + days + ";";
  document.cookie = "rangeLeft=" + rangeLeft + ";";
  document.cookie = "todayPosition=" + todayPosition + ";";
  document.cookie = "hourFirst=" + hourFirst + ";";
  document.cookie = "shownHours=" + shownHours + ";";
80
81
82
83
84
85
86
      87
88
89
      // set range rangeleft XPosition = 45 + rangeLeft * NAVBARCELLWIDTH - 22; rangeright XPosition = 45 + rangeLeft * NAVBARCELLWIDTH + days * NAVBARCELLWIDTH - \frac{1}{2}
\frac{92}{93}
94
95
96
97
       {\tt rangeleftXPositionStart\ =\ rangeleftXPosition}\;;
      // set width of first day in focus if ((document.documentElement.clientWidth - TIMEBARWIDTH - SCROLLBARWIDTH) / days > FIRSTDAYWIDTH)
```

```
100 FIRSTDAYWIDTH = (document.documentElement.clientWidth - TIMEBARWIDTH - SCROLLBARWIDTH) / days;
101
102 }
```

# 16.4.4 Hilfsfunktionen zur Formatierung

### Listing 6: format helpers.js

## 16.4.5 Hauptfunktionen

#### Listing 7: main.js

```
| /*
| creates and controls the main area
| creates and controls area
| creates and controls the main area
| creates and controls | creates area
| creates and creates area
| creates ar
```

```
53
 54
55
 56
 57
58
59
60
61
62
           }
document.write("</select>");
document.write("</h2>");
if (i != 0) {
  monthCounter = i - monthCounter;
  monthlengths.push(monthCounter);
  monthCounter = i;
 63
64
65
 66
67
68
69
70
71
72
73
74
75
76
77
78
80
81
82
83
84
85
          // \  \, draw \  \, column \, (s\,) \  \, for \  \, day \, (s\,)
          if (day.getDay() == 0 || day.getDay() == 6)
  weekendString = "_weekend";
            weekendString = "";
          if (today.getYear() == day.getYear() &&
    today.getMonth() == day.getMonth() &&
    today.getDate() == day.getDate()) {
    actualDay = true;
         selse {
  actualDay = false;
}
         86
87
88
89
90
91
92
           \begin{array}{c} ^{\backslash no>} + \\ ^{\prime\prime} </a>^{\prime\prime}); \\ document.write("<div\_class=\\ ^{\prime\prime} col\_dayappointment\_" + weekendString + "\">&nbsp;</div>^{\prime\prime}); \\ div>^{\prime\prime}); \\ \end{array} 
                                       "</h3>
         93
 94
 95
 96
 97
98
 99
100
101
106
         drawAppointments(day);
107
108
109
110
111
       }
        yearCounter = i - yearCounter;
yearlengths.push(yearCounter);
monthCounter = i - monthCounter;
monthlengths.push(monthCounter);
113
114
115
      /* draws appointments for a given day param day = the day to draw the appointments for */ function drawAppointments(day) \{
121
        \label{eq:formula} \textbf{for} \hspace{0.2cm} (\hspace{0.1cm} j \hspace{0.1cm} = \hspace{0.1cm} 0 \hspace{0.1cm} ; \hspace{0.2cm} j \hspace{0.1cm} < \hspace{0.1cm} APPOINTMENTS.length \hspace{0.1cm} ; \hspace{0.2cm} j \hspace{0.1cm} + \hspace{0.1cm} ) \hspace{0.2cm} \hspace{0.1cm} \{
122
123
         if (APPOINTMENTS[j]["start"].getYear() == day.getYear() &&
    APPOINTMENTS[j]["start"].getMonth() == day.getMonth() () &
    APPOINTMENTS[j]["start"].getDate() == day.getDate()) {
124
125
           APPOINTMENTS[j]["shown"] = true;
129
           document.\ write\ ("<\!div\_class=\!\! "appointment\_style"\ +\ APPOINTMENTS\ [j]\ ["style"]\ +\ "`"\_"
130
                                               131
                                                                                134
135
136
                                                                       LANG_END
137
                                                                                                                          "end"].getMinutes())
+ "\"=" +
```

```
139
140
143
           }
         }
148
149
        /* switches to another month param element = option field that contains the number of the new month param dayCounter = at which column the option field appears */ function monthSwitcher(element, dayCounter) {
150
151
151
152
153
154
155
156
          {\tt var dateNew = new Date(mainDay.getTime());}
         if (dayCounter == 1) {
   dateNew.setMonth(element.value);
} else {
   dateNew = new Date(dateNew.getTime() + dayCounter * DAYMILLISEC);
   dateNew.setMonth(element.value);
   dateNew = new Date(dateNew.getTime() - dayCounter * DAYMILLISEC);
}
\begin{array}{c} 157 \\ 158 \end{array}
159
160
161
162
163
          window.location.href = "index.html?" + dateNew.getFullYear() + "-" + leadZero(dateNew.getMonth()) + "-" + leadZero(dateNew.getDate());
165
        /* switches to another year
    param event = helper variable to get the pressed key
    param element = text field that contains the new year
    param dayCounter = at which column the text field appears
    param defaultYear = given year before changes */
function yearSwitcher(event, element, dayCounter, defaultYear) {
173
          if (event.keyCode != 13) return;
179
          // check if given year is a correct number
if (isNaN(element.value)) {
    alert(ERROR-YEAR);
    element.value = defaultYear;
            return;
          //\ switch\ to\ the\ given\ year
189
          var dateNew = new Date(mainDay.getTime());
          if (dayCounter == 1) {
  dateNew.setYear(element.value);
} else {
  dateNew = new Date(dateNew.getTime() + dayCounter * DAYMILLISEC);
  dateNew.setYear(element.value);
  dateNew = new Date(dateNew.getTime() - dayCounter * DAYMILLISEC);
}
195
196
197
198
199
200
201
```

### 16.4.6 Markierungen für Früh- und Spättermine

Listing 8: outside indicators.js

```
if (APPOINTMENTS[i]["shown"] > 0) {
22 \\ 23 \\ 24 \\ 25
           if (APPOINTMENTS[i]["start"].getHours() < hourFirst) {
  outsideTopAppointments.push(APPOINTMENTS[i]);
  APPOINTMENTS[i]["shown"] = 1;
} else if (APPOINTMENTS[i]["end"].getHours() > hourLast) {
  outsideBottomAppointments.push(APPOINTMENTS[i]);
  APPOINTMENTS[i]["shown"] = 1;
} else if (APPOINTMENTS[i]);
26
27
28
29
30
31
32
           } else {
APPOINTMENTS[i]["shown"] = 2;
           }
33
34
35
36
37
38
39
40
41
42
         }
       }
        var element;
var day = new Date(mainDay.getTime());
var ids = new Array();
        // deactivate all indicators for appointments outside the current time scope
43
44
45
46
47
48
49
       50
51
52
53
54
55
56
57
        58
59
60
61
62
        // activate indicators for appointments before the current scope of time for (i = outsideTopAppointments.length - 1; i >= 0; i--) { element = document.getElementById("outsidetop"+ outsideTopAppointments[i]["start" ].getFullYear() + "-" +
63
64
                                                                                              65
66
           element.innerHTML = "▲";
67
68
          if (element.className.substring(element.className.length - 6, element.className. length - 1) != "style") {
    element.className += "_style" + outsideTopAppointments[i]["style"];
    element.title = LANG_OUTSIDE;
69
70
71
72
73
74
75
76
        // activate indicators for appointments after the current scope of time
for (i = 0; i < outsideBottomAppointments.length; i++) {
    element = document.getElementById("outsidebottom"+ outsideBottomAppointments[i]["
        start"].getFullYear() + "-" +
                                                                                                     leadZero(outsideBottomAppointments[i]["
    start"].getMonth()) + "-" +
leadZero(outsideBottomAppointments[i]["
    start"].getDate()));
77
78
         start"].getDate()));

if (element.className.substring(element.className.length - 6, element.className.
    length - 1) != "style") {
    element.className += "-style" + outsideBottomAppointments[i]["style"];
    element.title = LANG_OUTSIDE;
}
79
80
81
82
83
84
85
86
```

### 16.4.7 Navigationsleiste

### Listing 9: navbar.js

```
1 /*
2 creates and controls the navigation bar
3 4 4 5 6 /* draws the navigation bar */
7 function drawNavBar() {
8 9  // collect data
10 1 var months = new Array(); // collects data about available months
12 var runningDay = new Date(mainDay.getTime()); // actual day to collect data from var lastDay = new Date(mainDay.getTime()); // helper to determine how long a month is
```

```
{\tt var\ tableWidth\ =\ 0\,;\ //\ width\ of\ the\ html\ table\ in\ days}
 15
16
17
18
         runningDay.setYear(runningDay.getFullYear() - 1);
runningDay.setDate(1);
 19
         20
 21
         var weekday = runningDay.getDay();
         for (i = 0; i < 25; i++) {
 \frac{24}{25}
           months[i] = new Object();
months[i]["month"] = runningDay.getMonth();
months[i]["year"] = runningDay.getFullYear();
 26
27
 28
 \frac{29}{30}
\frac{31}{32}
           if (months[i]["month"] != 11) {
  runningDay.setMonth(months[i]["month"] + 1);
          runningDay.setMonth(0);
runningDay.setYear(months[i]["year"] + 1);
 33
 34
 35
36
37
38
39
           \begin{split} & lastDay = new \ Date(runningDay.getTime() - DAYMILLISEC); \\ & months[i]["days"] = lastDay.getDate(); \\ & tableWidth += months[i]["days"]; \end{split}
 \frac{40}{41}
 43
44
45
46
47
48
49
         // draw navbar table
        50
51
52
53
54
55
         document.write("");
         document.write("");
for (month = 0; month < months.length; month++) {
  for (day = 1; day <= months[month]["days"]; day++) {</pre>
 56
 57
            if (months[month]["year"] == today.getFullYear() &&
    months[month]["month"] == today.getMonth() &&
    day == today.getDate()) {
    attributes = "_class=\"today\"";
} else if (weekday % 7 == 0 || weekday % 7 == 6) {
    attributes = "_class=\"weekend\"";
 60
61
62
 63
64
65
            } else
attributes = "";
 66
67
68
69
            document.write("<td" + attributes + ">" + day + "<br/>br_/>" + WEEKDAYS[weekday++ % 7] + "" );
 70
71
72
73
74
75
         document.write("");
         {\tt document.getElementById\,("navbar").style\,.width} \ = \ {\tt tableWidth} \ * \ {\tt NAVBARCELLWIDTH} \ + \ "px"
 76
77
78
79
80
81
82
83
84
85
         \label{eq:navbarXPosition} \begin{split} navbarXPosition = & ((actual Difference - range Left) * NAVBARCELLWIDTH - 5) * -1; \\ navbarXPositionStart = & navbarXPosition; \\ document.getElementById("navbar").style.left = & navbarXPosition + "px"; \\ \end{split}
         // set today marker
if (todayPosition != -999999) {
  checkTodayVisibility();
        } else {
todayPosition = rangeleftXPosition + NAVBARCELLWIDTH - 4;
checkTodayVisibility();
 86
 88
89
90
91
92
       /* scroll forward in the navbar */
function scrollForward() {
  navbarXPosition -= NAVBARCELLWIDTH;
  document.getElementById("navbar").style.left = navbarXPosition + "px";
  todayPosition -= NAVBARCELLWIDTH;
  checkTodayVisibility();
}
 93
 95
96
97
98
99
100
       /* scroll back in the navbar */
function scrollBack() {
  navbarXPosition += NAVBARCELLWIDTH;
  document.getElementById("navbar").style.left = navbarXPosition + "px";
  todayPosition += NAVBARCELLWIDTH;
  checkTodayVisibility();
101
107
108
```

```
/* loads a new day into the focus depending on the position of the navbar */ function loadDay() {
  114
115
  var dateNew = new Date();
  if (rangeleftXPosition != rangeleftXPositionStart) {
  120
121
122
123
  } else {
  DAYMILLISEC);
129
  133
134
135
136
```

# 16.4.8 Markierung des aktuellen Tages

Listing 10: today indicator.js

```
/st handles the indiator which shows the current day
     /* set focus on today */
function focusToday() {
  document.cookie = "todayPosition=-999999;";
  window.location.href = "index.html";
     /* check if today marker is visible and adjust it's appearance */function checkTodayVisibility() \{
      // left outside the view if (todayPosition < 45 - NAVBARCELLWIDTH) {
18
19
       \mathbf{if} \hspace{0.2cm} (\hspace{0.1cm} \mathtt{document.getElementById} \hspace{0.1cm} (\hspace{0.1cm} \mathtt{modaylink"} \hspace{0.1cm}) \hspace{0.1cm} .\hspace{0.1cm} \mathtt{innerHTML.length} \hspace{0.1cm} = \hspace{0.1cm} 1) \hspace{0.2cm} \hspace{0.1cm} \{
20
21
         {\tt document.getElementById}~("~todaylink"~)~. \\ {\tt innerHTML}~=~"\&\#x25c4~; \_"~+~LANG\_TODAY~+~":\_"~
22
23
                                                                                       today.getDate() + "._" +
MONTHNAMES[today.getMonth()] + "
                                                                                       today.getFullYear();
24
\begin{array}{c} 25 \\ 26 \\ 27 \\ 28 \\ 29 \\ 30 \\ 31 \\ 32 \\ 33 \\ 34 \\ 35 \\ 36 \\ 37 \\ 38 \end{array}
         document.getElementById("today").style.width = "150px";
document.getElementById("today").style.left = "45px";
document.getElementById("today").style.textAlign = "left";
      // right outside the view } else if (todayPosition > document.documentElement.clientWidth - 45){
         if \ (document.getElementById ("todaylink").innerHTML.length == 1) \ \{ \\
         39
                                                                                       _" + today.getFullYear() + "_►";
40
41
42
43
44
         45 - 130 + "px";
document.getElementById("today").style.textAlign = "right";
45
46
47
48
49
50
51
52
      // inside the view } else {
        {\tt document.getElementById("today").style.left = todayPosition + "px";}
53
        if (document.getElementById("todaylink").innerHTML.length != 1) {
```

```
document.getElementById("todaylink").innerHTML = "▼";
document.getElementById("today").style.width = NAVBARCELLWIDTH + "px";
document.getElementById("today").style.textAlign = "center";
}

60
61
62
63
}
```

#### 16.4.9 Ziehen von Elementen

# Listing 11: dragging.js

```
var element = ""; // name of the element to drag
    var xStart = -1;

var xBefore = -1;

var x = -1;
    var mode = "none"; // = none, click or drag
13
14
15
16
17
18
    \begin{array}{lll} var & yStartValue = -1; \\ var & yBefore = -1; \\ var & y = -1; \end{array}
    var yStarttimer = 120;
var yEndtimer = 141 + calendarareaHeight;
\frac{20}{21}
    var newHour = hourFirst;
    /******* DRAGGING HORIZONTAL
26
27
28
29
30
31
32
    /{*} \ \ dragging \ \ elements \ \ horizontally \ \ */function \ \ dragHorizontal(event) \ \ \{
     if (xStart == -1) {
      xStart = event.pageX;
xBefore = event.pageX;
x = event.pageX;
33
34
35
36
37
38
39
40
41
42
      // dragging the navigation bar
       if (element == "navbar") {
       xBefore = x;
x = event.pageX;
43
44
45
46
47
48
49
        navbarXPosition += x - xBefore;
document.getElementById("navbar").style.left = navbarXPosition + "px";
        todayPosition += (x - xBefore);
checkTodayVisibility();
50
51
52
53
54
55
56
57
58
       // dragging the left range limiter
      } else if (element == "rangeleft" && event.pageX >= TIMEBARWIDTH && event.pageX <= rangerightXPosition + 65 - NAVBARCELLWIDTH * 2) {
        \begin{array}{lll} {\tt xBefore} & = & {\tt x}\,; \\ {\tt x} & = & {\tt event.pageX}\,; \end{array}
        59
60
61
62
63
       64
65
        document.getElementById("triangleleft").style.left = rangeleftXPosition + 20 - triangleLeftWidth - 1 + "px";
66
       // dragging the right range limiter
68
69
70
71
72
73
74
75
76
77
       } else if (element == "rangeright" && event.pageX <= document.documentElement.clientWidth - TIMEBAR event.pageX >= rangeleftXPosition + 20 + NAVBARCELLWIDTH * 2)
                                                                                                     TIMEBARWIDTH &&
        xBefore = x;
x = event.pageX;
        rangerightXPosition += x - xBefore;
```

```
79
        82
 83
                                                                                                 triangleRightPosition
                                                                                                SCROLLBARWIDTH
        + "px";
document.getElementById("triangleright").style.left = triangleRightPosition + "px
 85
 86
       }
 88
89
90
91
92
93
94
      }
    /* grabbing the navigation bar */
function grabNavbar() {
  element = "navbar";
  mode = "drag";
  document.body.onmousemove = dragHorizontal;
 95
96
97
98
99
100
101
     /* releasing the navigation bar */ function release Navbar() {
     if (mode == "drag") {
       if (mode == "drag") {
document.body.onmousemove = void(0);
element = "";
mode = "none";
108
       \label{eq:mode} mode = "none"; \\ navbarXPosition += NAVBARCELLWIDTH / 2; // \textit{for a more intuitive mapping}
109
110
       loadDay();
     } else {
 clickedNavbar();
}
115
    }
116
117
    124
125
126
    /* activated when navigation bar was clicked, not dragged */function clickedNavbar() {  document.onclick = focusClickedDay; 
131
132
133
134
135
136
    /* click on the navigation bar */
function clickNavbar() {
  element = "navbar";
  mode = "click";
138
139
140
140
141
142
143
144
145
    /* change the mode from klick to change if the navbar is dragged */function changeMode() {    if (mode == "click") grabNavbar();
146
147
148
    /* grabbing the left range limiter */
function grabRangeleft() {
  element = "rangeleft";
  mode = "drag";
  document.body.onmousemove = dragHorizontal;
149
150
151
152
153
154
155
156
    161
162
163
165
                                                           NAVBARCELLWIDTH) + 3) + ";";
      loadDay();
\frac{167}{168}
169
170
```

```
/* grabbing the right range limiter */
function grabRangeright() {
  element = "rangeright";
  mode = "drag";
  document.body.onmousemove = dragHorizontal;
175
176
177
        /* releasing the right range limiter */
function releaseRangeright() {
  document.body.onmousemove = void(0);
  clowent = "".
182
          element = ""
mode = "none
183
          mode - none;
document.cookie = "days=" + (Math.floor((rangerightXPosition - rangeleftXPosition)
184
185
                                                                                              NAVBARCELLWIDTH) + 2) + ";";
         loadDay();
        190
191
        /* dragging elements vertically */function dragVertical(event) {
195
         if (yStartValue == -1) {
196
197
          yStartValue = event.pageY;
yBefore = event.pageY;
y = event.pageY;
198
199
200
201
202
203
         } else {
           // handling the start time limiter
204
205
           if (element == "starttimer" && event.pageY > 10) {
206
207
208
209
210
             yBefore = y;
y = event.pageY;
yStarttimer += y - yBefore;
document.getElementById(element).style.top = yStarttimer + "px";
document.getElementById("timerinfo").style.top = yStarttimer + "px";
211
212
213
             if (y > yStartValue) {
  newHour = hourFirst + Math.round((y - yStartValue) / hourborderHeight + 1) - 1;
} else if (hourFirst > 0) {
  interval = 100 / (hourFirst - 1);
  newHour = hourFirst - Math.floor((yStartValue - y) / interval);
  if (newHour < 0) newHour = 0;
}</pre>
214
214
215
216
217
218
219
220
221
222
             {\tt document.getElementById} \, (\, \tt{"timerinfo"} \, ) \, . \, \\ {\tt innerHTML} \, = \, \, \\ {\tt hourFormat} \, (\, \tt{newHour} \, ) \, ; \\ \\
223
224
225
226
            //\ \ handling\ \ the\ \ end\ \ time\ \ limiter
            } if (element == "endtimer" & & event.pageY < document.documentElement.clientHeight
227
             yBefore = y;
y = event.pageY;
yEndtimer += y - yBefore;
document.getElementById(element).style.top = yEndtimer + "px";
document.getElementById("timerinfo").style.top = yEndtimer + "px";
228
229
230
231
232
             if (y < yStartValue) {
    newHour = hourLast - Math.round((yStartValue - y) / hourborderHeight + 1) + 1;
} else if (hourLast < 24) {
    interval = 90 / (24 - hourLast);
    newHour = hourLast + Math.floor((y - yStartValue) / interval);
    if (newHour > 24) newHour = 24;
}
233
234
235
236
237
238
239
240
241
             document.getElementById("timerinfo").innerHTML = hourFormat(newHour);
242
243
244
245
246
247
248
         }
249
250
       /* grabbing the start time limiter */
function grabStarttimer() {
  element = "starttimer";
   document.body.onmousemove = dragVertical;
  document.getElementById("timerinfo").style.display = "block";
  document.getElementById("timerinfo").style.top = "120px";
  document.getElementById("timerinfo").innerHTML = hourFormat(hourFirst);
251
251
252
253
254
255
\frac{256}{257}
258
259
260
261
262
263
        /* releasting the start time limiter */
function releaseStarttimer() {
   document.body.onmousemove = void(0);
          element = "";
document.getElementById("timerinfo").style.display = "none";
document.cookie = "hourFirst=" + newHour + ";";
264
265
```

```
\label{eq:condition} \begin{array}{lll} \mbox{document.cookie} &= "\mbox{shownHours}=" & + \mbox{ (hourLast} & - \mbox{newHour}) & + \mbox{";";} \\ \mbox{loadDay();} & & & \end{array}
271
            /* grabbing the end time limiter */
function grabEndtimer() {
  element = "endtimer";
  document.body.onmousemove = dragVertical;
  document.getElementById("timerinfo").style.display = "block";
  document.getElementById("timerinfo").style.top = 141 + calendarareaHeight + "px";
  document.getElementById("timerinfo").innerHTML = hourFormat(hourLast);
273
278
279
280
            /* releasing the end time limiter */
function releaseEndtimer() {
   document.body.onmousemove = void(0);
   element = "";
   document.getElementById("timerinfo").style.display = "none";
   document.cookie = "shownHours=" + (newHour - hourFirst) + ";";
   loadDay();
}
281
281
282
283
284
285
286
287
288
289
290
291
292
             293
           /* tests if there is something to release and releases the dragged element if there is one */
function testReleasing() {
    if (element == "starttimer")
        releaseStarttimer();
    else if (element == "endtimer")
        releaseEndtimer();
    else if (element == "rangeleft")
        releaseRangeleft();
    else if (element == "rangeright")
    releaseRangeright();
    else if (element == "navbar")
    releaseNavbar();
}
295
300
```

### 16.4.10 Skalieren der Ansicht

#### Listing 12: scaling.js

```
/*
scales all elements depending on available room
*/
    /* scales the calendar horizontally depending on available room */ function scaleHorizontal() \{
     // day columns
     {
m var} colElementGroups = 6; // number of different element groups that use the CSS-
15
16
     17
18
19
20
21
22
     }
23 \\ 24 \\ 25 \\ 26
     elements = document.getElementsByClassName("appointment");
28
     for (i = 0; i < elements.length; i++) {
29
      id \ = \ elements \left[ \ i \ \right]. \ id. \ substring \left( 11 \right);
31
32
33
34
35
36
      if (APPOINTMENTS[id]["start"].getYear() == mainDay.getYear() &&
APPOINTMENTS[id]["start"].getMonth() == mainDay.getMonth() &&
APPOINTMENTS[id]["start"].getDate() == mainDay.getDate()) {
  elements[i].style.width = FIRSTDAYWIDTH - 1 + "px";
  elements[i].style.left = TIMEBARWIDTH + "px";
37
      } else {
   var appointmentTime = new Date(APPOINTMENTS[id]["start"].getTime());
   appointmentTime.setHours(12); // make sure that the day difference always uses
38
39
           40
\frac{41}{42}
```

```
}
 \begin{array}{c} 46\\ 47\\ 48\\ 49\\ 50\\ 51\\ 52\\ 53\\ 45\\ 55\\ 60\\ 61\\ 62\\ 63\\ 64\\ 65\\ 66\\ 69\\ 70\\ 77\\ 78\\ 79\\ 81\\ 82\\ 83\\ 84\\ 85\\ \end{array}
        }
         // years
         elements \ = \ document.\,getElementsByClassName("yearheader");
       // months
         {\tt elements} \ = \ document.\, {\tt getElementsByClassName} \, (\, "\, monthheader"\, ) \; ;
         \begin{aligned} & \textbf{for } ( \text{i} = 0; \text{ i} < \text{elements.length}; \text{ i} + + ) \mid \{ \\ & \text{elements}[\text{i}]. \text{style.width} = \text{monthlengths}[\text{i}] * (\text{colWidth} + 1) - 1 + \\ & \text{(i} == 0 ? \text{FIRSTDAYWIDTH} - \text{colWidth} - 1 : 0) + "px"; \\ & \text{elements}[\text{i}]. \text{style.left} = \text{TIMEBARWIDTH} + \text{offset} * (\text{colWidth} + 1) + \\ & \text{(i} != 0 ? \text{FIRSTDAYWIDTH} - \text{colWidth} - 1 : 0) + "px"; \\ & \text{offset} += \text{monthlengths}[\text{i}]; \end{aligned} 
       }
       /*\ scales\ the\ calendar\ ver\\ function\ scaleVertical()\ \{
                                                  vertically depending on available room */
         calendararea Height = document.document Element.client Height - 264;\\ hourborder Height = calendararea Height / (hour Last - hour First) - 1;\\
        \label{eq:continuous} \begin{array}{lll} & \text{document.getElementById("calendararea").style.height = calendarareaHeight + "px";} \\ & \text{document.getElementById("timeline").style.height = (hourborderHeight + 1) * 24 + "px";} \\ & \text{px";} \end{array}
  86
 87
        \label{eq:className} \begin{array}{ll} elements = document.getElementsByClassName("colback");\\ \textbf{for} \ (i = 0; \ i < elements.length; \ i++) \ \{\\ elements[i].style.height = (hourborderHeight + 1) * 24 + "px"; \} \end{array}
 88
89
90
91
92
93
         elements = document.getElementsByClassName("colline");
for (i = 0; i < elements.length; i++) {
  elements[i].style.height = (hourborderHeight + 1) * 24 + "px";
}</pre>
 94
95
96
97
98
99
         elements = document.getElementsByClassName("outsidebottom");
for (i = 0; i < elements.length; i++) {
   elements[i].style.top = 141 + calendarareaHeight + "px";</pre>
100
101
102
103
104
         = 141 + calendarareaHeight + 39
        document.getElementById("scrollright").style.top
+ "px";
105
                                                                                                          = 141 + calendarareaHeight + 39
106
107
         document.getElementById("marker").style.top
                                                                                                            = 141 + calendarareaHeight + 11
         + "px";
document.getElementById("triangleleft").style.top
108
         document, getElementById("triangleright").style.top = 141 + calendarareaHeight + 11
109
         document.getElementById("overlaydragger").style.top = 141 + calendarareaHeight + 41
110
         + "px";

document.getElementById("overlayclicker").style.top = 141 + calendarareaHeight + 63 + "px";
111
113
        document.getElementById("endtimer").style.top
                                                                                                            = 141 + calendarareaHeight
                            + "px";
        115
                                                                                                           = 141 + calendarareaHeight + 25
         elements = document.getElementsByClassName("hourborder");
for (i = 0; i < elements.length; i++) {
   elements[i].style.height = hourborderHeight + "px";</pre>
120
121
         elements = document.getElementsByClassName("time");
elements[0].style.marginTop = (hourborderHeight) / 2 - 1 + "px";
for (i = 0; i < elements.length; i++) {
   elements[i].style.height = (hourborderHeight + 1) + "px";
   elements[i].style.lineHeight = (hourborderHeight + 1) + "px";</pre>
122
123
124
125
126
128
        // appointments
129
```

```
{\tt minuteHeight = hourborderHeight / 60;}
      {\tt elements} \ = \ {\tt document.getElementsByClassName("appointment")} \ ;
      for (i = 0; i < elements.length; i++) {
135
136
       id = elements[i].id.substring(11);
137
138
139
140
141
       yStart = (APPOINTMENTS[id]["start"].getHours() * 60 +
APPOINTMENTS[id]["start"].getMinutes()) *
minuteHeight +
APPOINTMENTS[id]["start"].getHours() - 1;
\frac{142}{143}
       yEnd = (APPOINTMENTS[id]["end"].getHours() * 60 + APPOINTMENTS[id]["end"].getMinutes()) * minuteHeight +
144
145
146
147
148
149
150
151
152
                     minuteHeight +
APPOINTMENTS[id]["start"].getHours() - 1;
       elements[i].style.top = yStart + "px";
elements[i].style.height = yHeight + "px";
153
154
155
156
157
158
159
160
161
162
163
164
165
     /* rescales elements */ function scale() \{
166
      document.getElementById("rangeleft").style.left = rangeleftXPosition + "px";
document.getElementById("rangeright").style.left = rangerightXPosition + "px";
167
168
169
170
171
      172
      173
174
175
176
177
178
      var triangleRightPosition = rangeleftXPosition + days * NAVBARCELLWIDTH + 20;
document.getElementById("triangleright").style.width = document.documentElement.
clientWidth -
179
                                                                                                 triangleRightPosition
                                                                                                 SCROLLBARWIDTH +
180
      "px";
document.getElementById("triangleright").style.left = triangleRightPosition + "px";
181
182
183
```

158