

Personal Knowledge Desk

Erstellung eines Prototypen zur Verwaltung des persönlichen Wissens in Form eines virtuellen Schreibtischs

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Informatikdidaktik

eingereicht von

Romana Hofbauer

Matrikelnummer 0725844

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung

Betreuer: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Futschek

Wien, TT.MM.JJJJ

(Unterschrift Verfasserin)

(Unterschrift Betreuer)

Eidesstattliche Erklärung

"Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe."

Ort, Datum

Unterschrift

Kurzfassung

In der heutigen Zeit sammelt sich viel persönliches Wissen an. Daher bedarf es eines Systems, welches diese Daten zentral an einer Stelle vereint und eine Organisation dieses Wissens erlaubt. In dieser Arbeit wird ein Prototyp entwickelt, der dieses Ziel erreichen soll. Als Plattform wird eine Webapplikation ausgewählt, damit kein Installationsaufwand erforderlich ist und es von überall aus zugänglich und plattformunabhängig ist. Es werden nur ausgewählte, wichtige Features implementiert, um die Komplexität möglichst niedrig zu halten. Der Hauptfokus dieser Abhandlung liegt dabei auf dem User Interface: dieses soll intuitiv und ästhetisch gestaltet werden. Das Design ist dabei an einen Schreibtisch angelehnt, um besonders für ältere Personen Einstiegspunkte für eine einfachere Benutzung zu schaffen.

Als ersten Schritt für die Umsetzung wird eine umfassende Recherche in der Fachliteratur durchgeführt. Durch diese Methodik wird herausgefunden, welche open-source Technologien für die Entwicklung gut geeignet sind. HTML 5, CSS 3, Javascript und AJAX werden für die Struktur, das Styling und die dynamische Änderung des Inhaltes verwendet. Weiters kommt PHP und MySQL zum Einsatz, um auf Daten zuzugreifen und diese zu speichern. Zusätzlich werden diverse Frameworks verwendet, um die Implementierung der Oberfläche zu erleichtern. Die zweite Erkenntnis aus der Literaturrecherche ist ein Set von Guidelines, welches beschreibt, wie das User Interface möglichst benutzerfreundlich gestaltet werden kann.

Nach diesen Schritten wird die Methode des Evolutionary Prototypings angewendet, da somit bereits die Grundfunktionalität der zukünftigen Software entsteht. Dadurch kann überprüft werden, ob die verwendeten Technologien passend sind, und außerdem können die Guidelines für die Gestaltung des User Interfaces umgesetzt werden. Zusätzlich ermöglicht ein Prototyp mit Kernfunktionalität auch eine Evaluierung.

Nach dieser Herangehensweise entsteht ein Tool names Personal Knowledge Desk, welches folgende Features enthält: Pinnwand, Dateisystem, Kalender, Aufgabenliste, Wörterbuch, Telefonbuch, Notizblock und Lesezeichen.

Den Abschluss der Arbeit bietet eine umfassende Evaluierung des Prototypen in zwei Phasen. Zuerst wird eine Evaluation anhand von Usability Heuristiken durchgeführt, und anschließend wird Thinking Aloud Usability Testing verwendet. Durch den Einsatz von zwei unterschiedlichen Methoden soll sicher gestellt werden, dass der Prototyp hinsichtlich der Benutzerfreundlichkeit gut getestet wird und somit mögliche Schwachstellen gefunden werden können.

Abstract

Nowadays a lot of personal knowledge can be gathered. Therefore a system is needed which merges this data on a central position and is open for organisational measures. In this paper a prototype will be developed which should reach this goal. As a platform a webapplication will be chosen since there would be no need to install it separately and it can be accessed platform independent during anytime from anywhere. To keep the complexity low, only preselected important features will be implemented. The main focus of this thesis will be on the user interface: it should be designed to be intuitive and aesthetic. The look is inspired by an ordinary desk so that especially older people can find entry points on how to work with this system.

As a first step during the implementation process a research in the technical literature will take place. The results of this methodology are open-source technologies which are suitable for the development. HTML 5, CSS 3, Javascript and AJAX will be used for the structure, the styling and the dynamic changes of the content. To write and access data, PHP and MySQL will be put to use. Additionally several frameworks will be used to facilitate the implementation of the user interface. The second finding from the research is a set of guidelines which describes how to create a design which is as user-friendly as possible.

After these two steps, evolutionary prototyping will be applied so that the main functionality of the future software forms. During the process of developing the prototype it can be checked if the used technologies are suitable and also the guidelines for the design of the user interface can be processed. Additionally, the prototype enables an evaluation.

After this approach a tool named Personal Knowledge Desk results which contains the following features: a pinboard, a file system, a calendar, a list of tasks, a dictionary, a telephone book, a notepad and bookmarks.

The conclusion of the thesis is a comprehensive evaluation of the prototype in two phases. At first an evaluation through usability heuristics will be performed and afterwards Thinking Aloud Usability Testing comes to use. Through the utilization of two different methods it should be ensured that the prototype is tested against user-friendliness and that possible weak points can be found early on.

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während des Studiums und der Anfertigung dieser Diplomarbeit unterstützt und motiviert haben:

Ein besonderer Dank gilt Herrn **Gerald Futschek**, der diese Arbeit betreut hat und mich immer in die richtige Richtung gelenkt hat.

Daneben gilt mein Dank **meiner Familie** und besonders **meinen Eltern**, ohne deren Unterstützung das Studium nicht möglich gewesen wäre.

Weiters möchte ich mich bei den **TeilnehmerInnen der Evaluation** meines Prototyps für ihr wertvolles Feedback bedanken.

Zum Schluss danke ich noch Herrn **Manuel Zelenka**, der mir sowohl während des Studiums als auch bei der Erstellung der Diplomarbeit mental und fachlich immer beiseite stand.

Inhaltsverzeichnis

1	Aufgabenstellung und Methodik	1
2	State of the Art	7
3	Technologien	13
3.1	XAMPP - Extended Apache/MySQL/PHP/Perl	14
3.2	HTML5 - HyperText Markup Language 5	14
3.3	CSS 3 - Cascading Style Sheets 3	15
3.4	Javascript	15
3.5	MySQL - My Sequential Query Language	16
3.6	PHP - PHP Hypertext Preprocessor	16
3.7	Ajax - Asynchronous Javascript and XML	16
3.8	UI-widgets	16
4	Prototyping	19
4.1	Überlegungen beim Design	20
4.2	Entwicklung des Prototyp	32
5	Evaluierung	61
5.1	Evaluierung anhand von Usability Heuristiken	61
5.2	Thinking Aloud Usability Testing	64
5.3	Zusammenfassung der Ergebnisse	75
6	Zusammenfassung und Ausblick	79
	Literaturverzeichnis	85

Abbildungsverzeichnis

1.1	Personas: Beispiele	4
1.2	Szenario	5
2.1	Probleme und mögliche Lösungen beim momentanen Technikstand	11
2.2	Vergleich mit dem Prototyp	12
4.1	Usability Wordcloud	21
4.2	F-Pattern: Heat Map	23
4.3	Bildschirmauflösung 2010 - 2014	24
4.4	Schriftart Verdana	27
4.5	Scrolling	29
4.6	Desktop	33
4.7	Login	35
4.8	Menü beim Startbildschirm	36
4.9	Pinnwand	36
4.10	Hinzufügen von Elementen auf der Pinnwand	37
4.11	Entfernen von Elementen auf der Pinnwand	38
4.12	Dateisystem	38
4.13	Kalendarleiste	41
4.14	Kalender Wochenübersicht	41
4.15	Kalender Monatsübersicht	41
4.16	Kalender Tagesansicht	42
4.17	Aufgaben	43
4.18	Aufgaben ohne verfügbare Tasks	44
4.19	Erledigte Aufgaben anzeigen	45

4.20	Notizzettel	46
4.21	Lesezeichen	47
4.22	Lesezeichen ohne Daten	47
4.23	Wörterbuch	49
4.24	Wörterbuch ohne Daten	50
4.25	Telefonbuch	51
4.26	Telefonbuch ohne Daten	52
4.27	Aufbau der Module	53
4.28	vertikale Navigation	54
4.29	Suchfeld	54
4.30	Suche mit Ergebnissen	55
4.31	Suche ohne Ergebnisse	55
4.32	Tooltip beim Fotowürfel	56
4.33	Tooltip bei Icons	56
4.34	Icons und Buttons	57
4.35	Dialog	58
4.36	Success modal message	59
4.37	Error modal message	59
5.1	Anzahl der PropandInnen vs. gefundene Usability Probleme	67
5.2	Aufgaben beim Usability Test	69
5.3	Thinking-Aloud Protokoll	71
5.4	Buttons mit Text	76
5.5	Optimiertes Dateisystem	77
5.6	Optimiertes Telefonbuch	77
5.7	Optimiertes Wörterbuch	78
5.8	Optimierte Elemente der Startseite	78

Aufgabenstellung und Methodik

Wir leben in einer Gesellschaft, in welcher eine regelrechte Informationsflut herrscht. Man wird überall mit einer Vielzahl von Daten konfrontiert. Es ist nicht leicht, den Überblick zu behalten und dadurch entsteht das Problem, dass es immer schwieriger wird, Informationen wieder aufzufinden. Es gibt heutzutage sehr viele verschiedene Medien, in welchen die persönlichen Daten abgelegt werden können. Zu unterscheiden sind einerseits lokale Dateisysteme wie zum Beispiel interne und externe Festplatten und USB-Sticks, und andererseits das in den letzten Jahren immer stärker aufkommende Cloud Computing oder auch Netzwerkspeicher. Wenn nun eine bestimmte Datei gesucht wird, kann das Finden dieser durchaus zu einer Herausforderung werden, da nicht nur eines, sondern möglicherweise viele verschiedene Systeme durchsucht werden müssen. Um dieses Problem zu beheben, wurden schon einige Lösungen von den verschiedensten HerstellerInnen bereit gestellt, um alle Daten zentral an einer Stelle zu vereinigen. Diese Systeme sind teilweise sehr kompliziert und enthalten unnötige Features, sind nur mit Fachwissen zu installieren oder nur für bestimmte Betriebssysteme verfügbar. Das größte Problem dabei ist allerdings das User Interface: es ist meist nicht sehr intuitiv und außerdem nicht schön gestaltet.

Abgeleitet aus dieser Problemstellung werden in dieser Diplomarbeit folgende Fragen bearbeitet:

- **Wie soll das User Interface in Form eines Schreibtischs am besten gestaltet sein, damit die Benutzerfreundlichkeit am höchsten ist?**
- **Welche Technologien sind für die Umsetzung gut geeignet?**

Das erwartete Resultat ist ein Prototyp eines Systems, welcher die oben beschriebenen Problemstellungen lösen soll. Als Plattform wird eine Webapplikation entwickelt,

damit die Daten von jedem beliebigen Ort aus zugänglich sind. Um keinen Installationsaufwand zu haben, können sich UserInnen durch eine einmalige Registrierung einloggen und mit der Benutzung beginnen. Die Benutzeroberfläche soll als altbekannter Schreibtisch simuliert werden, und Features wie Notizblock, Kontakte, Kalender, Lesezeichen, Aufgabenliste und eine Pinnwand beinhalten. Weiters soll das User Interface intuitiv und ästhetisch gestaltet werden.

Zusammenfassend soll also eine Webapplikation mit dem Namen Personal Knowledge Desk entwickelt werden, welche es ermöglicht, den eigenen Schreibtisch mit den wichtigsten Dingen immer und überall dabei zu haben. Somit soll das persönliche Wissen leicht abgelegt und wieder gefunden werden können, und außerdem soll die Benutzung des Tools Spaß machen.

Stand der Technik

Es gibt schon einige Systeme, welche versuchen die oben angesprochenen Probleme zu beheben, wie zum Beispiel Apple's Cloud-Service iCloud oder open-source Services wie ownCloud. Weiters wurden auch schon Programme entwickelt, die dem angestrebten Prototypen ähnlicher sind und alternative Interfaces zur Verfügung stellen. Mehr dazu im Kapitel State of the Art.

Zielgruppe

Grundsätzlich ist die Zielgruppe des Personal Knowledge Desks sehr breit gefächert, da dieses Programm von Personen jedes Alters mit wenig Computererfahrung benutzt werden kann. Ein besonderer Fokus wird dabei auf SeniorInnen gelegt, welche in ihrem früheren Job bereits Computer verwendeten, jedoch noch nicht sehr viel Erfahrung damit haben. Durch das einfache User Interface in Form eines Schreibtischs soll ein leichter Einstieg garantiert werden.

Wissenschaftliche Methoden

Folgende wissenschaftliche Methoden werden angewendet, um diese Arbeit zu verfassen und das gesetzte Ziel zu erreichen:

1. Literaturrecherche:

Zu Beginn wird eine umfassende Recherche in der Fachliteratur durchgeführt, um gut geeignete Technologien für die Umsetzung des Personal Knowledge Desks

zu finden. Diese werden genau beschrieben um die jeweiligen Entscheidungen zu untermauern. Mehr dazu im Kapitel Technologien. Anschließend soll herausgefunden werden, wie das Design der Webapplikation am besten gestaltet werden kann, um eine ästhetische graphische Oberfläche zu schaffen und eine hohe Benutzbarkeit zu gewährleisten. Diese Aufgabe wird der Hauptteil der Recherchearbeiten sein.

2. Prototyping:

In diesem Schritt geht es um die konkrete Implementierung des Systems mit den entsprechenden Technologien. Dabei wird auf die Designansätze des vorigen Schrittes aufgebaut. Mehr dazu im Kapitel Prototyping.

3. Evaluierung:

Nach der Fertigstellung des Prototypen erfolgt die Evaluierung in zwei Phasen. Zuerst wird das System anhand von Usability Heuristiken getestet. Anschließend wird die Thinking Aloud Testing Methode herangezogen, um die Usability des Systems zu überprüfen. Hierbei müssen verschiedene ProbandInnen vordefinierte Aufgaben mit dem Programm lösen und ihre Gedanken dabei laut aussprechen. So wird auf mögliche Probleme aufmerksam gemacht und man erhält wertvolles Feedback. Mehr dazu im Kapitel Evaluierung.

Personas und Szenarios

Um einen noch besseren Überblick über die Ziele dieser Abhandlung zu geben, werden Personas und Szenarios angeführt. Als **Persona** wird ein verbreitetes Modell bezeichnet, welches besonders im Bereich der Human-Computer-Interaction eingesetzt wird. Jedes Persona repräsentiert fiktive UserInnen aus der Zielgruppe, wobei konkrete Eigenschaften, Charakteristiken und das Nutzungsverhalten beschrieben werden. Der große Vorteil durch die Verwendung von Personas ist die Veranschaulichung der Zielgruppe in allen Entwicklungsprozessen. Außerdem werden Personas in einer lesbaren Sprache verfasst, damit es keine Verständigungsprobleme gibt [30].

Ein **Szenario** beschreibt eine Zusammensetzung von Use Cases, also potentielle Handlungen, die von Personas ausgeführt werden und Interaktionen mit dem Produkt verlangen [5].

Roland Maier

Geburtsdatum: 22.11.1988

Beruf: Student (Soziologie)

Hobbies: Musik, Fußball, Parties, fotografieren

Eigenschaften: intelligent, schlampig, vergesslich, hilfsbereit, umweltbewusst, familienbewusst, sportlich



wöchentliche Routine

Roland ist Student, jedoch ohne einem fixen Stundenplan. Darum ändert sich seine Routine von Tag zu Tag. Normalerweise versucht er gegen 9 Uhr aufzustehen und sich für die Uni vorzubereiten. Wenn er einen wichtigen Kurs hat, fährt er eine halbe Stunde davor zur Universität, um sich mit KollegInnen zu unterhalten. Jeden Montag und Mittwoch hat er um 19.00 Fußballtraining und am Donnerstag geht er mit Freunden aus. Am Freitag steigt er um 15.00 in den Zug, um zu seiner Familie zu fahren.

Computererfahrung

Roland besitzt einen eigenen Laptop, und hat zu Hause bei seiner Familie einen Standrechner. Da er seine Arbeiten elektronisch erledigt, kennt er sich gut mit Computern aus und verwendet ihn jeden Tag. Da er außerdem ein großer Fan von sozialen Netzwerken und Plattformen wie YouTube ist, verbringt er viel seiner Zeit im Internet.

Maria Gruber

Geburtsdatum: 02.02.1929

Beruf: Pensionistin

Hobbies: Enkel, spazieren, Skat

Eigenschaften: freundlich, intelligent, wissensdurstig, ordnungsliebend, pünktlich



wöchentliche Routine

Der Morgen beginnt für Maria jeden Tag um 07.00, wo sie Frühstück für sich und ihren Mann zubereitet. Anschließend liest sie die Zeitung, um sich über die Neuigkeiten zu informieren und sieht fern. Während ihr Mann gegen Mittag das Essen zubereitet, bedient sie den Computer um online Sudoku zu spielen und die Statusmeldungen auf Facebook zu verfolgen. Den Nachmittag und Abend verbringt sie gemütlich mit ihrem Mann. Jeden Freitag ist außerdem Skat-Abend, und am Wochenende wird sie von ihrer Familie besucht.

Computererfahrung

Maria hat sich in den letzten Jahren ihrer Arbeitstätigkeit oberflächlich mit dem Computer beschäftigt, um Emails zu schreiben und ein Buchhaltungsprogramm zu gebrauchen. Seit ihrer Pension benutzt sie den PC häufiger, um zum Beispiel Sudoku online zu spielen. Ihr Enkel bringt ihr außerdem manchmal neue Sachen bei, wenn er sie am Wochenende besucht.

Abbildung 1.1: Personas: Beispiele

Scenario

Roland fährt direkt von der Uni zu seiner Oma. Da er so wenig Gepäck wie möglich mitnehmen wollte, hat er seinen Laptop im Studentenheim gelassen. Er hat in den letzten Tagen einige Fotos gemacht, die er seiner Oma zeigen will. Dazu loggt er sich im Personal Knowledge Desk ein, um sie zu speichern. Danach will er ein lustiges Katzenvideo vorführen, wobei er den Titel des Videos vergessen hat. Um dieses Problem zu beheben, geht er zu der Bookmarks Sektion in der Applikation wo er den Link abgespeichert hat, und mit einem Klick erscheint das Video auf dem Bildschirm. Er erstellt außerdem einen neuen Task, dass er seiner Oma das zweite Katzenvideo schicken soll. Nach einigen Minuten auf YouTube fällt Oma Maria ein, dass sie in den nächsten Tagen in der Stadt ist, um sich mit einer Freundin zu treffen. Weil sie ihren Enkel noch nie im Studentenheim besucht hatte, will sie das nachholen. Roland sieht im Kalender nach, ob er an diesem Tag Zeit hat. Leider ist genau zum selben Zeitpunkt ein wichtiges Seminar. Die beiden einigen sich auf einen Termin im nächsten Monat, und tragen diesen auch schon im Kalender ein. Nach einem Gespräch über das Abendessen entschuldigt sich Roland bei seiner Oma, und macht sich an seine Seminararbeit. Im Notizblock vom Personal Knowledge Desk hat er bereits einige Anmerkungen aufgeschrieben. Um nun mit der Arbeit zu beginnen, holt er sich seinen Zwischenstand von seinem angelegten Uni-Ordner. Er liest sich seinen bisherigen Text durch und öffnet eines der Lesezeichen, um mit der Recherche fortzufahren. Zwei Stunden später ist er fertig und stolz auf seine Arbeit. Um sie nicht zu verlieren, lädt er sie sofort im Personal Knowledge Desk hoch um sie später nach einem letzten Durchlesen im Abgabesystem hochzuladen. Er öffnet die Tasks wieder, und notiert sich was er am Wochenende noch machen muss, bevor er sich dem Abend widmet.

Abbildung 1.2: Szenario

State of the Art

Das Internet ist seit einigen Jahren die wichtigste Quelle, um sich Informationen zu beschaffen. Da UserInnen anspruchsvoller werden, müssen sich auch die Plattformen ändern, die den Zugang zu den Daten ermöglichen. Der Trend geht vom normalen Desktop weg in Richtung mobiler Geräte und Clouds [29]. Besonders Letztere sind aufgrund ihrer unumstrittenen Vorteile stark im Kommen: BenutzerInnen müssen sich wenig bis gar nicht um die Infrastruktur kümmern, da Daten in der Cloud abgespeichert werden. Somit wird das Risiko des Datenverlustes gemindert, da diese Systeme Backups anbieten. Ein weiterer Pluspunkt ist die Tatsache, dass die Informationen von überall aus zugänglich sind und nicht nur lokal benutzt werden können. Außerdem sind die Kapazitäten so hoch, dass beinahe alle Daten abgelegt werden können [28]. Weiters wird durch solche Systeme die Kollaboration unterstützt und Plattformunabhängigkeit geboten.

Große Firmen wie Google erkennen das Potential dieser neuen Technologien und versuchen auch Applikationen online zugänglich zu machen, wie zum Beispiel Google Drive, was ähnliche Funktionen wie Microsoft Office anbietet und nur durch einen Browser von jedem Gerät aus aufrufbar ist [31].

Aufgrund der Änderung des NutzerInnenverhaltens gibt es schon einige Bemühungen, aus bewährten Denkmustern auszubrechen und Alternativen für den gewöhnlichen Desktop zu liefern. Die nächsten Seiten widmen sich einige dieser Programme.

Sem-quickly

Sem-quickly ist ein System basierend auf Semantic Web, welches es ermöglicht, Daten vom Web, Email oder anderen Applikationen automatisch auf Sticky-Notes zu schreiben. Das Knowledge Management System baut auf RDF (Resource Description Frame-

work) auf und bietet auf einfache Weise eine umfassende Wissensgewinnung durch das Schreiben und Managen von Information. Um das Tool zu nutzen, müssen UserInnen lediglich einen Text in einer beliebigen Applikation oder Website selektieren und Ctrl+Shift+S auswählen. Dann wird dieser Text gemeinsam mit der URL der Website, Titel, Inhalt, Tags, Quelle, Erstellzeit und letzter Änderungszeit als RDF Ressource gespeichert und das Sticky-Note erstellt. Jeder erfasste Inhalt wird automatisch verarbeitet, indiziert und in die Suche eingebaut [39].

Vorteile: Das User Interface ist intuitiv und einfach gehalten. Es ist leicht möglich, Wissen in einer bestimmten Form zu speichern.

Nachteile: Alle Informationen sind zwar als RDF Ressource verfügbar, können aber nur als Sticky-Notes dargestellt werden. Dies kann leicht unübersichtlich werden, und außerdem sind Sticky-Notes als einzige Darstellungsform nicht ausreichend.

VirtHome

VirtHome ist ein System welches es ermöglicht, auf unterschiedlichen Rechnern auf die eigene Computerumgebung zuzugreifen. Dabei können auch individuell angepasste Applikationen sowie die eigenen Daten verwendet werden, solange sich der Computer in einem TCP/IP Netzwerk befindet. Dabei wird das Betriebssystem inklusive den installierten Applikationen und Daten in ein System image verpackt, welches von einem Browser von überall aus zugänglich ist [15].

Vorteile: Durch VirtHome kann der persönliche Arbeitsplatz von überall aus benutzt werden, egal wie weit die geographische Entfernung ist.

Nachteile: Um das System zu benutzen, müssen BenutzerInnen einige Computerkenntnisse aufweisen, da die Erstinstallation nicht trivial ist.

3D virtual desktop

Virtuelle Arbeitsplätze können auf viele unterschiedliche Arten implementiert werden. Die Schwierigkeit dabei ist eine intuitive Interaktion zu ermöglichen. 3D virtual desktop befasst sich mit diesem Problem und versucht es durch ein dreidimensionales Interface und Gesten zu beheben. Die Größe aller Applikationen kann dabei manuell durch Handgesten gesteuert werden um einen besseren Überblick zu erhalten. [3]

Vorteile: 3D virtual desktop ist eine interessante Herangehensweise der Darstellung von Applikationen auf Interfaces. Die Steuerung durch Gesten ist besonders intuitiv.

Nachteile: Für die Benutzung des Tools ist eine eigene Hardware erforderlich. Außerdem ist mit der individuellen Steuerung ein hoher Einarbeitungsaufwand verbunden.

MICE³

MICE³ ist ein webbasierter Informationsdesktop, wo Ressourcen wie Links zu Websites, Services und Dokumente abgelegt und organisiert werden können. Die Elemente am Desktop werden dabei als Online-Verzeichnis repräsentiert und mit Informationen wie Konzepten und Entitäten versehen. Weiters werden Kategorien verwendet um dem Verzeichnis mehr Struktur zu verleihen. Ein Konzept beinhaltet die Bedeutung und den Zweck einer Kategorie und besteht aus Stichwörtern. Jede Kategorie enthält das Konzept, welches vom Inhalt der Kategorie gelernt wurde. Daher können durch das Tool Informationen von den eigenen Ressourcen gesammelt werden. Damit die gewünschten Daten erhalten werden, kann entweder durch das Verzeichnis navigiert, in den Inhalten danach gesucht oder im Index nachgeschlagen werden [12].

Vorteile: MICE³ ist ein praktisches Tool, um Informationen zu organisieren und danach zu suchen.

Nachteile: Ressourcen sind auf Links zu Websites und Dokumente beschränkt. Außerdem sind die Funktionen nicht ausreichend.

EyeOS

EyeOS ist ein moderner, virtueller Desktop, der von sämtlichen Devices aufgerufen werden kann und auch überall synchronisiert wird. Durch die eyeOS Webengine können alle vorhandenen Windows und Linux Applikationen in eine HTML 5 Webapplikation verwandelt werden [40]. EyeOS läuft auf einem zentralen Server, erstellt eine private Cloud und kann über einen Browser verwendet werden. Es sind sehr viele Features verfügbar und kann besonders von Firmen als Ersatz für normale Desktopsysteme verwendet werden.

Vorteile: EyeOS ist ein umfangreiches und mächtiges System, wessen Funktionen durch die Nutzung eines Browsers zur Verfügung stehen. Auch das User Interface ist intuitiv gestaltet.

Nachteile: Dieses Tool ist eine proprietäre Software und eher nur für Unternehmen gedacht.

iCloud

iCloud ist ein Cloud-Service von Apple, welcher seinen Vorgänger MobileMe im Juli 2012 abgelöst hat. iCloud ermöglicht es, die Daten auf Server zu laden und auf den eigenen Geräten automatisch zu synchronisieren. Es werden Features wie Kalender, Or-

tungsdienst, Notizen usw. angeboten. Außerdem können Daten wie Fotos und Musik gespeichert werden.

Vorteile: iCloud ist sehr einfach aufzusetzen und bietet ein schön gestaltetes und simples User Interface.

Nachteile: Dieser Dienst kann nur von Apple KundeInnen und somit auch nur auf Apple Devices genutzt werden.

ownCloud

ownCloud bietet einen open source Cloud Service für hauptsächlich private KundInnen an. Es wird dabei eine virtuelle Plattform zur Verfügung gestellt, welche ähnliche Funktionen wie der normale Computerdesktop besitzt. ownCloud zeichnet sich besonders durch hohe Security aus, da alle Daten auf dem eigenen Server der BenutzerInnen gespeichert werden. Steht kein Server zur Verfügung, könnten solche über verschiedenste AnbieterInnen angeschafft werden. Ein zusätzlicher Pluspunkt ist, dass UserInnen die Plattform modifizieren und erweitern können.

Vorteile: ownCloud bietet fast alle Funktionen eines gängigen Desktops an und ist kostenlos.

Nachteile: Es wird ein eigener Server benötigt und der Installationsaufwand ist relativ hoch.

Da das Aussehen des Personal Knowledge Desks an einen Schreibtisch aus der Kindheit angelehnt ist, wird noch kurz das Thema Metaphern angeschnitten. Frøkjær et. Al [6] diskutierten die positive Auswirkung bei der Verwendung von Metaphern. Diese unterstützen die menschliche Intuition, obwohl aktive Interpretation benötigt wird.

Durch eine Studie in Form von Interviews [27] wurde jedoch auch herausgefunden, dass die Desktop-Metapher als aktueller UI Standard sowohl eine Lösung darstellt, als auch Probleme aufwirft. Die Herausforderung besteht darin, dass einige Konventionen gemerkt werden müssen, anstatt nur rein intuitiv zu handeln. Obwohl die Desktop-Metapher überall verwendet wird, wurden die Benutzung und Auswirkungen noch nicht vollständig erforscht.

Um diese Probleme zu vermeiden, wird beim Prototyp keine Desktop-Metapher verwendet, sondern ein physischer Desktop visuell abgebildet. Dies soll einen einfachen Einstieg in das Programm ermöglichen.

Zusammenfassend treten bei den aktuell verfügbaren Systemen also folgende Hindernisse auf:

- Plattformabhängigkeit
- hohe Komplexität
- hoher Installationsaufwand
- viele unnötige Features
- nur lokal abrufbar
- User Interface ist nicht intuitiv

Genau diese Probleme sollen durch die Implementierung des Personal Knowledge Desks vermieden werden. Durch die Entwicklung als Webapplikation ist das System plattformunabhängig, von überall erreichbar und es gibt keinen Installationsaufwand. Es werden nur die wichtigsten Features implementiert und diese werden einfach gehalten. Das User Interface ist wegen des Designs als Schreibtisch aus der Kindheit intuitiv benutzbar.

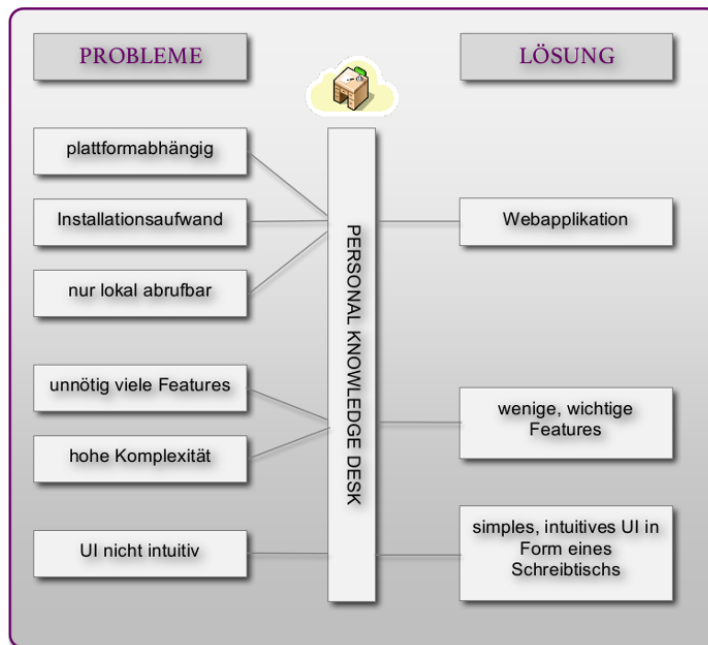


Abbildung 2.1: Probleme und mögliche Lösungen beim momentanen Technikstand

Die folgende Tabelle zeigt die Schwachstellen der Systeme und den Vergleich mit dem Prototypen:

	plattform-abhängig	Installations-aufwand	nur lokal verfügbar	komplex	Anzahl Features	Kosten	Technische Anforderungen	User Interface
Sem-quickly	Ja	Ja	Ja	Nein	Gering	Nein	Windows	minimalistisch
VirtHome	Nein	Ja	Nein	Ja	Hoch	Nein	Browser	Eigene Computer-umgebung
3D virtual desktop	Ja	Ja	Ja	Ja	Gering	Nein	Eigene Hardware	eigene Gesten, Einarbeitungs-aufwand
MICE³	Nein	Nein	Nein	Nein	Gering	Nein	Browser	minimalistisch
EyeOS	Nein	Nein	Nein	Ja	Sehr hoch	Ja	Browser	Benutzerfreundlich
iCloud	Ja	Nein	nein	Nein	Mittel	Nein	Browser	Benutzerfreundlich
ownCloud	Nein	Ja	Nein	Nein	Mittel	Nein	Eigener Server, Browser	Benutzerfreundlich
Personal Knowledge Desk	Nein	Nein	Nein	Nein	Mittel	Nein	Browser	Benutzerfreundlich

Abbildung 2.2: Vergleich mit dem Prototyp

Technologien

Seit dem Aufkommen von Web 2.0 ist das Internet nicht mehr aus unserer Gesellschaft wegzudenken. Es verändert die Art und Weise wie Menschen miteinander interagieren, ihre Arbeit erledigen, alltägliche Geschäfte tätigen, persönliche Daten ablegen und noch vieles mehr. Eine weitere interessante Entwicklung, die erst seit den letzten Jahren stattfindet, ist die Art, wie Software entwickelt wird. Früher wurde Software nur für bestimmte Computerarchitekturen, spezielle CPUs oder ausgewählte Betriebssysteme geschrieben. Heute wird die Mehrzahl der Applikationen für das Web entwickelt [36]. Diese stehen im Internet als Services zur Verfügung und die benötigten Ressourcen, wie zB Code und Daten, können sich überall auf der Welt befinden. Dieser Trend geht soweit, dass UserInnen in Zukunft nur noch einen lauffähigen Browser benötigen, um den Großteil ihrer Programme ausführen zu können. Die immer mehr aufkommenden Webapplikationen eröffnen völlig neue Möglichkeiten, mit dem Ziel die Vorteile von Desktopapplikationen (Usability) und des WWW (einfaches Deployment) zu vereinen. Die größten Vorzüge sind:

- **keine manuelle Installation bzw. Upgrades:**
Webbrowser fungieren als Laufzeitumgebung, daher muss keine Installation am Client durchgeführt werden. Sind neue Versionen der Software verfügbar, geschieht dies automatisch und muss nicht manuell erledigt werden.
- **einfaches Deployment:**
Die Applikation kann augenblicklich überall auf der Welt deployed und geteilt werden. Es ist irrelevant, ob AnbieterInnen und UserInnen geographisch getrennt sind.
- **Formate:**

Es werden offene, applikationsunabhängige Formate bei der Webentwicklung verwendet. Diese Formate sind für Menschen leserlich und von Computern leicht zu verarbeiten.

Ein weiterer Pluspunkt ist die Tatsache, dass Webapplikationen von jedem Computer aus zugänglich sind.

Aufgrund dieser Argumente wird der Personal Knowledge Desk als Webapplikation implementiert.

Im Folgenden werden die verwendeten Technologien zur Umsetzung des Prototyps genauer beschrieben.

3.1 XAMPP - Extended Apache/MySQL/PHP/Perl

Um das Projekt in die Tat umzusetzen, werden einige Komponenten benötigt, wie zum Beispiel ein Webserver. Da auf open-source Produkte gesetzt wird, fällt die Entscheidung auf den Apache Webserver, welcher unter der Apache License geführt wird. Dieser Webserver ist außerdem der am meisten verwendete Webserver¹. Zu den Vorzügen zählen außer der weiten Verbreitung auch die einfache Konfiguration, sowie die große Anzahl an Modulen und Erweiterungen. Da wie bereits oben erwähnt einige Komponenten gebraucht werden und das Aufsetzen von einem Webserver kombiniert mit Datenbanken und einer Skriptsprache eine Herausforderung darstellen kann, wird XAMPP verwendet um diese Hürde zu bestreiten. XAMPP ist eine Apache Distribution, welche MySQL, PHP and Perl automatisch enthält und zusätzlich viele Betriebssysteme unterstützt. Die Verwendung von XAMPP ermöglicht EntwicklerInnen außerdem das lokale Testen des Codes, ist jedoch nicht als Produktivsystem geeignet. Durch die Verwendung von XAMPP sind einige Technologie-Fragen schon von selbst beantwortet [8], trotzdem wird auf jede kurz eingegangen.

3.2 HTML5 - HyperText Markup Language 5

HTML wird oft als die Sprache des WWWs bezeichnet und ist bei der Erstellung von Webinhalten unverzichtbar. Die Vorgängerversion HTML 4 verzeichnete noch einige Schwachstellen, wodurch in manchen Situationen Probleme nicht allein durch diese Skriptsprache lösbar waren. Durch die aktuelle Version HTML 5 wird das Programmieren von Webanwendungen revolutioniert, da sehr viele Erleichterungen eingebaut

¹<http://news.netcraft.com/archives/category/web-server-survey>; Stand: Februar 2014

sind und außerdem die Spezifikation erheblich verbessert wurde. Die Integration von Javascript und CSS 3 wurde optimiert, es gibt ganz neue Möglichkeiten für die Einbindung von Medien und außerdem können Anwendungen offline verwendet werden. Auch Google, YouTube und Apple nutzen die neuesten Features aus und erweitern bzw. verbessern dadurch ihre Produktpalette.

Durch HTML 5, CSS 3 und Javascript stehen Webapplikationen den Desktopapplikationen beinahe in nichts mehr nach, auch wenn noch nicht alle Browser alle Funktionen vollständig unterstützen [9].

3.3 CSS 3 - Cascading Style Sheets 3

CSS ist eine deklarative Sprache und wird verwendet, um die Struktur des Webinhaltes (HTML) von der Präsentation zu trennen. Vereinfacht gesagt ist es zuständig für das Styling des Contents. Durch CSS 3 wurden ähnlich wie bei HTML 5 enorme Verbesserungen erzielt, die sowohl die visuelle Darstellung verbessern, als auch eine Zeitersparnis bei der Entwicklung aufzeigen [2].

3.4 Javascript

Javascript ist eine Skriptsprache mit dem Hauptzweck, Inhalte dynamisch zu erstellen und zu modifizieren. In den letzten Jahren erhielt diese interpretierte Sprache einen enormen Aufschwung, da die Verarbeitungsgeschwindigkeit erhöht wurde und sie durch die bessere Performance ein gutes Werkzeug für die Webentwicklung darstellt [36]. Einer der größten Vorteile ist, dass Schnittstellen für die Anwendungsprogrammierung in allen Browsern integriert sind und Javascript somit vielseitig einsetzbar ist. Außerdem kann direkt auf NutzerInneneingaben reagiert werden (ohne Kontakt mit dem Server aufzunehmen), da Javascript clientseitig funktioniert.

jQuery

jQuery ist eine externe Bibliothek basierend auf Javascript, welche kostenlos und frei zugänglich ist. jQuery erleichtert einige Dinge, wie zum Beispiel die Auswahl von HTML-Elementen, Animationen oder Events. Es kann auf einfache Weise mit HTML 5 kombiniert werden bzw. Funktionen ersetzen.

3.5 MySQL - My Sequential Query Language

MySQL ist ein rationales Datenbankmanagementsystem und wird laut DB-Engines² am zweithäufigsten verwendet. Am meisten genutzt wird Oracle, welches jedoch nicht open-source angeboten wird. Zu den Vorzügen zählt, dass MySQL schnell, zuverlässig, erweiterbar und leicht zu benutzen ist. Weiters ist es einfach zu installieren, bietet mehrere Speichermodelle, sowie eine umfassende Dokumentation an [7]. Auch große Webseiten wie Facebook, Twitter und YouTube verwenden dieses System.

3.6 PHP - PHP Hypertext Preprocessor

PHP ist eine objektorientierte, imperative Sprache, deren Ausführung am Server geschieht. Der Quellcode wird dabei im Gegensatz zu Javascript nicht lokal am Client, sondern auf einem Webserver ausgeführt. Laut dem TIOBE programming community index ist es die beliebteste Skriptsprache³. Im Gegensatz zu den größten Konkurrenten Ruby und Python wurde PHP extra für das Web entworfen und ist von den gängigsten Programmiersprachen beeinflusst. Ein weiterer Vorzug ist die breite Datenbankunterstützung [11].

3.7 Ajax - Asynchronous Javascript and XML

AJAX beschreibt einen neuen Weg, um Daten mit dem Webserver auszutauschen, ohne die ganze Seite neu laden zu müssen. Durch die asynchrone Übertragung können HTTP-Anfragen durchgeführt werden, während die HTML-Seite angezeigt wird. Da immer nur die veränderten Teile einer Applikation neu geladen werden, wird durch die Verwendung von AJAX auch die Benutzerfreundlichkeit gesteigert.

3.8 UI-widgets

Um die Benutzeroberfläche schöner zu gestalten, wird auf vordefinierte, jedoch leicht adaptierbare Widgets zurück gegriffen.

²<http://db-engines.com/en/ranking>; Stand: Jänner 2014

³<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>; Stand: Februar 2014

jQuery UI

jQuery UI ist eine Erweiterung von jQuery und stellt ein Set von Effekten, Widgets und sonstigen User Interface-Elementen zur Verfügung, welches leicht eingebunden werden kann. Durch das moderne Design und die verschiedenen Themes ist diese Technologie sehr beliebt bei WebprogrammiererInnen.

Kendo UI

Kendo UI ist ähnlich wie jQuery UI ein Framework, das verschiedene Widgets zur Verfügung stellt. Somit können leicht User Interface-Elemente eingebaut werden, die auf allen Browsern laufen und modern gestaltet sind. Beim Personal Knowledge Desk wurde Kendo UI vor allem für solche Elemente verwendet, welche von jQuery UI nicht angeboten werden.

CKEditor

CKEditor ist ein open-source HTML Text Editor und enthält viele Features, welche durch Programme wie zum Beispiel Microsoft Word bekannt sind. In der Applikation wird er als Inline-Editor verwendet und stellt alle notwendigen Funktionen zur Verfügung, damit AnwenderInnen einen Text schön gestalten und beliebig formatieren können. Der große Vorteil dieser Technologie ist die Tatsache, dass der Editor leicht adaptierbar ist und die User Experience verbessert.

Full Calendar

Full Calendar baut auf jQuery und enthält sehr viel Funktionalität, um einen Kalender bestmöglich abzubilden.

Prototyping

Prototyping beschreibt eine wissenschaftliche Methode, welche gerne bei der Entwicklung von Software eingesetzt wird, um Planung und Erfolg von Projekten zu optimieren. Horst Lichter et. Al [18] haben eine gute Zusammenfassung dieser Vorgehensweise erarbeitet:

"Prototyping is an approach based on an evolutionary view of software development, affecting the development process as a whole.

Prototyping involves producing early working versions of the future application system and experimenting with them."

Prototyping ist somit die Implementierung eines geplanten Systems, jedoch nicht in vollem Funktionsumfang und nicht mit der geplanten Endqualität. Die Methode bietet eine Kommunikationsbasis für alle involvierten Stakeholder, um in einer frühen Phase des Entwicklungsprozesses bereits Ergebnisse zu sehen, Probleme zu finden und diese zu beheben. Somit kann unmittelbar Feedback gegeben werden, was zu einer enormen Qualitätssteigerung des Endprodukts führt.

Grundsätzlich wird zwischen drei Arten unterschieden [18]:

- **Exploratory Prototyping:** Das genaue Ziel sowie die detaillierte Umsetzung des Systems ist noch nicht genau festgelegt. Es werden sowohl UserInnen- als auch Managementanforderungen geprüft, um eine Anforderungsspezifikation zu erhalten.
- **Experimental Prototyping:** im Vordergrund steht die technische Implementierung der Entwicklungsziele. Hier ist besonders die Kommunikation zwischen EntwicklerInnen und UserInnen gefragt.

- **Evolutionary Prototyping:** Hierbei ist das Ziel ein System mit der notwendigen Grundfunktionalität. Die Anwendung wird mit Hilfe von Feedback iterativ verbessert.

In dieser Arbeit wird die Methode des Evolutionary Prototyping verwendet, um ein Grundsystem zu erhalten und dieses evaluieren zu können.

4.1 Überlegungen beim Design

Bevor auf das Design des Prototypen näher eingegangen wird, soll an dieser Stelle die Forschungsfrage wiederholt werden:

“Wie soll das User Interface in Form eines Schreibtischs am besten gestaltet sein, damit die Benutzerfreundlichkeit am höchsten ist? “

Besonders wichtig bei der Forschungsfrage ist das Wort Benutzerfreundlichkeit. Daher wird kurz erklärt was es bedeutet und warum es wichtig ist: Usability ist eine Qualitätseigenschaft und besagt, wie intuitiv oder einfach eine Benutzeroberfläche verwendet werden kann und definiert sich unter anderem durch folgende fünf Kennzeichen [23]:

1. **Zufriedenheit:** Sind die Personen mit dem Design der Oberfläche zufrieden?
2. **Erlernbarkeit:** Wie einfach ist es, simple Aufgaben zu tätigen ohne das Programm bzw. das Design zu kennen?
3. **Effizienz:** Wie schnell können AnwenderInnen Problemstellungen lösen, wenn das Design bereits bekannt ist?
4. **Fehler:** Wie häufig treten Fehler bei der Benutzung des Systems auf? Wie schwerwiegend sind diese Fehler?
5. **Wiedererkennung:** Wie schnell können AnwenderInnen wieder in normalem Tempo mit dem System arbeiten, wenn sie es einige Zeit nicht benutzt haben?

Obwohl die Definitionen von Usability je nach Kontext variiert, ist diese generische Definition von ISO¹ doch sehr treffend:

“The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”

¹International Standards Organisation (ISO) 9241-11, “Ergonomics requirements for office work with visual display terminals (VDTs) – Part 11 Guidance on Usability, 1998.

Es wird empfohlen, mindestens zehn Prozent des Budgets vom Design eines Produkts für die Benutzerfreundlichkeit aufzuwenden [23].

Eine Studie [24] aus dem Jahre 2008 fand heraus, dass nach einem Usability Redesign Unternehmenskennzahlen wie Umsatzgenerierung, Seitenaufrufe oder Leistung um 83 % gestiegen sind.

Neben diesen Vorteilen existieren noch andere Vorzüge aus Unternehmenssicht [4]: Durch die Einfachheit der Benutzung werden Programme in der Regel gerne verwendet, was zu einer erhöhten Verkaufszahl und zu niedrigeren Supportkosten führt. Berücksichtigt man Usability bereits während des Softwarezyklus, werden auch die Entwicklungskosten gesenkt.

Werden IT-Systeme mit hoher Benutzerfreundlichkeit am Arbeitsplatz angewendet, so ergibt sich eine höhere Produktivität bei den ArbeitnehmerInnen, reduzierte Personalfluktuation und führt zu weniger Trainingskosten.

Es gibt natürlich auch andere Faktoren die dazu beitragen, ob AnwenderInnen ein Programm gerne verwenden. Beschäftigt man sich mit der Benutzeroberfläche, so stellt sich bald die Frage ob der Fokus eher auf Usability oder Ästhetik liegen sollte. Diese Thematik wird oft diskutiert und bis zum jetzigen Zeitpunkt wurde noch kein Konsens gefunden.

Ästhetik bezeichnet allgemein ein angenehmes Erscheinungsbild, bei der Softwareentwicklung also eine schöne Darstellung der Benutzeroberfläche. Ästhetik ist rein auf das Aussehen beschränkt und somit subjektiv, da alle Menschen verschiedene Empfindungen und Geschmäcker haben.

Yusof et. Al [41] haben sich 2010 mit diesem Thema genauer beschäftigt. Sie führten eine Studie durch und kamen zu dem Ergebnis, dass Ästhetik einen großen Einfluss darauf hat, ob ein System gerne benutzt und weiter erkundet wird. Ist eine Seite jedoch nicht benutzerfreundlich, so frustriert es AnwenderInnen und sie tendieren zum Verlassen der Website bzw. zum Nicht-Verwenden einer Applikation.

Folglich sind beide Aspekte essentiell für den Erfolg eines System. Es soll daher versucht werden, eine Balance zu finden und beide Gesichtspunkte berücksichtigen.

In den nächsten Seiten wird kurz das allgemeine Verhalten von UserInnen im Internet bzw. ihr Umgang mit Programmen beschrieben und anschließend wird darauf eingegangen, wie die verschiedenen Elemente des User Interfaces gestaltet werden sollen, um die Usability zu erhöhen.

F-shaped pattern

Inhalte im Web werden allgemein meist nicht genau durchgesehen, sondern lediglich kurz abgescannt. Die Nielsen Norman Group [21] führte eine umfassende, dreijährige Studie durch, um mit Hilfe von Eyetracking und 232 TeilnehmerInnen herauszufinden, wie Menschen mit Webinhalten interagieren. Das Ergebnis ist, dass AnwenderInnen den Informationsgehalt einer Seite extrem schnell durchsehen. Weiters deckten sie bei der Prüfung von vielen verschiedenen Websites auf, dass ein bestimmtes Lesemuster vorherrscht, das ähnlich dem Buchstaben F aufgebaut ist.



Abbildung 4.2: F-Pattern: Heat Map; Bild aus [21]

Dieses Pattern besagt, dass zuerst horizontal entlang des oberen Inhaltsbereiches abgescannt wird. Anschließend wandert der Blick nach unten, und dort wird wieder horizontal gelesen. Zum Schluss widmet sich das Auge dem linken, vertikalen Raum. Natürlich kann es je nach Aufbau des Inhaltes zu einigen Abweichungen kommen. Die Konsequenz aus diesem Muster zeigt folgende Guidelines auf:

- Die ersten beiden Absätze sollten Inhalte mit der meisten Substanz enthalten, da diese am wichtigsten sind und zuerst durchgesehen werden.

- Aufzählungen, Zwischentitel und Paragraphen sollen mit aussagekräftigen Wörtern beginnen, da ansonsten nicht weitergelesen wird.
- BenutzerInnen lesen nicht jedes Wort im Web durch. Daher sind sehr lange Texte nicht zu empfehlen und es ist notwendig, die wichtigsten Informationen im sichtbaren Bereich darzustellen.

Bildschirmauflösung

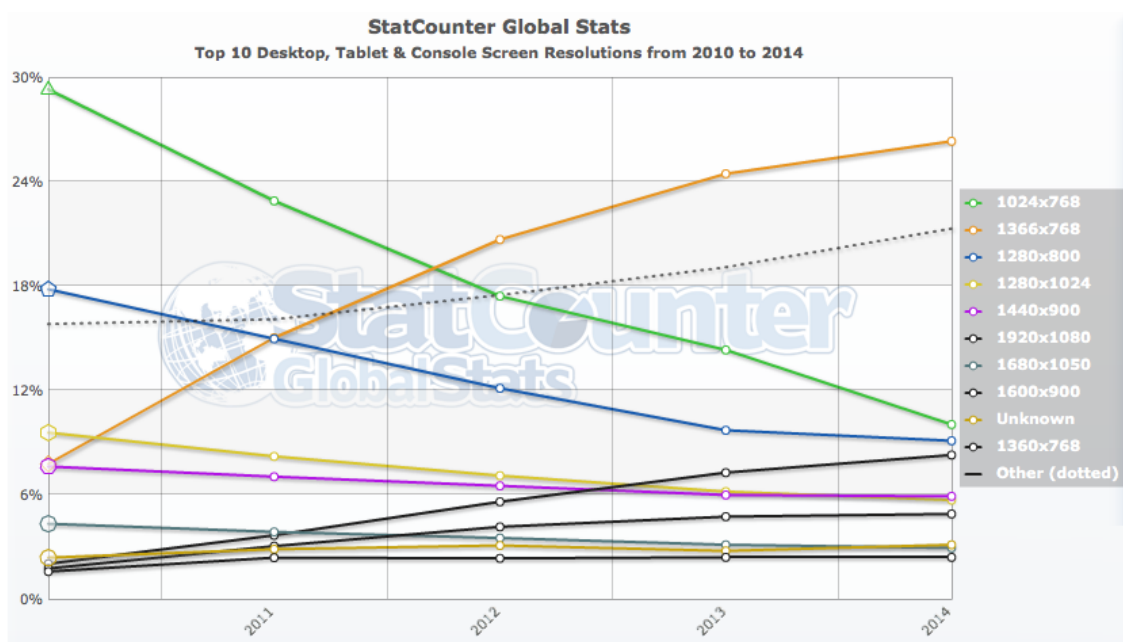


Abbildung 4.3: Bildschirmauflösung 2010 - 2014; Bild aus [34]

Beim Design einer Webapplikation stellt sich am Beginn die Frage, für welche Bildschirmauflösung das Programm optimiert werden soll. Dieser Punkt ist besonders wichtig für die Visibilität, Lesbarkeit und der primären Ästhetik. Da immer mehr verschiedene Monitorgrößen im Einsatz sind, kann nie auf alle BenutzerInnen Rücksicht genommen werden. Es empfiehlt sich jedoch, nicht mit fixen Größen sondern mit Prozentangaben zu arbeiten und so den Inhalt an alle Auflösungen anzupassen. Dadurch wird erreicht, dass die Applikation zwar für eine bestimmte Bildschirmauflösung optimiert wird, jedoch auch bei anderen Größen korrekt dargestellt wird.

Die Graphik von statcounter.com² veranschaulicht den Trend der Monitorgrößen seit 2010. Wie deutlich abzulesen ist, geht die Tendenz deutlich zu höheren Auflösungen. War im Jahr 2010 noch 1024x768 am beliebtesten, so ist es 2014 bereits 1366x768. Obwohl es in Richtung höherer Auflösung geht, sollte trotzdem noch für 1024x768 optimiert werden, um auch kleineren Bildschirmen gerecht zu werden [26].

Suche

“Search is the user’s lifeline for mastering complex websites.“ Dieses Zitat vom Usability Experten Jacob Nielsen verdeutlicht die enorme Bedeutung der Suche. Besonders bei umfangreichen Websites oder Webapplikationen empfiehlt es sich, eine Suche einzusetzen. Somit wird erreicht, dass UserInnen die gewünschten Informationen so schnell wie möglich und ohne unnötiges herumklicken erhalten.

Das Design der Suche ist nicht trivial. Es müssen einige Dinge beachtet werden, um die Komplexität zu verstehen [26].

Suchbox

Die Suche soll unmittelbar gesehen und erkannt werden, und außerdem leicht zu benutzen sein. Das Design sollte möglichst simpel und dem allgemeinen Look and Feel angepasst sein, aber trotzdem auffallen. Folgende Faktoren sind wichtig:

- **Position:** Die Suche sollte sich am linken oder rechten oberen Bildschirmrand befinden, weil sie an dieser Stelle von BenutzerInnen erwartet wird.
- **Zusammensetzung:** Die Suche sollte aus einem Eingabefeld und einem Button bestehen.
- **Eingabefeld:** Die Länge dieses Feldes soll so gewählt werden, dass mindestens 27 Zeichen angezeigt werden können. Dies hat den Vorteil, dass bessere Resultate erzielt werden, da AnwenderInnen längere Suchanfragen eintippen. Zusätzlich werden Rechtschreibfehler reduziert, da alles oder viel der Query sichtbar bleibt. Weiters soll das Eingabefeld die Form einer Box haben, da BenutzerInnen die Seite nach einer breiten Box abschnappen. Eine Abrundung des Feldes sieht userfreundlicher aus, auch wenn nicht bewiesen ist dass es die Benutzbarkeit steigert. Um den AnwenderInnen eine Vorstellung von möglichen Abfragen zu geben, kann eine

²StatCounter ist ein Analysetool und wird verwendet, um diverse Statistiken wie zum Beispiel über die häufigste Bildschirmauflösung oder die am öftesten verwendeten Browser zu erzeugen.

vordefinierte Query im Suchfeld angezeigt werden. Diese muss jedoch entfernt werden, sobald das Feld angeklickt wird.

- **Button:** Damit klar ersichtlich ist wo die Suche gestartet werden kann, soll der Button auch als tatsächlicher Button dargestellt werden. Die Beschriftung soll "Search" oder "Find" anstelle von "Go" sein.
- **Advanced Search:** Diese Funktion sollte nur in speziellen Fällen eingesetzt werden. Die Advanced Search Option soll nicht standardmäßig angezeigt werden, da diese nur von erfahrenen UserInnen benutzt wird und sonst zu Verwirrungen führt.

Ergebnisseite

Wird eine Query in ein Suchfeld eingetippt, erwarten die BenutzerInnen Ergebnisse. Die Resultate sollten stets in einer neuen Seite angezeigt werden, auch wenn nur ein Ergebnis zurückgeliefert wird. Eine sofortige Navigation zu diesem Element sollte in diesem Fall vermieden werden, da ein inkonsistentes Verhalten die AnwenderInnen meist mehr verwirrt als es Nutzen bringt. Die Resultate sollen in einer linearen Liste ohne Nummerierung dargestellt werden. Jedes Ergebnis sollte eine hervorgehobene und anklickbare Überschrift haben, gefolgt von einer kurzen zwei- bis dreizeiligen Zusammenfassung. Die ersten zwei Wörter sind am Wichtigsten, weil viele BenutzerInnen sich nicht die Mühe machen und alles durchlesen. Es ist außerdem von Vorteil, das Suchfeld auf der Ergebnisseite anzuzeigen, damit UserInnen wissen, wonach gesucht wurde und wenn nötig sogleich eine neue Anfrage starten können.

Es kann natürlich auch vorkommen, dass die Query nichts zurück liefert. In diesem Fall sollte deutlich dargestellt werden, dass nichts gefunden wurde. Weiters hilft es, Tipps zu geben wie bessere Resultate erzielt werden können bzw. auf typische Fehler aufmerksam zu machen. Somit wird verhindert, dass BenutzerInnen frustriert sind und die Seite bzw. Applikation verlassen.

Text

Der Text auf einer Webseite ist eines der wichtigsten Dinge, da damit Informationen jeglicher Art dargestellt werden können. Dabei gibt es folgende drei Dinge zu beachten, welche im Anschluss genauer beschrieben werden: Schriftart, Schriftgröße, Schriftfarbe [26].

Schriftart

Es wird empfohlen, "Verdana" zu verwenden, weil diese Schriftart:

- online auch bei kleineren Schriftgrößen am besten lesbar ist
- bei den meisten Browsern vorinstalliert ist
- eine moderne, einfache und professionelle Schriftart ist

Es können auch mehrere Schriftarten verwendet werden (zum Beispiel für Labels, Überschriften, Logo). Eine Variation kann hilfreich sein, wenn zwischen verschiedenen Bereichen unterschieden werden soll, wie zum Beispiel zwischen normalem Text und Code-Snippets. Auch das Zuweisen von Attributen wie Schriftstile oder Farben kann von Vorteil sein. Wichtig ist, dass nur eine sehr kleine Anzahl an verschiedene Schriften verwendet wird und auch Schriftschnitte sparsam eingesetzt werden.

8 pt	Verdana
10 pt	Verdana
12 pt	Verdana
14 pt	Verdana
16 pt	Verdana
14 pt, fett	Verdana
14 pt, kursiv	<i>Verdana</i>

Abbildung 4.4: Schriftart Verdana

Schriftgröße

Die Schriftgröße ist unter Anderem abhängig vom Zielpublikum. Folgende Größen werden empfohlen:

Publikum	Schriftgröße
allgemeines Publikum	10pt -12pt
SeniorInnen/Personen mit visuellen Beeinträchtigungen	12pt -14pt
Kinder/LeseanfängerInnen	12pt - 14pt
Teenager/junge Erwachsene	10pt - 12pt

Generell gilt allerdings, eher eine größere Schriftgröße zu verwenden, um Probleme bei manchen LeserInnen zu vermeiden. Als Faustregel gilt mindestens 10pt bei längerem Text.

Schriftfarbe

Verschiedene Farben werden gerne verwendet, um wichtige Elemente hervorzuheben. Es sollen jedoch nicht mehr als vier unterschiedliche Farben benutzt werden, da das System sonst unstrukturiert, unprofessionell und überladen aussieht. Weiters ist es vorzuziehen, schwarze Schrift auf einem weißen Hintergrund oder ähnliche Farben zu benutzen. Allgemein eignen sich dunkle Farben am besten für Text, und helle, entsättigte Farben für den Hintergrund. Wenig Kontrast senkt die Lesbarkeit und führt zu einer Überanstrengung der Augen. Es soll auch vor Kombinationen wie zum Beispiel Lila und Gelb gewarnt werden, weil der Inhalt trotz eines hohen Kontrastes schwer lesbar ist.

Buttons

Buttons sind wichtige Elemente einer Benutzeroberfläche, da sie auf mögliche Aktionen hinweisen. Beim Design ist Folgendes zu berücksichtigen [14] [16]:

- Durch das Aussehen soll klar sein, dass der Button betätigt werden kann. BenutzerInnen sollten nie überlegen müssen, ob ein Element eine Aktion hervorruft oder nicht.
- Konsistenz: Ähnliche Funktionen sollen durch gleiche Buttons gekennzeichnet sein. So sollten zum Beispiel Create-, Edit- und Delete Buttons gleich aussehen, ein Registrieren-Button kann jedoch wieder anders gestaltet sein. Wichtigere Buttons können zusätzlich hervorgehoben werden.
- Zustände: Berührt der Cursor den Button, soll sich dies farblich auswirken. Wenn der Button betätigt wird, soll ein ähnliches Anklicken wie bei physischen Buttons simuliert werden.

- Icons: Kleine Bilder helfen, den Zweck des Buttons visuell zu beschreiben.
- Reihenfolge: Auch die Reihenfolge spielt eine Rolle. In einem Dialog sollte sich beispielsweise der *Save-Button* links vom *Cancel-Button* befinden. Gibt es an mehreren Stellen im System die selben Buttons, so sollte die Reihenfolge stets konsistent sein.
- Platzierung: Buttons sollen so platziert werden, dass sie leicht gefunden werden.
- Naming: Für die Beschriftung sollen Verben verwendet werden, um mit einem Blick deutlich zu machen, welche Aktion das Betätigen hervorruft. Somit muss beispielsweise in Dialogen nicht der ganze Text vollständig gelesen werden, weil die Beschriftung der Buttons ausreicht um die möglichen Ereignisse zu beschreiben und zu verstehen.

Scrolling

Laut Jacob Nielsen [20] sind 80 % seiner Usability Guidelines von 1990 immer noch gültig. Scrolling betrifft das eine Fünftel der Richtlinien, das aufgrund von Änderungen im NutzerInnenverhalten modifiziert werden musste. Noch vor einigen Jahren waren UserInnen überhaupt nicht bereit zu scrollen, was heute nicht mehr vollständig zutrifft. Eine Studie [26], welche mit 2163 TeilnehmerInnen aus der ganzen Welt und 3992 verschiedenen Websites, die länger als ein Browserfenster waren, durchgeführt wurde, zeigte dass die die Hälfte der TeilnehmerInnen gar nicht scrollt, 14 % sich mehr wie 2 Seiten und nur 1 % über mehr als 7 Seiten vollständig angesehen hat.

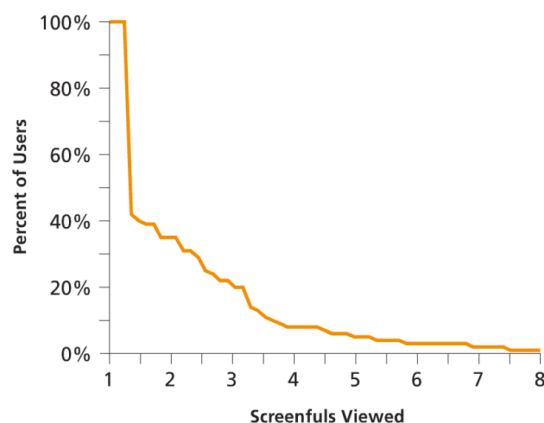


Abbildung 4.5: Scrolling; Bild aus [26]

Die wichtigste Richtlinie bezüglich des Scrollings besagt, dass sich die essentiellsten Informationen im "page fold"³ befinden sollen. Weiters sollen folgende Punkte berücksichtigt werden:

- Scrolling soll wenn möglich sparsam eingesetzt werden, da die Aufmerksamkeitsspanne von AnwenderInnen limitiert ist.
- Werden unbedingt längere Seiten benötigt (zum Beispiel für umfassende Artikel), soll alles auf einer Seite angezeigt und somit Scrolling verwendet werden. Paging als Alternative wird nur in sehr speziellen Situationen empfohlen.
- Wenn die Länge einer Seite über den Bildschirmrand hinausgeht, soll eine Scrollbar angezeigt werden. Ist der Platz ausreichend für den gesamten Inhalt, sollte die Scrollbar nicht gezeigt werden.
- Scrollbars sollen als solche erkannt werden, daher sollen wenig bis keine Adaptierungen des Designs vorgenommen werden.
- Horizontales Scrolling sollte ganz vermieden werden, weil BenutzerInnen nur vertikal scrollen. Wird beides verwendet, so müssen UserInnen den Blickwinkel in zwei Dimensionen bewegen, was zu mehr Anstrengung führt. Außerdem wird horizontales Scrolling von Hardwarekomponenten wie Maus oder Touchpad nur selten unterstützt.

Navigation

Die Navigation ist ein essentieller Bestandteil von Systemen, da sie unumgänglich ist um zu anderen Inhalten zu gelangen. Laut einer Studie [21], die mit Hilfe von Eye-tracking durchgeführt wurde um das Verhalten von AnwenderInnen auf Websites zu beobachten, kam heraus dass die ProbandInnen 24% der gesamten Zeit auf die horizontale globale Navigationsleiste am oberen Bildschirmrand blicken. Mehr als die Hälfte der Zeit sehen sie auf die horizontale Subnavigationsleisten oben und 49% auf die vertikale Navigation auf der linken Seite.

Wird die Navigation falsch designed, kann dies verheerende Auswirkungen haben. Daher sollte Folgendes beachtet werden [17]:

- Die globale Navigation soll unverändert sein wenn Elemente in der Subnavigation ausgewählt werden.

³Als page fold wird der Teil einer Seite bezeichnet, welcher ohne weitere Aktionen sichtbar ist.

- Es gibt einen einfach erkennbaren Weg, um zur Homepage zurück zu kehren bzw. zwischen verwandten Seiten zu navigieren.
- Die wichtigsten Informationen für UserInnen sind von den meisten Seiten erreichbar.
- Die Elemente in der Navigation sind auf logische Weise angeordnet. Dies könnte zum Beispiel alphabetisch, in Gruppen oder logisch zusammenhängend sein.
- Der Mauszeiger ändert sein Aussehen wenn ein Element anklickbar ist.
- Wird der *Back-Button* geklickt, werden BenutzerInnen zu der Seite zurückgebracht, von der sie kamen.

Screen-based controls (widgets)

Zum Schluss werden noch Tipps angegeben, um widgets allgemein besser zu gestalten [17]:

- Sind Felder obligatorisch, so sollen sie auch dementsprechend gekennzeichnet sein. Selbiges gilt auch für optionale Felder.
- Eingabefelder sollen aussagekräftig benannt werden.
- Labels sollen örtlich nahe bei den Eingabefeldern dargestellt werden, damit klar ist welche Information erforderlich ist. Außerdem wird empfohlen, dass die Beschriftung über den zugehörigen Feldern ist anstelle von links daneben.
- Radio Buttons sollen verwendet werden, wenn nur eine Option gewählt werden kann. Dementsprechend sollen Checkboxes benutzt werden, falls mehrere Alternativen bestimmt werden können.
- Der Cursor soll in dem ersten möglichen Eingabefeld platziert werden.
- Die Methode der Dateneingabe soll solange wie möglich gleich sein, damit AnwenderInnen nicht unnötig oft zwischen Maus und Tastatur wechseln müssen.
- Texteingabefelder haben die gleiche Länge wie die erwartete Eingabe.
- Dropdowns, Radiobuttons und Checkboxes werden Texteingabefeldern vorgezogen, um Fehler zu vermeiden und Optionen vorzugeben.

4.2 Entwicklung des Prototyp

In dieser Sektion wird beschrieben, wie die Benutzeroberfläche des Personal Knowledge Desks implementiert wird. Dabei wird besonders auf die Funktionen und das Design detailliert eingegangen.

Zur Gestaltung der einzelnen Elemente werden die Guidelines von dem vorigen Kapitel herangezogen. Somit soll gewährleistet werden, dass das Design auf einer wissenschaftlichen Basis geschieht und außerdem auf Usability Rücksicht genommen wird. Zusätzlich werden vor der Implementierung bestimmte Kriterien bezüglich der Benutzerfreundlichkeit festgelegt, welche während des Prototypings berücksichtigt werden:

- **Konsistenz:** Alle Elemente des User Interfaces sollen einheitlich gestaltet sein und auch konsistent funktionieren. Dieses Kriterium ist sehr bedeutend, da einzelne Abweichungen innerhalb des Systems Verwirrungen stiften können.
- **Feedback:** Es ist für UserInnen wichtig, Feedback für getätigte Aktionen zu erhalten. Dieses kann verschiedene Formen annehmen, wie zum Beispiel Textnachrichten oder andere graphische Darstellungsformen.
- **klare Navigation:** BenutzerInnen sollen in jedem Teil des Systems die Möglichkeit haben, zu den anderen Modulen zu navigieren. Weiters soll die Navigation so positioniert sein, dass sie unmittelbar gefunden werden kann.
- **Fehlervorbeugung:** Durch gezielte Maßnahmen soll es möglich sein, dass UserInnen weniger Fehler innerhalb des Systems machen können. Werden zum Beispiel in einem Zahlenfeld keine Buchstaben erlaubt, so schafft diese Eigenschaft bereits ein Problem aus der Welt.
- **lesbarer Text:** Jeglicher Text muss von allen Personen der Zielgruppe gelesen werden können. Daher sollten große Schriftgrößen und viel Kontrast verwendet werden. Weiters soll darauf geachtet werden, dass an keiner Stelle zu viel Text gelesen werden muss um bestimmte Aktionen ausführen zu können.
- **Wissen über momentanen Zustand:** BenutzerInnen sollen zu jedem Zeitpunkt darüber Bescheid wissen, wo sie sich gerade im System befinden und welche Aktionen von ihnen getätigt werden können.

Nach der Fertigstellung des Prototypen wird anhand einer Evaluierung in zwei Schritten festgestellt, ob das Ziel der hohen Benutzerfreundlichkeit erreicht wurde.

Desktop



Abbildung 4.6: Desktop

Der Desktop ist der Start-Bildschirm des Personal Knowledge Desks und fungiert im Prinzip als Navigation, um die anderen Module zu erreichen. Die Navigationselemente, welche beliebig via Drag and Drop verschoben werden können, sind nicht als Links angeordnet, sondern als Bilder. Diese Abbildungen symbolisieren Elemente eines realen Schreibtischs:

- **Lesezeichen:** Um zu den Bookmarks zu gelangen, wird ein Lesezeichen verwendet, welches in Büchern genutzt wird, damit die zuletzt gelesene Stelle wieder leicht gefunden werden kann.
- **Kalender:** Dieses Modul wird durch einen Wochenkalender mit Ringbindung simuliert.
- **Aufgaben:** Um die Tasks zu erreichen, wird ein beschriebenes Post-it verwendet, welches am Schreibtisch klebt.
- **Adressbuch:** Dieses Modul wird als Ringbuchblock für Kontakte dargestellt.
- **Wörterbuch:** Damit zum Dictionary navigiert werden kann, wird ein Bild des bekannten Webster's Collegiate® Dictionary benutzt.

- **Notizzettel:** Der Link zum Notebook wird als Notizblock mit Kugelschreiber dargestellt.
- **Dateisystem:** Das Filesystem sieht wie eine Ablage mit vier Ordnern aus.
- **Fotowürfel:** Auch ein Fotowürfel befindet sich auf einigen Schreibtischen und darf beim Personal Knowledge Desk nicht fehlen. Es können sechs verschiedene Bilder durch einen Doppelklick hinzugefügt werden. Soll der Würfel gedreht werden, ist dies mittels den Pfeiltasten möglich. Damit die Funktionsweise deutlich gemacht wird, beschreibt ein Tooltip die notwendigen Schritte.

Dieses konkrete Aussehen wurde bewusst so gewählt, damit ein physischer Schreibtisch simuliert wird. Es soll dabei helfen, sich leichter in der Applikation zurecht zu finden. Marcel Heerink [10] fand heraus, dass besonders SeniorInnen den Umgang mit technischen Geräten fürchten, da sie keinen besonderen Bezug dazu haben und auch die notwendige Übung mit solchen Systemen fehlt. Um diese Angst zu mindern, können bekannte Einstiegspunkte helfen. Da die meisten Personen vor dem digitalen Zeitalter einen ähnlich aufgebauten Schreibtisch in ihrem Kinder- oder Arbeitszimmer hatten, fällt die Wiedererkennung leichter. Dies ist ein besonders wichtiger Punkt ("Erkennen ist besser als Erinnern [19]") und gilt auch als eine der zehn wichtigsten Usability Heuristiken.

Zusätzlich zu der Navigation bietet der Desktop noch andere Funktionen an: Login, Suche und Menü.

Login

Beim Betreten des Personal Knowledge Desks ist der Schreibtisch dunkel eingefärbt, die Lampe ist ausgeschaltet und BenutzerInnendaten müssen eingegeben werden. Sind noch keine vorhanden, muss eine Registrierung mit wenigen Schritten durchgeführt werden. Nach dem Login kann die Applikation verwendet werden. Dies wird durch einen von der eingeschalteten Lampe erhellten Schreibtisch symbolisiert.

Suche

Um die Suche zu verwenden, muss ein Stichwort in das Suchfeld eingegeben und anschließend der Search-Button bzw. die Enter-Taste betätigt werden. Danach erfolgt eine Weiterleitung zu der Ergebnisseite. Mehr dazu in der Sektion Suche.

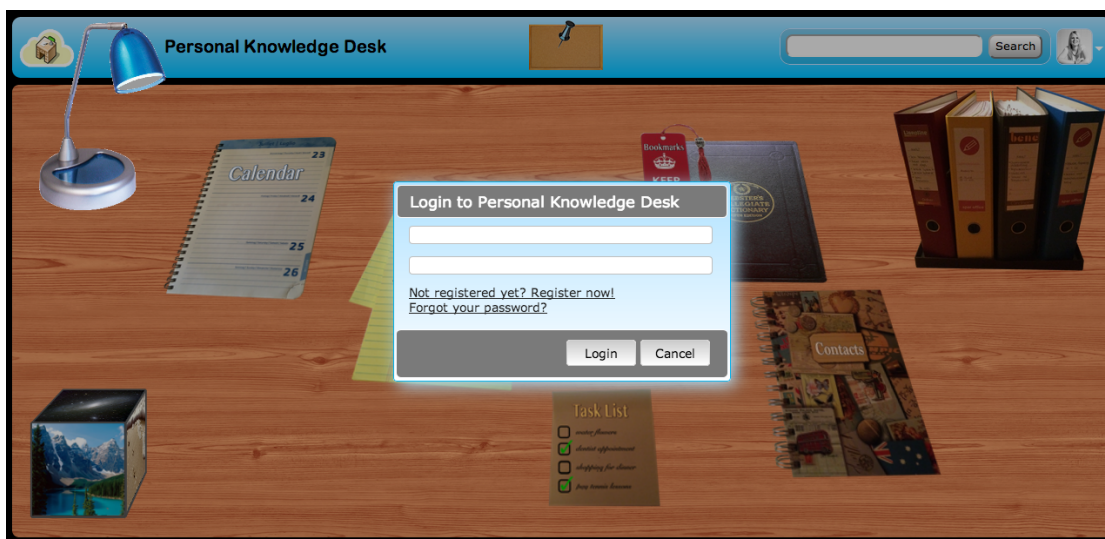


Abbildung 4.7: Login

Menü

Rechts oben befindet sich das Menü, um zusätzliche Einstellungen zu ermöglichen:

- **Change desktop color:** Standardmäßig wird ein Holzhintergrund dargestellt. Durch diese Option ist es möglich, den Hintergrund beliebig zu ändern.
- **Help:** Hier wird eine Hilfestellung gegeben falls Funktionen nicht klar sind.
- **About:** Im About Dialog werden Informationen zum Programm und den HerstellerInnen gegeben.
- **Logout:** Durch dieses Menüelement loggen sich Personen vom Personal Knowledge Desk aus. Nach dieser Aktion wird wieder der verdunkelte Startbildschirm und das Login-Formular angezeigt und die BenutzerInnendaten müssen für die Weiterverwendung erneut eingegeben werden

Beim Prototyp wird das Menü zwar angezeigt, allerdings wurde die Funktionalität hinter den einzelnen Menüpunkten nicht implementiert.

Pinwand

Im Kinderzimmer war über dem Schreibtisch auch meist eine Pinwand aufgehängt, um Fotos oder andere Erinnerungen aufzuhängen. Darum wurde dieses Feature auch

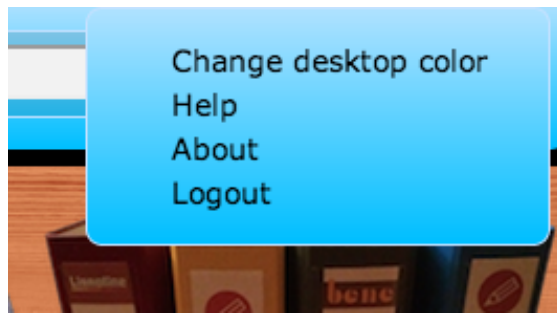


Abbildung 4.8: Menü beim Startbildschirm

beim Personal Knowledge Desk implementiert.

Auf die Pinnwand können Sticky-Notes in verschiedenen Farben und Fotos "gehängt" werden. Falls die Position nicht zufriedenstellend ist, können alle Elemente beliebig im Pinnwandbereich durch Drag and Drop an eine andere Stelle gebracht werden. Da dieses Modul über keine eigene Navigation verfügt, kann der *Back to Desktop-Button* benutzt werden um wieder zurück zum Schreibtisch zu gelangen.



Abbildung 4.9: Pinnwand

Element hinzufügen

Damit ein Element zur Pinnwand hinzugefügt werden kann, muss der *Add-Button* betätigt werden. Danach öffnet sich ein Dialog, in welchem ausgewählt werden muss ob ein

Sticky-Note oder ein Foto hinzugefügt werden soll. Bei Ersterem muss zusätzlich die Farbe ausgewählt werden. Bei Letzterem wird das gewünschte Bild mit Hilfe eines File Pickers selektiert. Durch Bestätigung mittels des *Add-Buttons* wird das entsprechende Element auf der Pinnwand angezeigt und ausgewählte Fotos im Zuge dessen auf den Webserver des Personal Knowledge Desks geladen.

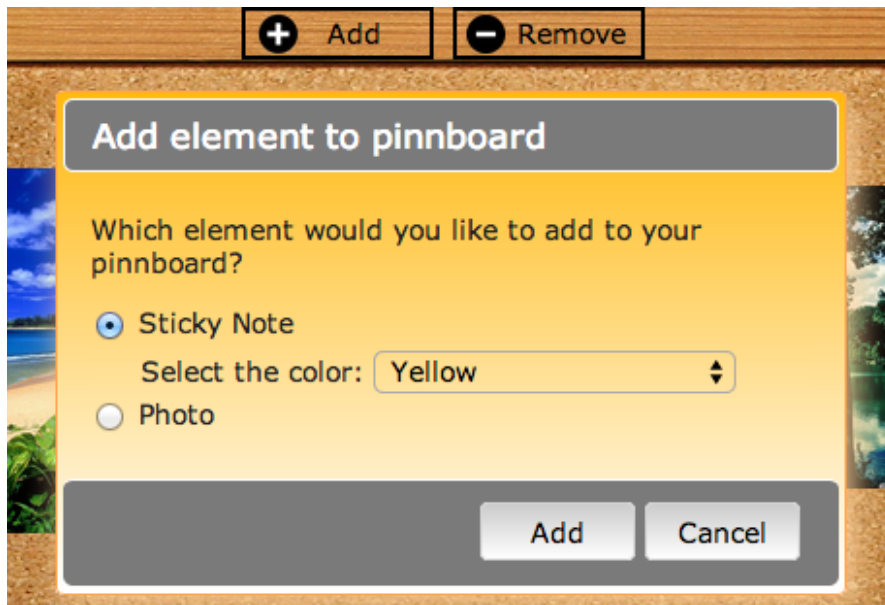


Abbildung 4.10: Hinzufügen von Elementen auf der Pinnwand

Element löschen

Sollen Elemente gelöscht und somit nicht mehr auf der Pinnwand dargestellt werden, ist dies mittels des *Remove-Buttons* möglich. Wird dieser gedrückt, so erscheint am rechten unteren Rand jedes Elements ein *Delete-Icon*. Wird dieses betätigt, so warnt ein Dialog davor, dass das Element entfernt wird und diese Aktion nicht mehr rückgängig gemacht werden kann. Anschließend wird das entsprechende Bild/Sticky-Note gelöscht und ist nicht mehr zu sehen. Wird der *Remove-Button* erneut angeklickt, werden die *Delete-Icons* nicht mehr angezeigt.



Abbildung 4.11: Entfernen von Elementen auf der Pinnwand

Dateisystem

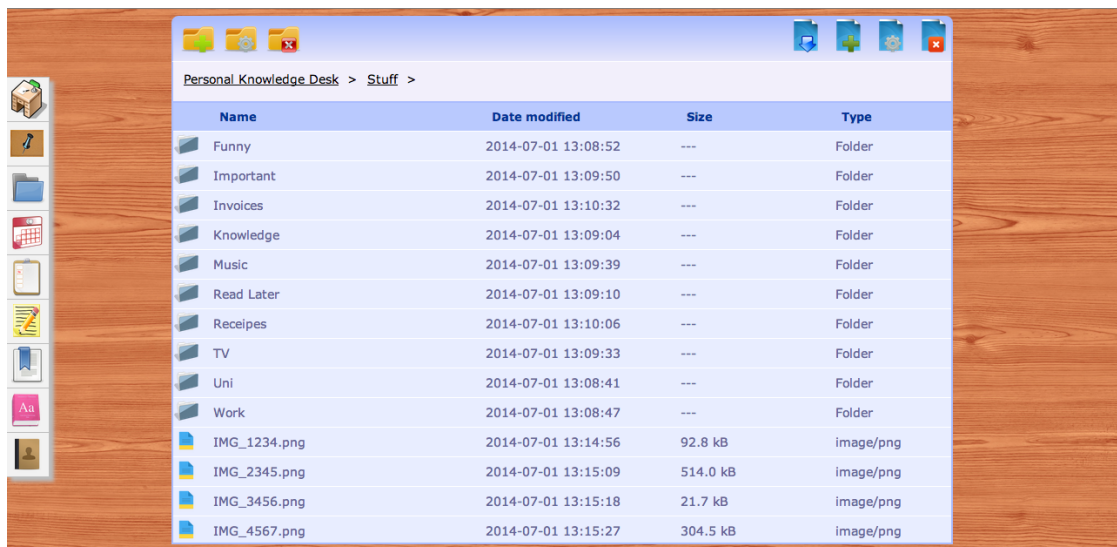


Abbildung 4.12: Dateisystem

Das Filesystem ist ähnlich der gewohnten Ordnerstruktur von Desktopsystemen aufgebaut. Dadurch wird es ermöglicht, Dateien hochzuladen und auch wieder herunterzuladen. Um die persönlichen Daten zu strukturieren und besser wiederzufinden,

können alle Dateien Ordnern zugewiesen werden. Befindet man sich in Unterordnern, so kann mit Hilfe eines Klicks auf die Breadcrumb-Leiste zu den spezifischen Überordnern navigiert werden.

Ordner hinzufügen

Durch einen Klick auf das *Add Folder-Icon* wird ein Dialog geöffnet, welcher es ermöglicht, neue Ordner zu erstellen. Dabei muss der Name des neuen Ordners in das Inputfeld eingegeben und der *Create-Button* betätigt werden. Somit wird der Ordner erstellt und in der Liste angezeigt. Zusätzlich signalisiert eine Modal message das erfolgreiche Hinzufügen des Ordners. Der Folder wird dabei immer an der Stelle der Hierarchie erstellt, in welcher sich der/die BenutzerIn momentan befindet.

Ordner bearbeiten

Um einen Ordner zu bearbeiten, muss dieser zuerst selektiert werden. Anschließend wird das *Edit Folder-Icon* gedrückt, welches einen Dialog öffnet. Hier kann der neue Name des Ordners eingegeben werden, wobei der alte Name per default im Feld angezeigt wird. Nach der Bestätigung erfolgt die Änderung des Namens, was sich in Folge in der Liste wieder spiegelt.

Ordner löschen

Soll ein Ordner gelöscht werden, geschieht dies durch das *Delete Folder-Icon*. Zuvor muss der entsprechende Ordner markiert werden. Nach Betätigen des Icons öffnet sich ein Dialog, welcher davor warnt den Ordner zu löschen und anmerkt, dass diese Aktion unwiderruflich ist und auch alle im Folder enthaltenen Files und Subfolders gelöscht werden. Wird die Aktion durch den *Delete-Button* bestätigt, ist der entsprechende Ordner sowie alle Files gelöscht und wird nicht mehr in der Liste angezeigt. Eine Modal message bekräftigt den Abschluss dieser Aktion.

File herunterladen

Um ein markiertes File herunterzuladen, muss das *Download File-Icon* gedrückt werden. Anschließend öffnet sich ein Dialog, in welchem eine Bestätigung des Downloads erforderlich ist. Danach wird die Datei heruntergeladen und im lokalen Download Ordner gespeichert.

File hinzufügen

Will man eine Datei zum Personal Knowledge Desk hinzufügen, ist dies im *Add File-Dialog* möglich. Hierbei muss der Pfad der gewünschten Datei im File Picker ausgewählt werden. Ist dies passiert, wird der Name der Datei angezeigt um nicht irrtümlich die falsche Datei hochzuladen. Nach Betätigung des *Add-Buttons* wird die Datei auf den Server geladen und im Ordnersystem dargestellt.

File bearbeiten

Um eine Datei zu editieren, muss der *Edit File-Dialog* mittels des *Edit File-Icons* geöffnet werden. Hier muss nur der Name des neuen Files eingegeben werden, wobei der alte Name per default im Eingabefeld angezeigt wird. Nach Bestätigung durch den *Edit-Button* wird die Datei editiert und im Filesystem entsprechend angezeigt.

File löschen

Wird eine Datei nicht mehr benötigt, kann diese gelöscht werden. Dazu muss im *Delete File-Dialog* bestätigt werden, dass die Datei tatsächlich gelöscht werden soll. Danach wird das File entfernt und nicht mehr im System angezeigt.

Kalender

Dieses Modul stellt die Funktionalität eines Kalenders dar. Es ist somit möglich, Einträge zu erstellen, zu bearbeiten und zu entfernen. Beim Betreten der Seite wird standardmäßig die Wochenansicht dargestellt. Mit den beiden Pfeilen links oben kann im Kalender zurück- bzw. vorgeblättert werden. Soll die Ansicht auf Monate oder Tage geändert werden, so ist dies rechts oben durch die Schaltflächen *month* und *day* implementiert. Der *today-Button* ermöglicht das sofortige Zurückspringen zum aktuellen Datum.

Die Einträge können mittels Drag and Drop verschoben werden. Dies hat zur Folge, dass auch Zeit und Datum entsprechend geändert werden.

Um die Dauer eines Events zu verändern, kann dies direkt im Kalender durch Verkleinern bzw. Vergrößern des Eintragsfeldes vorgenommen werden.



Abbildung 4.13: Kalenderleiste

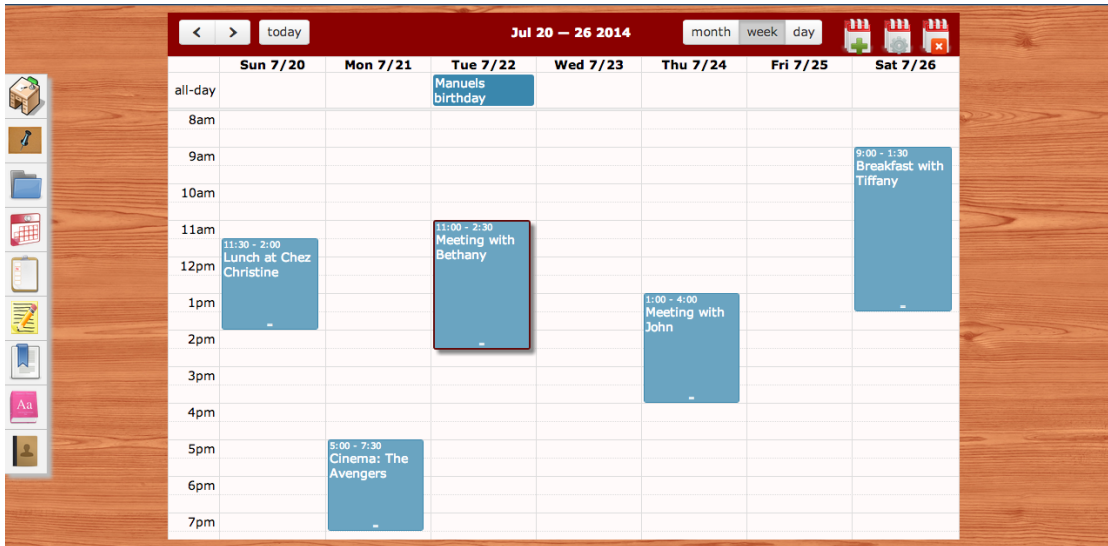


Abbildung 4.14: Kalender Wochenübersicht

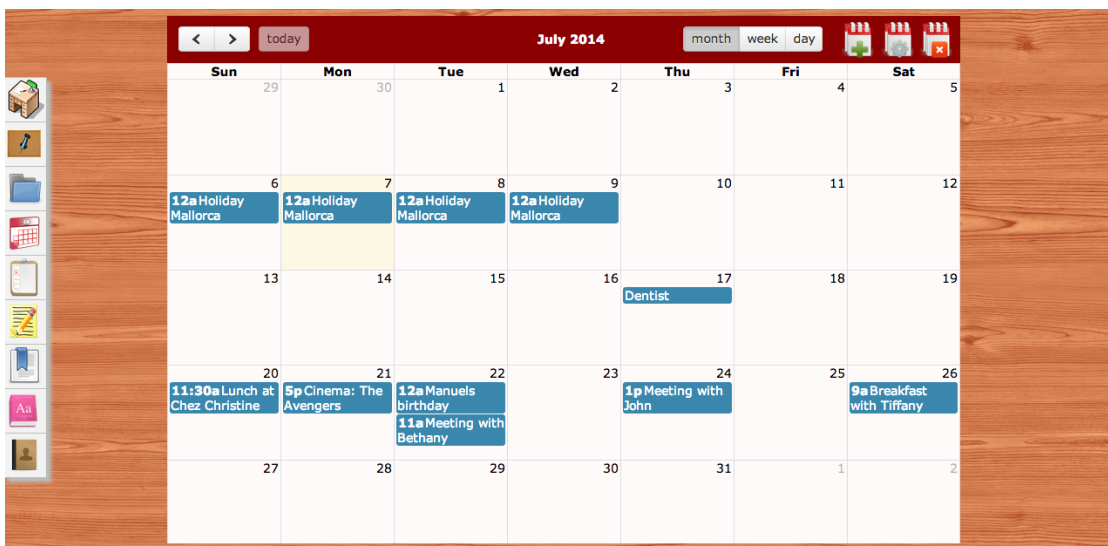


Abbildung 4.15: Kalender Monatsübersicht

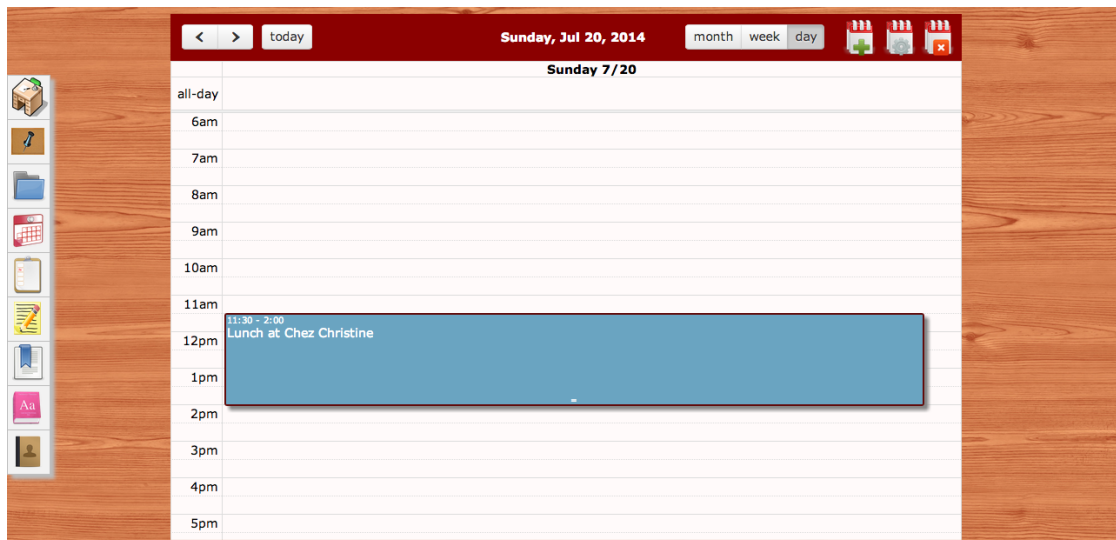


Abbildung 4.16: Kalender Tagesansicht

Eintrag erstellen

Durch Klicken auf das *Add Entry-Icon* wird ein Dialog geöffnet, in dem der Name, das Datum und die Uhrzeit des neuen Ereignisses eingegeben werden können. Durch Bestätigung durch den *Add-Button* wird der Eintrag hinzugefügt und an der entsprechenden Stelle im Kalender angezeigt. Eine Modal message signalisiert zusätzlich die erfolgreiche Erstellung.

Um nicht das *Add Entry-Icon* benutzen zu müssen, kann ein neues Event auch direkt über einen Klick zum Kalender hinzugefügt werden. Standardmäßig wird das Ereignis an der Stelle und somit zu der Uhrzeit erstellt, an welcher der Klick erfolgt.

Eintrag bearbeiten

Das Bearbeiten eines Eintrags ist durch Aktivierung des *Edit Entry-Icons* möglich. Es wird ein Dialog geöffnet, in welchem der Name, das Datum und die Uhrzeit geändert werden können. Der Vorgang wird durch Drücken des *Edit-Buttons* beendet und durch eine Modal message bestätigt.

Alternativ kann durch einen Doppelklick auf das gewünschte Ereignis auch der Dialog geöffnet werden, um das Element zu editieren.

Eintrag löschen

Soll ein Eintrag gelöscht werden, muss zuerst das *Delete Entry-Icon* gedrückt werden. Danach erscheint ein Dialog, in dem diese Aktion bestätigt werden muss. Es wird zusätzlich davor gewarnt, dass das Ereignis gelöscht wird und dass diese Handlung nicht rückgängig gemacht werden kann. Wird der *Delete-Button* betätigt, so wird das Ereignis aus dem Kalender entfernt.

Durch einen Doppelklick auf das entsprechende Element kann es ebenfalls gelöscht werden.

Sowohl das *Delete Entry-Icon* als auch das *Edit Entry-Icon* sind nur aktiviert, wenn ein Ereignis ausgewählt wurde. Andernfalls sind diese Icons deaktiviert. Ein Tooltip erteilt die Information, dass zuerst ein Element ausgewählt werden muss um es zu modifizieren oder zu entfernen.

Aufgaben

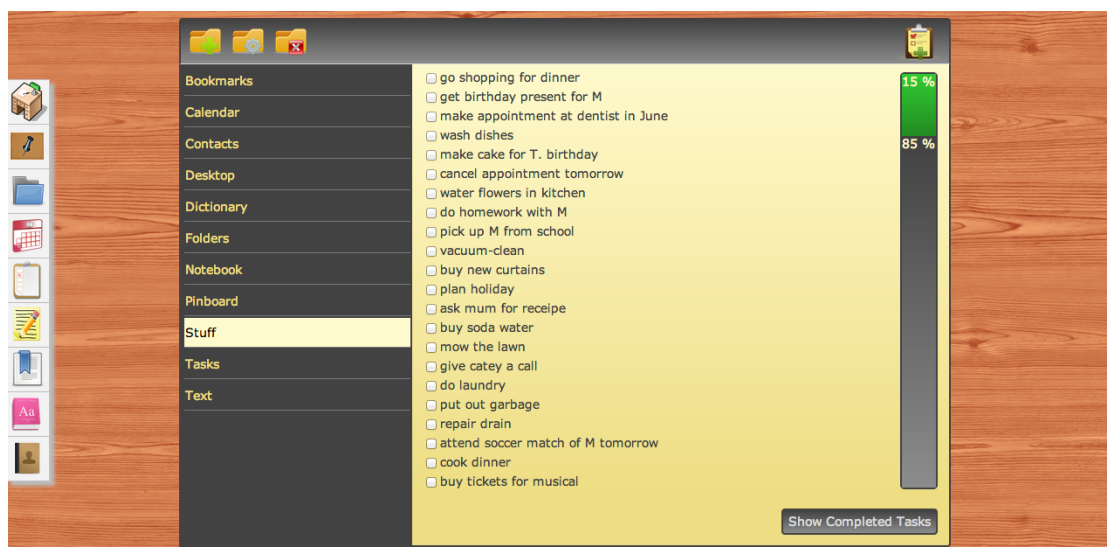


Abbildung 4.17: Aufgaben

Das Modul Tasks simuliert eine Aufgabenliste. Die Hauptfunktion ist das Anlegen von Tasks, die Kategorien zugeordnet werden können um einen besseren Überblick zu erhalten. Im linken Bereich sind alle verfügbaren Kategorien in alphabetischer Reihenfolge

ge angeordnet. Wird eine solche ausgewählt, erscheinen die zugehörigen Aufgaben auf der rechten Seite. Standardmäßig werden nur alle unerledigten Tasks angezeigt. Wenn eine Tätigkeit fertig gestellt wird, kann diese angehakt werden um deren Vollständigkeit zu signalisieren.

Sollen auch bereits erledigte Aufgaben dargestellt werden, ist dies durch Betätigen des *Show Completed Tasks-Buttons* möglich. Um einen zusätzlichen Überblick über den Fortschritt zu geben, zeigt eine kleine Graphik am rechten Rand den Anteil an fertigen und nicht-fertigen Aufgaben graphisch an. Außerdem wird prozentuell dargestellt, wie viele der Tätigkeiten bereits gelöst (grüner Bereich) und noch zu erledigen (grauer Bereich) sind. Sind noch überhaupt keine Aufgaben verfügbar, so wird anstelle der Aufgabenliste eine Information angezeigt, dass noch keine Tasks vorhanden sind und wie welche erstellt werden können. Außerdem wird die Status Graphik nicht dargestellt.



Abbildung 4.18: Aufgaben ohne verfügbare Tasks

Kategorie erstellen

Durch Klicken auf das *Add Folder-Icon* wird eine neue Kategorie mit dem gewünschten Namen erzeugt. Der Cursor ist bereits im Namensfeld positioniert, um ein sofortiges Tippen zu ermöglichen. Nach Bestätigung durch den *Create-Button* wird die Aufgabe erstellt und im rechten Bereich zur zugehörigen Kategorie hinzugefügt. Eine Modal message signalisiert zusätzlich die erfolgreiche Erstellung des Tasks.

Kategorie bearbeiten

Soll der Name einer Kategorie geändert werden, so geschieht dies durch den *Edit Category-Dialog*. Davor muss das entsprechende Element ausgewählt werden. Der Name des zu verändernden Ordners wird per default im Inputfeld angezeigt, damit Änderungen leichter vorgenommen werden können und außerdem der Name der zu bearbeitenden Kategorie nicht vergessen wird. Nach Bestätigung durch den *Edit-Button* wird der neue Name der Kategorie in der Liste angezeigt.

Kategorie löschen

Wird gewünscht dass eine Kategorie gelöscht wird, ist dies durch einen Klick auf das *Delete Category-Icon* möglich. Zur Absicherung erscheint ein Dialog mit einer Warnung, dass die markierte Kategorie gelöscht wird und diese Aktion nicht rückgängig gemacht werden kann. Außerdem werden auch alle Aufgaben entfernt, welche dieser Kategorie zugeordnet sind. Wird die Aktion durch den *Delete-Button* bekräftigt, so wird der Ordner nicht mehr in der Liste angezeigt.

Aufgabe erstellen

Soll eine neue Aufgabe kreiert werden, wird dies durch das *Add Task-Icon* ermöglicht. Nach der Eingabe des Texts wird der Task angezeigt und der aktuellen Kategorie zuge-wiesen.

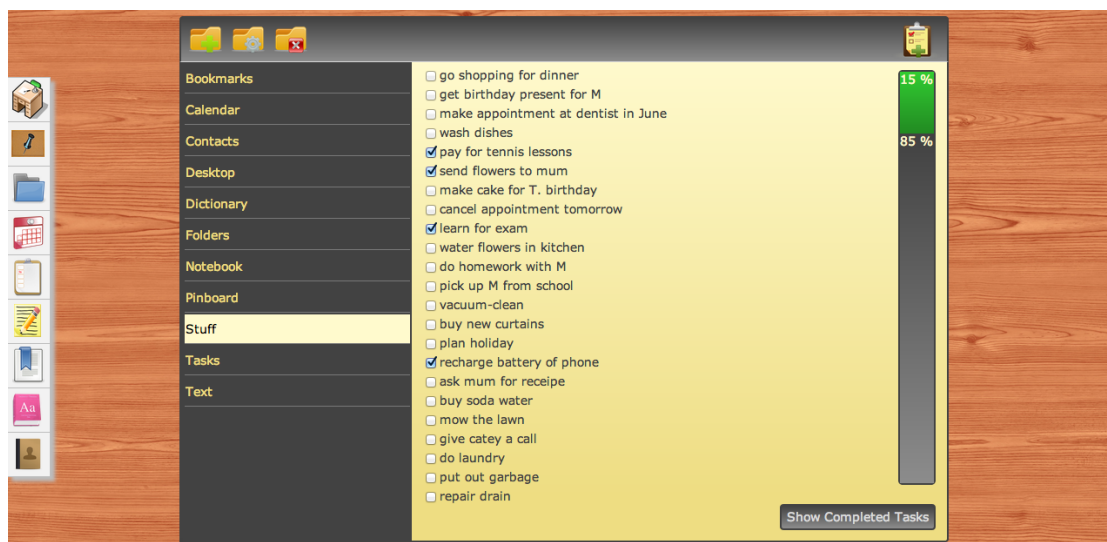


Abbildung 4.19: Erledigte Aufgaben anzeigen

Erledigte Aufgaben anzeigen

Wird der *Show Completed Tasks-Button* betätigt, werden auch bereits erledigte Aufgaben dargestellt. Diese werden durch ein Häkchen in der Checkbox von den unerledigten Tasks abgegrenzt.

Notizzettel

Der Hauptzweck des Notizzettels ist das spontane Verfassen von Text. Beliebige lange Notizen können ohne Formatierungen festgehalten werden.

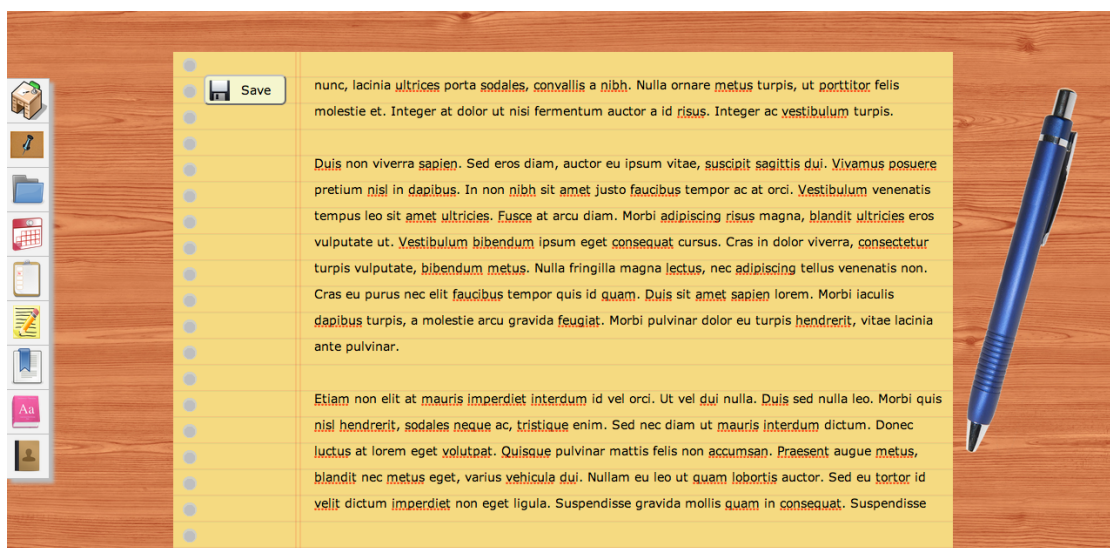


Abbildung 4.20: Notizzettel

Speichern

Durch Drücken des *Save-Buttons* wird der gesamte Inhalt gespeichert.

Lesezeichen

Das Modul Lesezeichen funktioniert ähnlich wie bekannte Lesezeichen-Manager in einem Browser. Es können verschiedene Bookmarks abgelegt werden, die bestimmten Kategorien zugeordnet werden. Diese sind auf der linken Seite angeordnet und können mittels drag- and drop in Unterkategorien angeordnet werden. Auf der rechten Seite sind alle Lesezeichen der ausgewählten Kategorie mit Name, Beschreibung und URL ersichtlich.

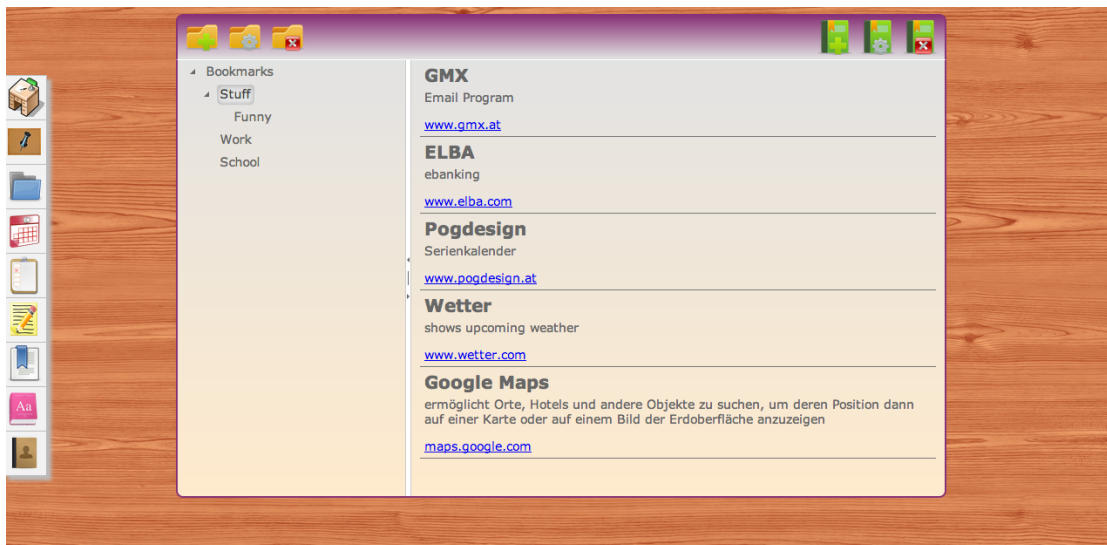


Abbildung 4.21: Lesezeichen

Sind in einer Kategorie noch keine Bookmarks verfügbar, wird dieser Umstand textuell vermerkt.



Abbildung 4.22: Lesezeichen ohne Daten

Kategorie erstellen

Damit eine neue Kategorie erstellt werden kann, muss zuerst das *Create Category-Icon* betätigt werden. Darauf folgend wird ein Dialog geöffnet, in welchem der Name eingetragen werden muss. Dieses Feld ist sogleich im Fokus, damit das Tippen ohne Umwege möglich ist. Durch Betätigung des *Create-Buttons* wird die Kategorie erstellt und auf der linken Seite angezeigt. Eine Modal message bestätigt diesen Vorgang.

Kategorie bearbeiten

Soll eine Kategorie editiert werden, ist das Drücken des *Edit Category-Icons* erforderlich. Es erscheint ein Dialog, in welchem der neue Name der Kategorie eingegeben werden kann. Der alte Name wird per default angezeigt um ein leichteres Bearbeiten zu ermöglichen und sicher zu gehen, dass AnwenderInnen nicht vergessen, welche Kategorie gerade überarbeitet wird. Nach der Dateneingabe und der Bestätigung mittels des *Edit-Buttons* ist der Vorgang abgeschlossen. Eine Modal message signalisiert erneut die erfolgreiche Änderung und der neue Name der Kategorie wird auf der linken Seite angezeigt.

Kategorie löschen

Um eine Kategorie zu löschen, muss das *Delete Category-Icon* betätigt werden. Anschließend wird ein Dialog angezeigt, der vor dem Entfernen der Kategorie warnt und eine Bestätigung erfordert. Wird dieser Schritt gemacht, so ist die Kategorie mit allen darin enthaltenen Lesezeichen gelöscht und wird nicht mehr in der Liste auf der linken Seite dargestellt.

Lesezeichen erstellen

Wenn ein neues Lesezeichen erstellt werden soll, muss zuerst das *Create Bookmark-Icon* gedrückt werden. Anschließend können die Daten eingegeben werden. Der Name und die Kategorie sind wie durch einen Stern gekennzeichnet obligatorisch, die anderen Felder sind optional. Der Cursor fokussiert automatisch das Namensfeld, um sofort tippen zu können. Die weitere Navigation kann entweder via Maus oder Tastatur erfolgen. Weiters ist die aktuelle Kategorie bei der Erstellung vorselektiert, was jedoch leicht über die drop-down Liste geändert werden kann. Nach Bestätigung durch den *Create-Button* wird das Lesezeichen erstellt und in der Liste angezeigt. Um zusätzliches Feedback zu geben, erscheint eine Modal message mit der Nachricht, dass das Lesezeichen erfolgreich erstellt wurde.

Lesezeichen bearbeiten

Das Editieren eines selektierten Lesezeichens erfolgt durch das Betätigen des *Edit Bookmark-Icons*. Es wird ein Dialog geöffnet, in welchem alle Eigenschaften verändert werden können, wobei die alten Werte als Hilfestellung angezeigt werden. Die Navigation durch den Dialog erfolgt gleich wie beim Erstellen der Lesezeichen. Nach der Bestätigung durch den *Edit-Button* wird das Lesezeichen erfolgreich geändert und mit den entsprechenden Eigenschaften in der Liste angezeigt. Zusätzlich erscheint eine Modal message, damit gekennzeichnet wird dass der Vorgang erfolgreich war.

Lesezeichen löschen

Das Löschen von Bookmarks ist möglich, in dem das *Delete Bookmark-Icon* gedrückt wird und davor das gewünschte Element markiert wird. Der Dialog zeigt eine Warnung an, dass das Lesezeichen gelöscht wird und diese Aktion nicht mehr rückgängig gemacht werden kann. Wird der *Delete-Button* betätigt, so wird der Löschvorgang durchgeführt. Zur Bestätigung wird das Lesezeichen nicht mehr in der Liste angezeigt und zusätzlich wird eine Modal message mit entsprechendem Text angezeigt.

Wörterbuch



Abbildung 4.23: Wörterbuch

Das Wörterbuch ermöglicht es, verschiedenste Begriffe zu erklären, um diese jederzeit nachschlagen zu können. Die linke Hälfte zeigt eine Übersicht aller Ausdrücke in alphabetischer Reihenfolge. Wenn ein Letter ausgewählt wird, werden alle Begriffe mit dem jeweiligen Anfangsbuchstaben dargestellt. Ist noch kein Eintrag verfügbar, erscheint ein Text, der diese Tatsache beschreibt und eine Hilfestellung gibt, wie neue Elemente erstellt werden können. Die Beschreibung eines Eintrags wird auf der rechten Seite sichtbar, wenn dieser ausgewählt wird. Dieser Teil ist ein Editor, damit der Inhalt beliebig gestaltet werden kann. Es sind folgende Formatierungen zulässig: Ändern der Schriftart, Schriftgröße, Schriftschnitte, Schriftfarbe und Hintergrundfarbe, Suchen und Ersetzen, Aufzählungen und Nummerierungen und Verändern des Einzuges. Weiters können Bilder, Links, Tabellen, spezielle Zeichen und Smileys hinzugefügt werden.

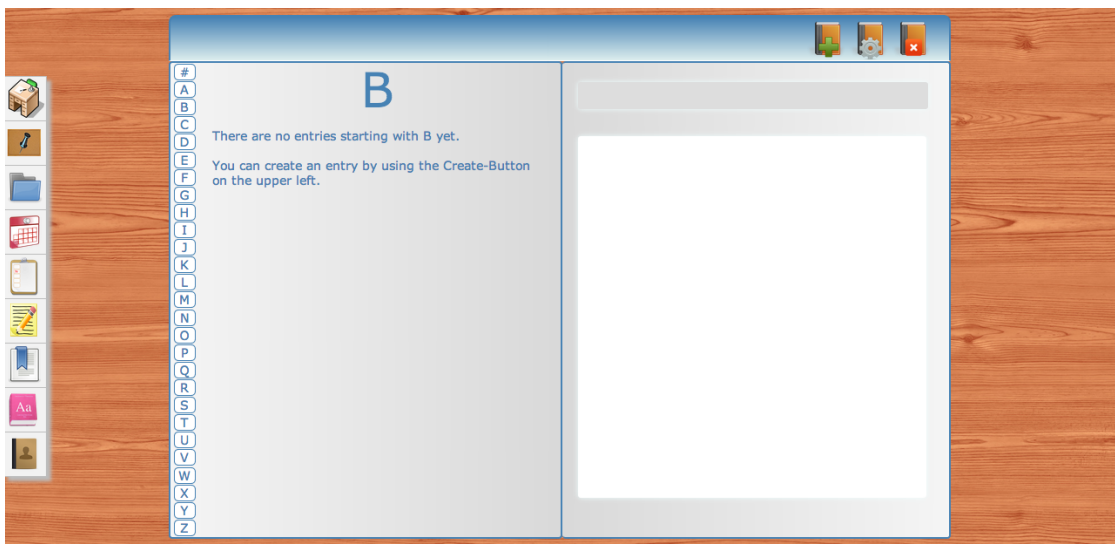


Abbildung 4.24: Wörterbuch ohne Daten

Eintrag hinzufügen

Um einen neuen Eintrag zu erstellen, muss das *Create Entry-Icon* gedrückt werden. Es erscheint ein leeres Formular auf der rechten Seite, in welchem der Name des Begriffs und die Beschreibung eingegeben werden können. Der Cursor ist im Namensfeld positioniert. Sobald ein Text im Editor vorhanden ist, kann dieser auch formatiert werden. Dazu werden wie bei Microsoft Word oder ähnlichen Programmen Passagen markiert und anschließend gestaltet. Ist die Beschreibung des Begriffes fertig gestellt, wird durch Betätigen des *Save-Buttons* gespeichert. Eine Modal message bestätigt den Vorgang.

Eintrag bearbeiten

Durch einen Klick auf das *Edit Entry-Icon* kann ein selektierter Begriff bearbeitet werden. Ähnlich wie beim Erstellen wird der Texteditor aktiviert und es können sämtliche Formatierungen vorgenommen werden. Durch den *Save-Button* wird die Handlung bestätigt. Eine Modal message gibt zusätzlich Rückmeldung zum erfolgreichen Vorgang.

Eintrag löschen

Soll ein markierter Begriff gelöscht werden, ist dies durch das *Delete Entry-Icon* möglich. Es erscheint ein Dialog, welcher vor dem Entfernen warnt und zusätzlich eine Bestätigung erfordert. Wird dies gemacht, so ist der Eintrag gelöscht und wird nicht mehr in der Liste angezeigt.

Telefonbuch



Abbildung 4.25: Telefonbuch

In diesem Modul ist es möglich, die persönlichen Kontakte zu verwalten. Auf der linken Seite befindet sich eine Übersicht aller verfügbaren Kontakte in alphabetischer Reihenfolge. Wird ein Buchstabe selektiert, so ändert sich die Anzeige und es werden nur noch Namen mit dem jeweiligen Anfangsletter dargestellt. Ist noch kein Eintrag zu dem Buchstaben vorhanden, wird dies textuell erklärt und ein Hinweis auf die Erstellung gegeben. Die Details eines Kontaktes können durch Auswählen eines Namens

eingesehen werden. Folgende Felder sind möglich: Name, Straße, Postleitzahl, Stadt, Land, private Telefonnummer, berufliche Telefonnummer, Email-Adresse, Geburtstag und Notizen. Alle Daten außer dem Namen sind optional und müssen daher nicht ausgefüllt werden. Ohne Betätigen des Bearbeiten-Buttons sind alle Felder deaktiviert, um nicht versehentlich etwas zu verändern. Beim Betreten dieses Moduls wird per default der erste Kontakt des Buchstabens A mit den jeweiligen Details dargestellt.

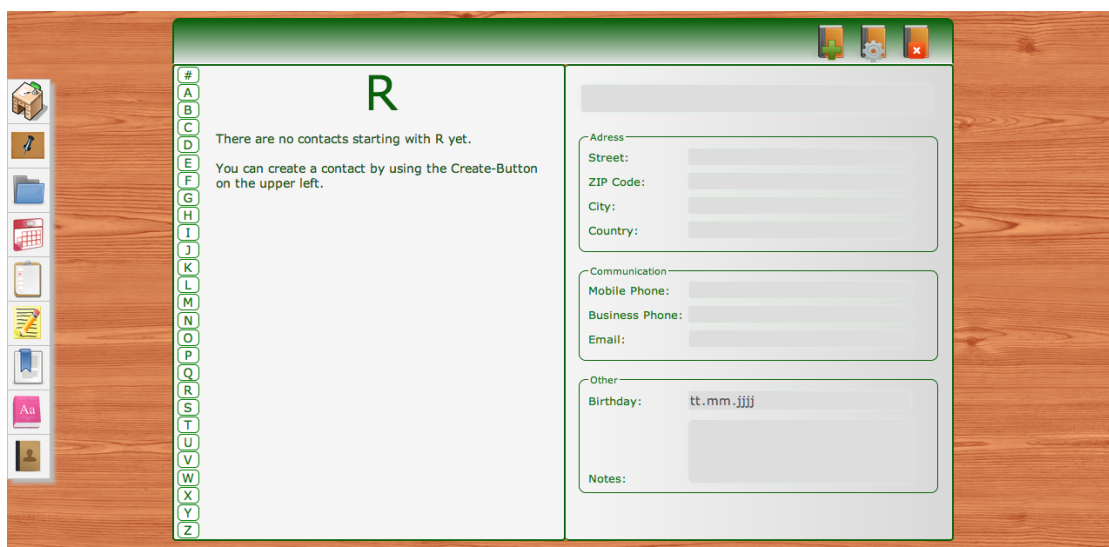


Abbildung 4.26: Telefonbuch ohne Daten

Kontakt hinzufügen

Wird das *Create Entry-Icon* betätigt, so erscheint ein leeres, editierbares Formular auf der rechten Seite. Der Cursor ist auf das Namensfeld gerichtet, um sofort mit dem Tippen beginnen zu können. Es ist außerdem möglich, nur mit der Tastatur durch alle Felder zu navigieren, damit nicht immer zwischen Keyboard und Maus gewechselt werden muss. Durch Betätigen des *Save-Buttons* wird der Eintrag gespeichert, was zusätzlich durch eine Modal message bestätigt wird. Weiters werden nun alle Kontakte mit dem gerade erstellten Buchstaben angezeigt.

Kontakt bearbeiten

Um einen Eintrag zu bearbeiten, muss dieser zuerst selektiert werden. Anschließend wird das *Edit Entry-Icon* betätigt, damit alle Datenfelder aktiviert werden und somit editierbar sind. Jedes Element kann verändert werden. Sind die Änderungen fertig ge-

stellt, wird der Vorgang durch den *Save-Button* abgeschlossen und einer Modal message bestätigt.

Kontakt löschen

Soll ein Eintrag gelöscht werden, muss dieser markiert werden. Nach Betätigen des *Delete Entry-Icons* erscheint ein Dialog, in welchem erneut vor dem Löschen gewarnt wird und der Vorgang bestätigt werden muss. Anschließend ist der Kontakt entfernt und scheint nicht mehr in der Liste auf.

Sonstige Patterns

Nachfolgend werden allgemeine Muster beschrieben, die im ganzen System angewendet werden und nicht einem bestimmten Modul zugeordnet sind.

Aufbau der Module

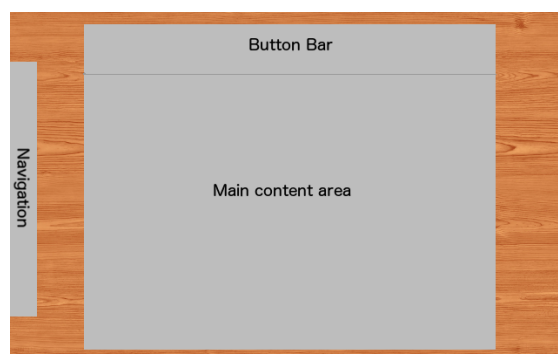


Abbildung 4.27: Aufbau der Module

Der Aufbau der Module ist meist sehr ähnlich gestaltet, um die Konsistenz zu fördern und AnwenderInnen nicht zu verwirren.

Am linken Rand des Bildschirms befindet sich die vertikale Navigation. Diese ist wenn vorhanden überall an der selben Stelle positioniert. In der Mitte befindet sich der Inhalt des Moduls. Am oberen Rand sind die Buttons angeordnet, welche verschiedene Aktionen nach sich ziehen. Meistens sind dies das Erstellen, Editieren und Löschen von Inhalten. Darunter befindet sich der Hauptteil des jeweils aktuellen Features, welcher den Content enthält.

Navigation

Die Navigation ist vertikal angeordnet und auf jeder Seite an der gleichen Stelle am linken Fensterrand positioniert. Berührt der Cursor einen Menüpunkt, so wird der Hintergrund des Moduls hervorgehoben und auch der Name als Text angezeigt. Andernfalls werden nur die entsprechenden Icons dargestellt.

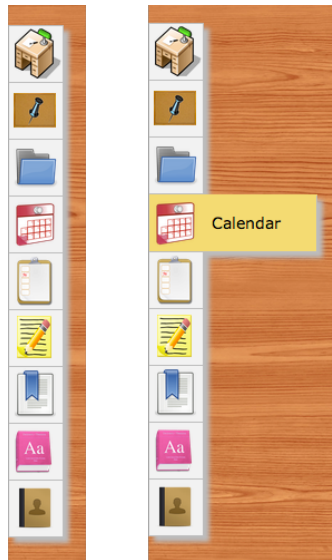


Abbildung 4.28: vertikale Navigation

Suche

Die Suche ist eine der wichtigsten Funktionen in Webapplikationen und muss genau durchdacht werden. Das Suchfeld wurde rechts oben im System angeordnet und besteht aus einem Suchfeld und einem Button.

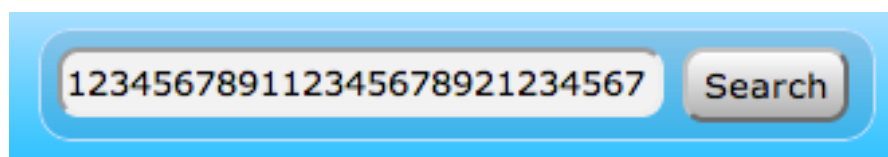


Abbildung 4.29: Suchfeld

Wird der *Search-Button* oder die *Enter-Taste* betätigt, erscheint eine neue Seite mit den Suchergebnissen. Das Suchfeld mit der Anfrage wird dabei erneut gezeigt, falls ein an-

derer Suchvorgang durchgeführt werden soll. Zusätzlich wird die Anzahl der gefundenen Treffer dargestellt.

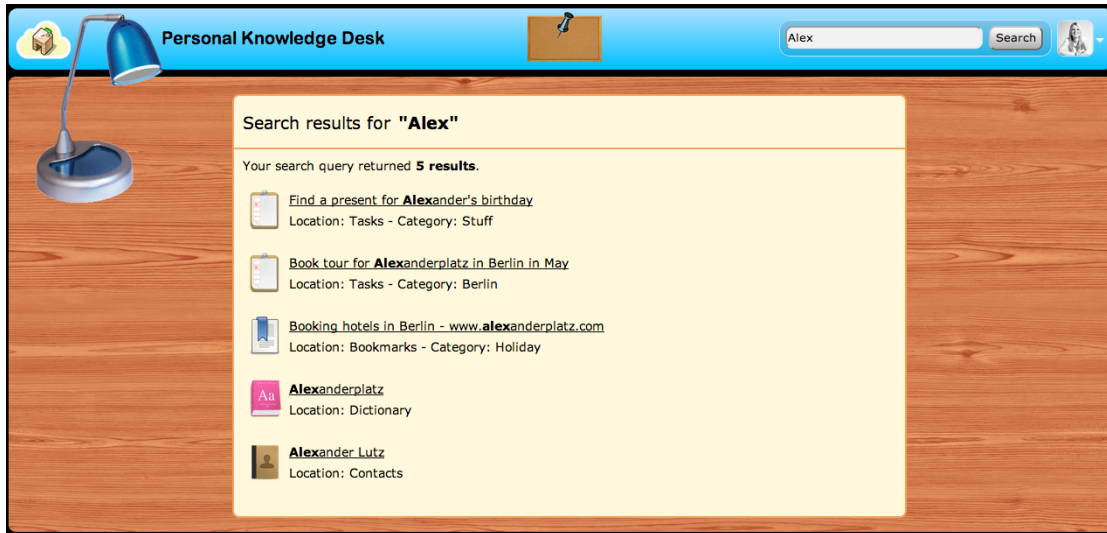


Abbildung 4.30: Suche mit Ergebnissen

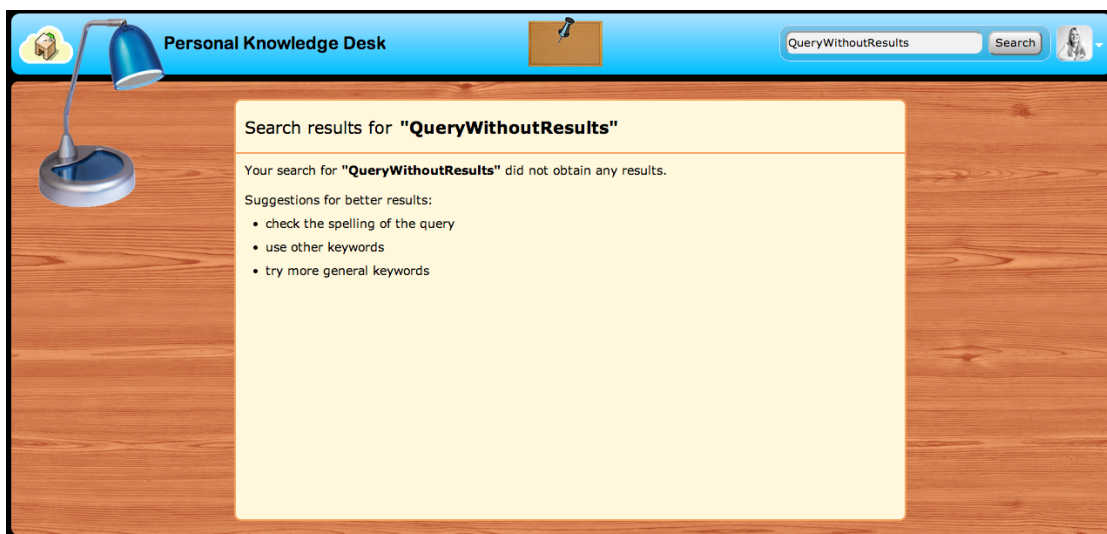


Abbildung 4.31: Suche ohne Ergebnisse

Der Hauptteil der Seite zeigt das Icon, in welchem Modul sich der Treffer befindet, den betreffenden Text wo der Suchtext enthalten ist sowie die Location, wo der Treffer

erzielt wurde. Wenn ein Text ausgewählt wird, erfolgt die Navigation zu dem Treffer im entsprechenden Modul. Die Reihenfolge der Ergebnisse ist analog zu den Elementen in der vertikalen Navigation, da diese nach der Wichtigkeit geordnet sind.

Wird kein Treffer gefunden, so erscheint wiederum die Ergebnisseite. Der Header ist gleich, allerdings werden im Hauptteil Tipps gegeben, wie die Suche optimiert werden kann und somit mehr Übereinstimmungen erzielt werden können.

Tooltips

Tooltips sind kleine Boxen mit Informationen, welche nur gezeigt werden wenn die Maus bestimmte Elemente berührt. Sie verbessern die Benutzerfreundlichkeit, da es möglich ist textuelle Beschreibungen oder Hilfestellungen zu geben, aber nur wenn diese auch benötigt werden.

Beim Prototyp wurden Tooltips bei allen Icons und auch bei anderen Elementen implementiert.

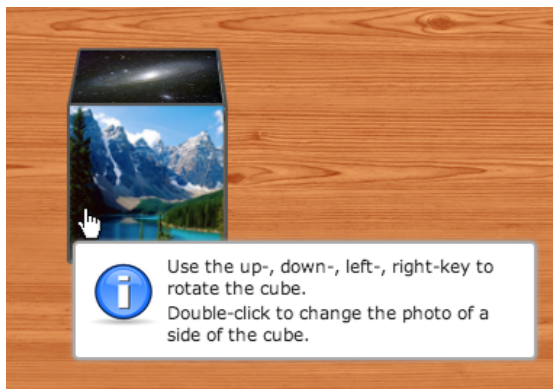


Abbildung 4.32: Tooltip beim Fotowürfel

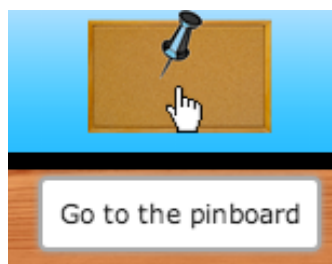


Abbildung 4.33: Tooltip bei Icons

Buttons

Buttons sind unverzichtbare Elemente eines Systems um entsprechende Aktionen auszuführen. Beim Personal Knowledge Desks wurden zwei verschiedene Designs von Buttons verwendet.

- **Icons:** Für *Create*-, *Edit*- und *Delete-Operationen* werden Icons verwendet, um die jeweiligen Aktionen durchzuführen. Das Aussehen wurde so designed, dass beim ersten Blick offensichtlich ist, wofür die Icons stehen. Bei jedem Modul wurden die Icons konsistent verwendet und für speziellere Zwecke angepasst. Außerdem wurden Tooltips implementiert, um eine zusätzliche Hilfestellung zu geben.
- **Buttons:** Einzelne Aktionen werden nicht als Icon sondern als konkrete Buttons implementiert. Diese enthalten sowohl den Text als auch ein Bild, um den Zweck noch besser darzustellen.



Abbildung 4.34: Icons und Buttons

Dialoge

Dialoge werden meist für das Erstellen, Bearbeiten oder Löschen von Elementen verwendet und sind immer gleich aufgebaut. Der Titel beschreibt das Thema der nachfolgenden Aktion, wie zum Beispiel "Create Bookmark". Dadurch soll erneut klar gemacht werden, welche Aktion gerade durchgeführt wird. Im Hauptteil sind die Felder angeordnet, welche je nach Zweck variieren. Den Abschluss bildet die Button-Bar, wo immer genau zwei Buttons angeordnet sind. Auf der rechten Seite ist der *Cancel-Button*, um den Dialog zu schließen und die Aktion zu verwerfen. Der linke Button ist je nach Zielsetzung entweder *Create*, *Edit* oder *Delete*.

Create Bookmark

Name*
TV Calendar

Description
This calendar shows all upcoming TV shows with a description, a trailer and the upcoming date

Url
http://www.pogdesign.co.uk/cat/

Folder*
Stuff

Create Cancel

Abbildung 4.35: Dialog

Modal messages

Um den AnwenderInnen Feedback über Ereignisse zu geben, werden Modal messages verwendet. Diese werden angezeigt, sobald ein bestimmter Schritt geschehen ist. Beim Prototyp werden zwei verschiedene Arten verwendet: Eine grüne Box induziert, dass eine Aktion erfolgreich beendet wurde, wie zum Beispiel "The task was added successfully." Somit wird UserInnen signalisiert, dass die Aufgabe erfolgreich erstellt wurde und keine weitere Tätigkeit mehr notwendig ist. Die zweite Art ist eine rote Box, welche ankündigt, dass etwas nicht in Ordnung ist. Zum Beispiel sagt folgender Text "Please select the type of element you want to add." aus, dass nichts zur Pinnwand hinzugefügt worden ist, da noch keine Auswahl getroffen wurde. Somit wird klar gestellt, dass Anwenderinnen zuerst ein Element auswählen müssen, um es an die Pinnwand zu hängen.

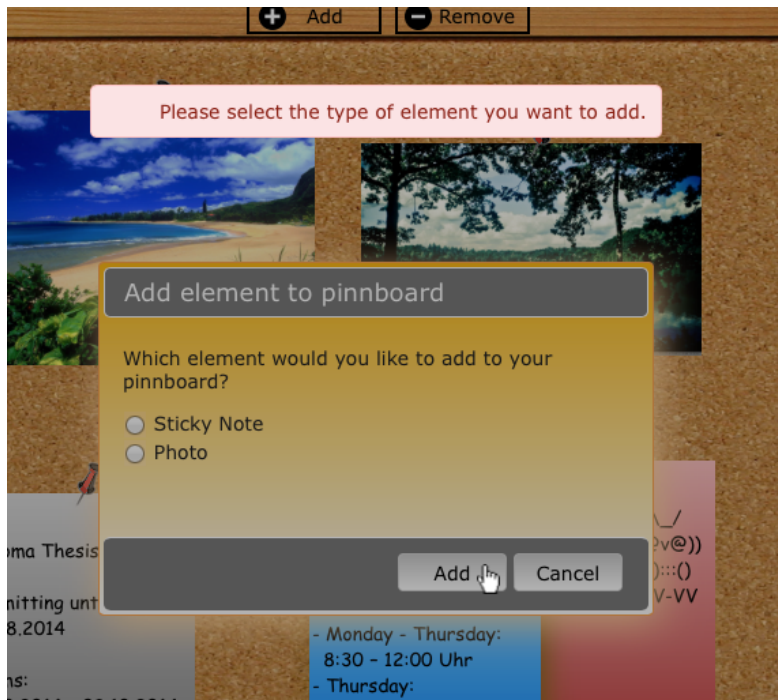


Abbildung 4.36: Modal message, die einen Error beschreibt

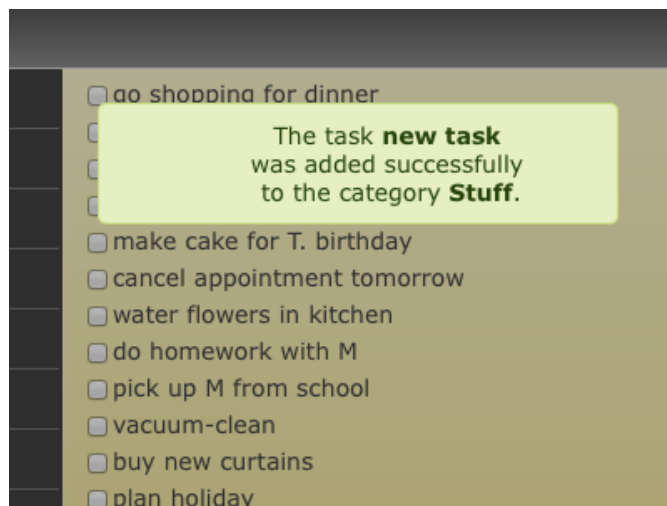


Abbildung 4.37: Modal message, die eine erfolgreiche Aktion signalisiert

Evaluierung

Checklisten, spezielle Evaluations-Tools und Fragebögen sind nur einige von vielen Möglichkeiten, um die Usability von Systemen zu testen [1]. Unterschieden wird dabei grundsätzlich, ob BenutzerInnen bei dem Test miteinbezogen werden oder nicht, wobei ersteres vorzuziehen ist, weil dadurch ein persönlicheres Feedback gewonnen wird.

Zur Evaluierung des Personal Knowledge Desks wurde einerseits eine Evaluierung anhand von Heuristiken und andererseits Thinking-Aloud-Usability Testing gewählt. Durch Anwendung von zwei verschiedenen Methoden soll sicher gestellt werden, dass so viele Schwachstellen wie möglich erkannt werden.

5.1 Evaluierung anhand von Usability Heuristiken

Eine heuristische Evaluation [32] hinterfragt die Benutzerfreundlichkeit eines Systems und bewertet ein bestimmtes Produkt anhand von Guidelines. Obwohl ExpertInnen das Programm begutachten, wird aus der Sichtweise der potentiellen Zielgruppe begutachtet. Das Ziel ist eine Liste von verschiedensten Usability Problemen, die laut ExpertInnen gegen diese Heuristiken verstoßen.

Die angewendeten Guidelines stammen von Jacob Nielsen, welche als die zehn allgemeinsten Prinzipien bezüglich des Interaction Designs gelten [19]. Diese wurden original 1990 verfasst und später basierend auf 249 Usability Problemen verfeinert, um abstrakte, mächtige Heuristiken zu erschaffen. Im Folgenden werden die zehn Guidelines angeführt und diskutiert, ob sie beim Personal Knowledge Desk zutreffen oder ob Verbesserungsbedarf herrscht.

1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Diese Richtlinie trifft vollkommen zu, da UserInnen zu jedem Zeitpunkt wissen, wo sie sich gerade im System befinden. Außerdem geben Modal messages unmittelbar nach jedem Ereignis Auskunft über den Status der getätigten Aktionen.

2. Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Diese Guideline wurde erfolgreich umgesetzt. Alle Beschriftungen, Rückmeldungen oder anderen sprachlichen Konstrukte sind in einfachem Englisch gehalten und es wird niemals Code angezeigt. Weiters werden Nachrichten stets direkt nach entsprechenden Aktionen angezeigt. Ist die Reihenfolge der Elemente von Bedeutung, wurde immer darauf geachtet, dass sie logisch angeordnet sind.

3. User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Diese Heuristik wurde nur teilweise eingehalten, weil kein "Rückgängig machen" oder "Wiederherstellen" implementiert wurde. Jedoch ist es nur schwer möglich, überhaupt Fehler zu machen die eine solche Aktion benötigen würden. Außerdem ist es von allen Punkten im Programm aus möglich, zu den anderen Inhalten zu navigieren.

4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Diese Richtlinie wurde zu großen Teilen erfüllt. Die selben Ereignisse rufen immer die selben Konsequenzen hervor und auch das Wording ist überall gleich. Allerdings ist

das Erstellen, Editieren und Löschen von Inhalten bei fast allen Modulen über Dialoge gelöst, während die Kontakte und das Wörterbuch ohne solche auskommen. Diese Inkonsistenz könnte die BenutzerInnen verwirren.

5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Error prevention wurde relativ gut umgesetzt, da es nur wenige Situationen gibt, welche Fehler verursachen. Sollte ein Ereignis in einer bestimmten Situation nicht möglich sein, sind die entsprechenden Trigger deaktiviert. Weiters ist es beispielsweise bei Zahlenfeldern nicht möglich, Buchstaben einzugeben.

6. Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Diese Richtlinie ist vollständig erfüllt, da es keine versteckten Funktionen gibt. Alle möglichen Aktionen sind zu jeder Zeit ersichtlich.

7. Flexibility and efficiency of use

Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Es wurden keine Beschleunigungen wie zum Beispiel ein Context-Menü oder Shortcuts eingebaut. Daher wurde diese Guideline nicht umgesetzt.

8. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Die Dialoge im System enthalten nur die notwendigsten Informationen und sind textuell sehr kurz gehalten. Außerdem ist das Naming der Buttons ausreichend, um die Konsequenzen zu beschreiben.

9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Es gibt zwar nur vereinzelte Fehlermeldungen im System, diese sind jedoch in einfacher Sprache gehalten und beschreiben genau das Problem.

10. Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Die Erfüllung dieser Richtlinie kann nicht beantwortet werden, da keine Dokumentation existiert. Jedoch wurde das System sehr einfach gehalten, wodurch eine derartige Hilfestellung nicht unbedingt benötigt wird.

Die heuristische Evaluierung wurde nur von einer Person durchgeführt. Daher ist es eher nicht möglich, dass alle Schwachstellen bezüglich der Benutzerfreundlichkeit gefunden wurden. Aus diesem Grund besteht der Hauptteil der Evaluierung aus der Thinking Aloud Usability Testing Methode, welche auf den nächsten Seiten detailliert beschrieben wird.

5.2 Thinking Aloud Usability Testing

Diese Methode ist besonders gut geeignet, um hochwertiges Feedback von der Zielgruppe zu erhalten und damit eines der wertvollsten Usability Instrumente überhaupt. Die Durchführung klingt in der Theorie sehr einfach: UserInnen beschäftigen sich mit dem User Interface eines Systems und sprechen ihre Gedanken laut aus, während sie vorgegebene Aufgaben lösen. In der Praxis ist die Abwicklung nicht ganz so trivial, da es für Menschen eher unnatürlich ist, alles Gedachte zu erzählen.

Der größte Vorzug dieser Technik ist, dass durch die Überlegungen von AnwenderInnen gut und schnell erkannt werden kann, welche Probleme es beim Design gibt und wo noch Verbesserungsbedarf herrscht. Außerdem ist diese Methode [22]:

- **billig:** Es werden weder spezielle Einrichtungen noch besonderes Equipment benötigt. Es bedarf lediglich eines Computers (mit Internetverbindung) und eines Schreibtischs. InstrukteurInnen sitzen neben den TeilnehmerInnen und hören zu bzw. notieren sich die wichtigsten Erkenntnisse. Weiters muss nicht viel Zeit investiert werden, da meist ein Tag ausreicht um die Tests abzuhalten und die Daten auszuwerten.
- **robust:** Auch wenn die Methode nicht ganzheitlich korrekt durchgeführt wird, ist meist nur wenig Schaden angerichtet und man erhält trotzdem wertvolles Feedback.
- **flexibel:** Thinking-Aloud kann und soll bei jeder Phase im Softwareentwicklungszyklus eingesetzt werden. Überdies ist die Methode bei allen Typen von Software und Technologien einsetzbar.
- **überzeugend:** ManagerInnen bzw. EntwicklerInnen nehmen Usability Probleme teilweise nicht sehr ernst. Werden sie jedoch durch die Ergebnisse des Tests darauf aufmerksam gemacht, dass Teile des Systems von AnwenderInnen nur schwer verstanden werden, lassen sie sich eher dazu überreden, Ressourcen freizugeben und gewisse Probleme zu beheben.
- **leicht erlernbar:** Thinking Aloud besteht nur aus wenigen Schritten. Der Test kann daher leicht angewendet werden und bedarf keines besonderen Trainings.

Diese Methode ist jedoch nicht für alle Zwecke geeignet, da der Output keine detaillierten, statistischen Daten liefert und der Test außerdem in einem relativ unnatürlichen Setting stattfindet. Wendet man Thinking-Aloud jedoch mit der richtigen Intention an, wird das Ergebnis zufriedenstellend sein, da so am meisten Informationen über die Probleme der Benutzeroberfläche gewonnen werden können [38].

Mögliche Probleme bei der Durchführung

Bei der Durchführung dieser Methode gibt es zwei Arten von Feedback, wobei nur eine davon wünschenswert ist [13]:

- Rückmeldung über die Bedienung des Systems: Diese Informationen sind essenziell und erwünscht. Man erhält viel Feedback über die möglichen Probleme bei der Bedienung, wie zum Beispiel "Diese Aktion dauert länger als erwartet" oder "Ich drücke den Button und erwarte mir ein Popup".
- Rückmeldung zum Design: Diese Art der Rückmeldung ist nicht gerne gesehen, da nicht das Design, sondern die Benutzung evaluiert wird. Beispiele sind "Der Hintergrund sollte blau sein" oder "Ich würde den Button kleiner machen".

Um nun genau die erwünschte Art von Aufklärung zu erhalten, werden folgende Techniken vorgeschlagen:

- Instructive Practice: Vor dem Start einer Session wird die Methode detailliert erklärt und die erwünschten Informationen geschildert. Es empfiehlt sich außerdem, ein kurzes Beispiel zu geben um den TeilnehmerInnen die Funktionsweise ohne Druck zu verdeutlichen.
- Operant Conditioning
 - Reinforcement: BenutzerInnen sollen gelobt werden, wenn man das gewünschte Feedback erhält. Sätze wie "Sie machen das toll." oder "Diese Informationen helfen immens." motivieren und lenken in die richtigen Richtung.
 - Extinction: Wird unerwünschtes Feedback zum Design oder Ähnliches gegeben, sollen InstruktorenInnen diese Vorschläge ignorieren und mit der nächsten Aufgabe fortfahren. Somit werden ProbandInnen nicht ermutigt, diesen Einschlag weiter zu verfolgen.

Zusätzlich muss darauf geachtet werden, dass TeilnehmerInnen nicht unbewusst durch InstruktorenInnen beeinflusst werden. Daher sollen die Aufgaben objektiv gestellt werden. Weiters dürfen Fragen zur Benutzung des Systems während der Durchführung nicht beantwortet werden.

Auswahl der ProbandInnen

Es werden nur fünf TeilnehmerInnen benötigt, um das beste Kosten-Nutzen-Verhältnis zu erzielen [25]. Eine weitere Hinzunahme von ProbandInnen ist sogar eine Verschwendung von Ressourcen. Steht ausreichend Budget zur Verfügung, sollten eher verschiedene Methoden zur Rate gezogen werden im Gegensatz zu mehreren TeilnehmerInnen für denselben Test.

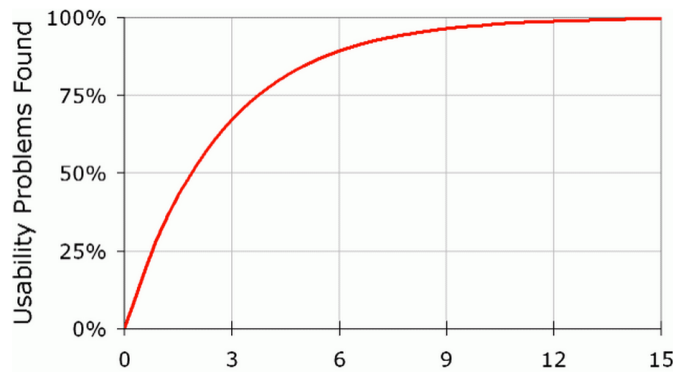


Abbildung 5.1: Anzahl der ProbandInnen vs. gefundene Usability Probleme

Bei der Auswahl der ProbandInnen gibt es einige Dinge zu beachten, die schon in zahlreichen Guidelines formuliert wurden [33].

Eine wichtige Entscheidung vor der Durchführung ist, wie viele Personen pro Test befragt werden sollen. Dabei gibt es zwei Möglichkeiten [37], die für die Evaluierung des Prototypen in Frage kommen:

- Concurrent thinking-aloud (CTA): Nur eine Testperson evaluiert das System. Alle Aufgaben müssen alleine bewältigt werden.
- Constructive interaction test (CI): Es werden zwei TeilnehmerInnen gleichzeitig befragt, welche als Team agieren und die Aufgaben gemeinsam lösen.

Van den Haak et. Al [37] führten eine Studie durch, um die zwei Arten zu vergleichen. Beide Gruppen führten Thinking-Aloud unter exakt denselben Bedingungen durch, um mögliche Beeinflussungen oder Veränderungen zu verhindern. Dabei kam heraus, dass die ProbandInnen des CTA vermehrt ihre Aktionen erklärten, wie zum Beispiel "Ich drücke auf diesen Button und schreibe das hinein". Weiters fühlten sich die TeilnehmerInnen selbst getestet, obwohl ausdrücklich erklärt wurde, dass das Produkt evaluiert wird und nicht sie selbst. Beim CI-Test wird ein eher natürlicheres Setting geschaffen, weil zwei Personen im Team arbeiten und sich so weniger beobachtet und getestet fühlen, da sie miteinander kommunizieren. Sie erklärten dabei weniger oft unnötige Schritte und haben sich sicherer gefühlt, obwohl sie die InstrukteurInnen ähnlich wie beim CTA als Aufsichtsperson wahr genommen haben.

Vorbereitung des Tests

Durch die oben beschriebenen Erkenntnisse wird für die Evaluierung des Prototyps der CI-Test mit fünf Zweier-Teams angewendet. Die Durchführung lehnt sich an die Vorgehensweise von Steve Krug [35] an.

Planung der Aufgaben

Am Beginn soll überlegt werden, welche Aufgaben den TeilnehmerInnen gestellt werden. Dazu wird eine Liste erstellt, in welcher alle möglichen Tasks beschrieben werden. Danach wird diese Liste nach den folgenden Kriterien gefiltert:

- Welche Aufgaben müssen unbedingt funktionieren (Kernfunktionalität)?
- Welche Aufgaben sind kritisch und legen den Verdacht nahe, dass sie noch nicht so funktionieren wie erwartet?
- Welche Aufgaben sind laut anderen Testmethoden noch wenig benutzerfreundlich?

Bei der Erstellung der Aufgaben ist zu beachten, dass die Suche nicht getestet werden soll. Dies hat den Grund, dass ProbandInnen ansonsten alle Aufgaben nur noch über die Suche erledigen wollen, was die Ergebnisse verändern würde.

Nach der Durchführung dieser Schritte ergeben sich die folgenden Aufgaben. Es wurde darauf geachtet, dass die Abarbeitung der Aufgaben zwischen 15 und 30 Minuten in Anspruch nimmt:

- Notizen am Notizblock machen
- zur Pinnwand navigieren und ein Post-it erzeugen
- einen speziellen Kalendereintrag modifizieren
- einen neuen Kontakt hinzufügen
- einen Eintrag im Wörterbuch optimieren
- ein Lesezeichen löschen
- ein Dokument in die Ordnerstruktur hinzufügen
- einen Task anlegen

Stehen die Aufgaben fest, so sollen daraus Szenarios entwickelt werden, um den AnwenderInnen den notwendigen Kontext zu geben, den sie für die Aufgaben benötigen. Dabei ist zu berücksichtigen, nur die essentiellsten Informationen weiter zu geben und keine Anhaltspunkte für den Lösungsansatz zu liefern. Für die Evaluierung des Prototyps wurde für jedes Feature ein Szenario erzeugt.



Abbildung 5.2: Aufgaben beim Usability Test

Aufgaben testen

Um zu vermeiden, dass die Aufgaben zweideutig oder schlecht beschrieben sind, wird ein kurzer Pilottest durchgeführt. Dazu wird einer Testperson (zum Beispiel einem Familienmitglied) das Tool präsentiert und anschließend die Szenarios vorgelesen. Danach wird begonnen, die Aufgaben zu lösen, welche aber nicht fertig gestellt werden müssen. So wird schnell erkannt, ob die Aufgaben klar gestellt sind.

Aufgaben ausdrucken

Nach der Fertigstellung der Szenarios sollen diese ausgedruckt werden. Den TeilnehmerInnen soll pro Szenario ein eigener Zettel ausgehändigt werden, um die Übersichtlichkeit zu steigern. Wird eine bestimmte Reihenfolge erwünscht, so können die Aufgaben durchnummeriert werden. Es soll der Sicherheit halber auch eine Kopie für die AufseherInnen angefertigt werden.

Protokoll verfassen

Besonders bei der ersten Durchführung von Usability-Tests ist es wichtig, sich davor ein Protokoll zurecht zu legen. Es empfiehlt sich sogar, während des Tests das in großer Schrift ausgedruckte Dokument vorzulesen und wenig bis gar nicht zu improvisieren. Auch wenn das laute Vorlesen eines Skripts unnatürlich erscheint hat es den Vorteil, dass nichts Wichtiges vergessen werden kann.

Durchführung des Tests

Zu Beginn der Testphase wurde ein Pilottest inszeniert, um zu überprüfen ob die Aufgaben verständlich formuliert und in der Zeit lösbar sind. Anschließend wurden wenige kleine Änderungen durchgeführt, um die Szenarios zu optimieren.

Die tatsächliche Realisierung der Tests erfolgte an verschiedenen Tagen mit fünf Zweier-Teams. Die jeweiligen Personen im Team hatten stets die selben Computerkenntnisse, um Schiebungen zu vermeiden. Obendrein wurden die ProbandInnen so ausgewählt, dass eine möglichst breite Streuung vorherrscht. Weil der Prototyp besonders für SeniorInnen entwickelt wurde, wurden 4 Personen dieser Zielgruppe in den Test mit einbezogen.

Die Teams wurden zwei Wochen vor dem jeweiligen Termin per Email benachrichtigt. Sie wurden weiters über den Zweck und die Dauer der Evaluierung unterrichtet. Jeder

Willkommen - ca. 3 Minuten

Hallo [Name der TeilnehmerInnen], mein Name ist Romana Hofbauer, und ich werde heute mit Ihnen den Thinking-Aloud-Usability Test durchführen.

Bevor wir anfangen, werde ich Ihnen noch kurz die Vorgehensweise erklären. Dieses Protokoll werde ich laut vorlesen, um sicherzustellen dass nichts vergessen wird.

Wie Sie bereits wissen, habe ich einen Prototypen einer Webapplikation für meine Diplomarbeit entwickelt. Ich möchte gleich zu Beginn klar stellen, dass das System getestet wird, und nicht Sie. Bei diesem Test können Sie absolut nichts falsch machen, jede Information ist für mich von Bedeutung. Während Sie das Programm benutzen, sollen Sie alle Ihre Gedanken laut aussprechen. Sie sollen mir erzählen, wo sie gerade hinsehen, was Sie versuchen zu machen und was Sie dabei denken. Ich werde Sie auch immer wieder daran erinnern, damit der Redefluss nicht anhält. Sie müssen dabei auch keine Angst haben, meine Gefühle zu verletzen, falls etwas nicht funktioniert oder Sie es sich anders vorgestellt hätten. Ich würde gerne das System verbessern, daher sind solche Aussagen sehr hilfreich.

Gibt es während des Tests Fragen, dann zögern Sie nicht, diese zu stellen. Meistens kann ich diese jedoch nicht sofort beantworten, da ich gerne wissen würde wie Sie das Programm benutzen, wenn keine Hilfe zur Verfügung steht. Gibt es zum Schluss immer noch Fragen, so freue ich mich natürlich diese zu beantworten. Falls Sie aus irgendwelchen Gründen eine Pause benötigen, können Sie das jederzeit sagen.

Gibt es bisher Fragen?

Fragen - ca. 3 Minuten

Bevor wir nun beginnen, würde ich Ihnen noch gerne einige Fragen stellen.

Zuerst würde ich gerne wissen, was Ihr Beruf ist?

Meine nächste Frage befasst sich mit Ihren Internetgewohnheiten. Wie viele Stunden denken Sie, dass Sie pro Woche im Internet verbringen? Privat und beruflich?

Welche Arten von Websites besuchen Sie, wenn Sie Internet surfen? Zum Schluss würde ich noch gerne wissen, was Ihre Lieblingsseiten sind?

Aufgaben - ca. 20 Minuten

Nun können wir mit dem Usability-Test und somit den Aufgaben beginnen. Ich werde die Aufgabe laut vorlesen, und Ihnen anschließend die ausgedruckte Version geben. Für bessere Ergebnisse würde ich Sie bitten, die Suche nicht zu verwenden, da ich so mehr über die Benutzung des Systems lerne. Weiters möchte ich an dieser Stelle erneut betonen, dass es sehr wertvoll für mich ist wenn Sie alles Gedachte laut aussprechen. Wenn es keine Fragen mehr gibt, können wir beginnen.

Abbildung 5.3: Thinking-Aloud Protokoll

Test fand zu Hause bei den ProbandInnen statt, gearbeitet wurde am Notebook der Beobachterin.

Test 1

Datum: 12.07.2014

Dauer der Testdurchführung: 17 Minuten

	Testperson 1	Testperson 2
Alter	36	35
Computererfahrung	sehr viel	sehr viel
Internetgewohnheiten	10 h/Woche im Internet; besuchte Seiten: Nachrichten, Gala, Facebook	20 h/Woche im Internet; besuchte Seiten: Nachrichten, Youtube, berufliche Seiten

Ergebnis: Alle Aufgaben konnten gelöst werden. Die TeilnehmerInnen mussten nirgends lange überlegen, um eine Aufgabe zu erledigen und waren rasch damit fertig.

Probleme:

- Dateisystem: Es war nicht unmittelbar klar, dass ein Doppelklick erforderlich ist um in einen Ordner zu navigieren.
- Dateisystem: ProbandInnen versuchten durch Klicken auf das Ordner-Icon in den Unterordner zu kommen, was nicht implementiert wurde.

Test 2

Datum: 01.08.2014

Dauer der Testdurchführung: 22 Minuten

	Testperson 3	Testperson 4
Alter	26	27
Computererfahrung	sehr viel	sehr viel
Internetgewohnheiten	20 h/Woche im Internet; besuchte Seiten: Facebook, Email, Wetter, Reisen, Youtube	25 h/Woche im Internet; besuchte Seiten: Facebook, Nachrichten, Youtube, beruflich

Ergebnis: Es konnten alle gestellten Aufgaben gelöst werden. Die Partizipantinnen diskutieren gemeinsam, wenn etwas für sie nicht logisch war um auf den richtigen Lö-

sungsweg zu kommen.

Probleme:

- Kontakte: Die TeilnehmerInnen wussten nicht auf Anhieb, wo der Name eingetragen werden muss.
- Wörterbuch: Es war nicht klar, dass vor dem Editieren das Bearbeiten-Icon gedrückt werden muss. Durch gemeinsames Nachdenken wurde die Aufgabe aber doch gelöst.

Test 3

Datum: 12.07.2014

Dauer der Testdurchführung: 25 Minuten

	Testperson 5	Testperson 6
Alter	59	62
Computererfahrung	wenig	wenig
Internetgewohnheiten	7 h/Woche im Internet; besuchte Seiten: Wetter, Nachrichten, Youtube, Facebook	5 h/Woche im Internet; besuchte Seiten: Nachrichten, Wetter

Ergebnis: Die meisten Aufgaben konnten gelöst werden, jedoch mussten die ProbandInnen teilweise einige Sekunden überlegen um eine Aufgabe zu erledigen. Allerdings konnten Szenarios im Laufe des Tests schneller gelöst werden, da die Benutzung bereits erlernt wurde.

Probleme:

- Pinnwand: Der *Add-Button* wurde nicht sofort gefunden.
- Dateisystem: Es war nicht klar, dass ein Doppelklick erforderlich ist, um in den Unterordner zu navigieren.
- Dateisystem: Eine Datei hinzufügen war erst beim zweiten Versuch möglich, da die TeilnehmerInnen die Icons nicht erkannt haben.
- Lesezeichen: Es war für die Personen nicht unmittelbar klar, ob beim Klick auf das Plus-Icon ein Ordner oder ein Lesezeichen erstellt wird.

- Aufgaben: Wiederum waren die Icons nicht klar, um neue Elemente zu erstellen.
- Adressbuch: Die TeilnehmerInnen mussten einige Sekunden überlegen, wo der Name des Kontakts eingetragen werden soll.
- Wörterbuch: Das Editieren eines Eintrags stellte sich als Herausforderung dar, weil der Editor deaktiviert ist wenn das *Edit-Icon* nicht betätigt wird.

Test 4

Datum: 02.08.2014

Dauer der Testdurchführung: 20 Minuten

	Testperson 7	Testperson 8
Alter	29	31
Computererfahrung	viel	viel
Internetgewohnheiten	10 h/Woche im Internet; besuchte Seiten: Facebook, Reisen, Preisjäger, Nachrichten	20 h/Woche im Internet; besuchte Seiten: Nachrichten, Facebook, Wikipedia, berufliche Seiten

Ergebnis: Es konnten fast alle Aufgaben gelöst werden. TeilnehmerInnen konnten manche Elemente nicht unmittelbar identifizieren, waren aber selbstständig in der Lage durch Nachdenken alle Szenarios zu lösen.

Probleme:

- Die linke Navigation wurde zu Beginn mit einer Toolbar verwechselt. Nach genauerem Ansehen und der textuellen Hilfe wurde die tatsächliche Funktion bekannt.
- Kontakte: Es war nicht unmittelbar klar, wo der Name hingeschrieben werden muss.
- Wörterbuch: Das Bearbeiten eines Eintrags konnte nicht umgesetzt werden, da nicht herausgefunden wurde, dass zuerst das Bearbeiten-Icon angeklickt werden muss.

Test 5

Datum: 02.08.2014

Dauer der Testdurchführung: 26 Minuten

	Testperson 9	Testperson 10
Alter	62	65
Computererfahrung	wenig	wenig
hline Internetgewohnheiten	5 h/Woche im Internet; besuchte Seiten: Wetter, Nachrichten, Wikipedia	3 h/Woche im Internet; besuchte Seiten: Nachrichten, Sport

Ergebnis: Die TeilnehmerInnen konnten die meisten Aufgaben ausführen, hatten aber ähnlich zum Team von Test 3 anfängliche Schwierigkeiten, die jedoch im Verlaufe des Tests gelöst werden konnten.

Probleme:

- Die Icons in den einzelnen Modulen waren nicht bekannt. Im Laufe der Testdurchführung waren sie aufgrund der Platzierung der Icons in der Lage, deren Zweck herauszufinden.
- Adressbuch: Die Personen wussten nicht, wo der Name eines Kontaktes hinzugefügt werden muss.
- Wörterbuch: Das Editieren eines Eintrags war eine Schwierigkeit, da der deaktivierte Editor Verwirrung stiftete.

5.3 Zusammenfassung der Ergebnisse

Die beiden Testmethoden waren sehr aufschlussreich. In dieser Sektion werden die Ergebnisse zusammengefasst und Lösungen dargeboten.

Die Evaluierung anhand der Usability Heuristiken zeigt auf, dass folgende Verbesserungen notwendig sind:

- Die Funktionen "Rückgängig machen" und "Wiederherstellen" des Browsers sind weit verbreitet und sollten implementiert werden. Wird einer dieser Buttons betätigt, so soll im Prototyp entweder vor oder zurück gesprungen werden.

- Kontext-Menü bzw. Shortcuts: Um erfahrenen BenutzerInnen die Möglichkeit zu geben, Schritte schneller zu absolvieren, sollten ein Kontext-Menü bzw. Shortcuts verwendet werden. Diese Funktionen stören weniger erprobte UserInnen nicht und sind eine große Hilfe für jene, die sich mit dem Programm besser auskennen.
- Dokumentation: Diese Verbesserung ist optional, da sie beim Personal Knowledge Desk eher nicht benötigt wird. Trotzdem würde es nicht schaden, eine solche Hilfestellung zu verfassen.

Durch die Thinking-Aloud Usability Testing Methode wurden ebenfalls Schwachstellen gefunden, die folgendermaßen behoben werden können:

Unverständliche Icons

Besonders Testpersonen mit wenig Computererfahrung hatten Schwierigkeiten, die *Create-, Edit- und Delete-Icons* zu deuten, da sie die Abbildungen nicht erkannten und somit die Funktionalität dahinter nicht klar war.

Aus diesem Grund sollten anstelle von Icons Buttons mit einem aussagekräftigen Text verwendet werden, um diese Hürde zu überwinden.

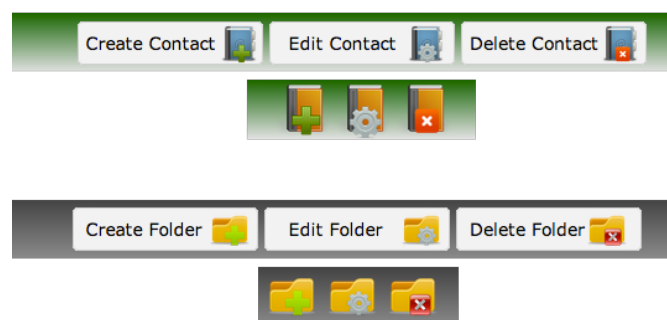


Abbildung 5.4: Buttons mit Text

Probleme beim Dateisystem

Beim Dateisystem ist es nicht möglich, die ganze Reihe inklusive Icon anzuklicken, um in einen Unterordner zu navigieren. Weiters war den ProbandInnen nicht klar, dass ein Doppelklick erforderlich ist, um einen Ordner zu betreten.

Diese Probleme können so umgangen werden, dass nur ein einfacher Klick notwendig ist, um in einen Folder zu navigieren. Weiters soll die ganze Reihe inklusive Icon

angeklickt werden können. Zusätzlich wäre eine Hervorhebung durch Unterstreichen förderlich, um den aktuellen Ordner zu markieren.

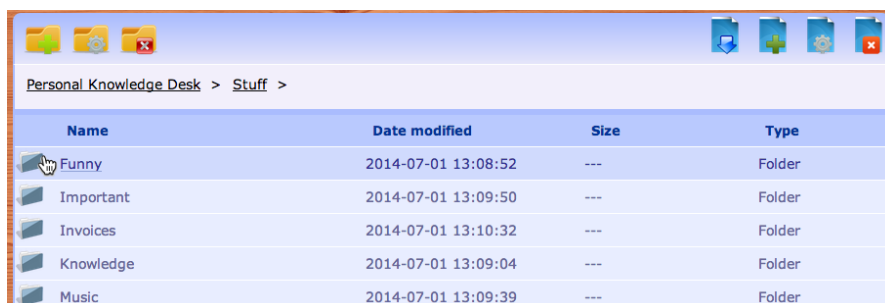


Abbildung 5.5: Optimiertes Dateisystem

Probleme beim Telefonbuch

Einigen TeilnehmerInnen des Usability Tests war nicht klar, wo der Name des Kontakts eingetragen werden soll. Daher soll "Insert name" in das leere Inputfeld geschrieben werden, um dies zu verdeutlichen.

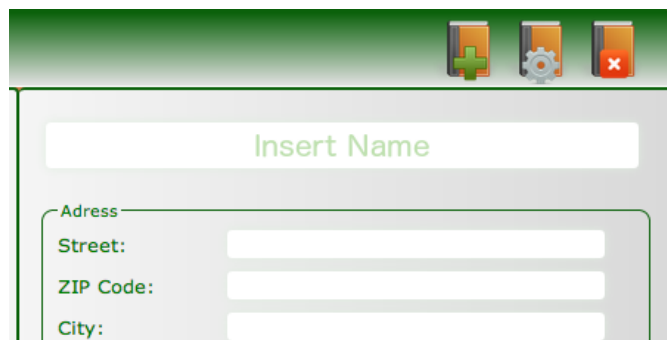


Abbildung 5.6: Optimiertes Telefonbuch

Probleme beim Wörterbuch

Das Wörterbuch rief die meisten Probleme hervor, da einigen Testpersonen nicht bewusst war, dass zuerst das *Edit-Icon* geklickt werden muss, damit der Editor aktiviert wird und somit ein Eintrag bearbeitet werden kann. Bisher wurde die Toolbar des Editors deaktiviert angezeigt, wenn ein Klick in das entsprechende Feld erfolgt. Besser wäre es jedoch, den Editor gar nicht anzuzeigen bevor das *Edit-Icon* angeklickt wird.

Außerdem würde ein Tooltip helfen, der beschreibt dass das Icon gedrückt werden muss um das Bearbeiten zu aktivieren.

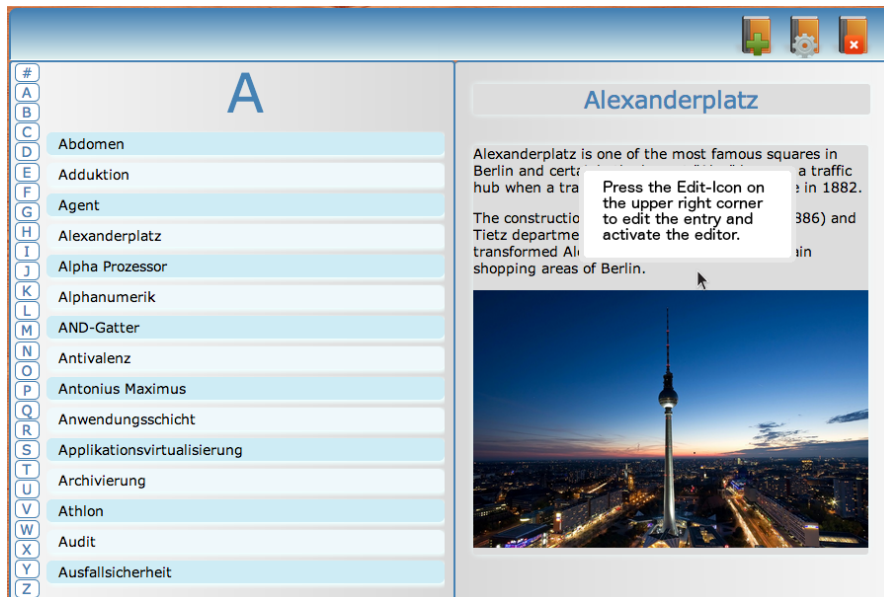


Abbildung 5.7: Optimiertes Wörterbuch

Probleme bei der Startseite

Obwohl alle PartizipantInnen die Bilder richtig zuordnen konnten, waren sich nicht alle sicher ob diese auch anklickbar sind. Es könnte helfen, einen Link zusätzlich anzuzeigen wenn die Maus eines der Elemente berührt.



Abbildung 5.8: Optimierte Elemente der Startseite

Zusammenfassung und Ausblick

Damit alle persönlichen Daten zentral an einer Stelle abgelegt werden können, wurde in dieser Arbeit ein System namens Personal Knowledge Desk entwickelt. Es wurde als Webapplikation umgesetzt, um der Plattformabhängigkeit entgegen zu wirken und keinen Installationsaufwand zu benötigen. Die größte Schwierigkeit war es dabei, ein schönes und ästhetisches User Interface zu schaffen, was gleichzeitig den Hauptfokus der Arbeit darstellt. Diese Fragestellung wurde bewusst gewählt, da viele bekannte bzw. bereits vorhandene Systeme aufgrund der Komplexität allgemein oder einer schlecht gestalteten Benutzeroberfläche schwierig zu benutzen sind.

Zuerst wurde eine Recherche in der Fachliteratur durchgeführt, um passende open-source Technologien für die Umsetzung zu finden und die Forschungsfrage "Welche Technologien sind für die Umsetzung gut geeignet?" in einem ersten Schritt zu beantworten. Die Wahl für den Webserver fiel dabei auf Apache, da dieser über XAMPP das lokale Testen des Codes ermöglicht, was für die Erstellung des Prototypen eine große Erleichterung darstellt. HTML5 und CSS3 wurden für die Struktur und das Styling des Inhaltes verwendet. Um den Content dynamisch zu verändern wurde auf Javascript bzw. jQuery und AJAX zurück gegriffen. Damit die Daten gespeichert werden können, kam MySQL zum Einsatz. Der Zugriff und die Speicherung erfolgte durch PHP. Zusätzlich wurden einige Frameworks wie zum Beispiel jQuery UI und Kendo UI eingesetzt, damit das User Interface benutzerfreundlicher gestaltet werden kann. Die oben beschriebenen Technologien waren ausreichend für die Erstellung des Systems. Im Nachhinein betrachtet wäre die Verwendung von CSS-Frameworks förderlich gewesen, um alle Inhalte konsistent zu erstellen. Besonders Bootstrap hätte sich geeignet, um die Oberflächengestaltungselemente anzufertigen und den Prototypen im Sinne des Responsive Webdesigns zu gestalten.

Die zweite Forschungsfrage lautet "Wie soll das User Interface in Form eines Schreibtischs am besten gestaltet sein, damit die Benutzerfreundlichkeit am höchsten ist?" Diese Frage wurde wiederum teils durch eine Literaturrecherche beantwortet. Für jedes graphische Element des Systems wurde herausgefunden, wie dieses am besten gestaltet werden soll hinsichtlich Kriterien wie Größe, Positionierung oder Gestaltung. Um das NutzerInnenverhalten im Internet besser zu verstehen, wurde auch dieses Thema erkundet. Weiters wurden allgemeine Patterns recherchiert, wie zum Beispiel welche Schriftgröße sich am besten für das Lesen eignet, wie die Suche aufgebaut und funktionieren soll oder wieviele Farben kombiniert werden können.

Diese Guidelines wurden verwendet, um anschließend einen Prototypen umzusetzen und das User Interface speziell dafür zu gestalten. Somit konnte erkundet werden ob die Technologien, welche durch theoretische Überlegungen ausgewählt wurden, auch in der Praxis geeignet waren. Da keine größeren Schwierigkeiten bei der Implementierung auftraten, wurden die Technologien als passend empfunden. Allerdings wurde erst im Laufe der Erstellung des Systems ersichtlich, dass verschiedene Frameworks wie zum Beispiel CKEditor und Full Calendar, sowie die asynchrone Übertragung mittels AJAX benötigt werden. Ein weiterer Grund für die Erstellung des Prototypen ist die Möglichkeit einer Evaluierung. Es wurde die Methode des Evolutionary Prototyping ausgewählt, da man hierbei bereits Software mit einer Grundfunktionalität erhält. Somit kann überprüft werden, ob das System auch wirklich benutzerfreundlich ist.

Nach diesen Schritten entstand ein System namens Personal Knowledge Desk. Da das Design ähnlich eines Schreibtischs ist wurden auch Features implementiert, welche auf einem solchen erwarten werden würden. Nach dem Einloggen befindet man sich am Schreibtisch, dessen reale Bilder statt einfachen Links als Navigationselemente funktionieren. Eine Pinnwand ermöglicht es, Post-its zu erstellen bzw. Fotos aufzuhängen. Im Dateisystem können Dateien hinzugefügt, heruntergeladen und organisiert werden. Der Zweck des Notizblocks ist das spontane Erstellen von unformatiertem Text. Der Kalender ermöglicht es, Termine zu erstellen und zu managen. Im Modul Wörterbuch können Begriffe erstellt und durch einen Editor formatiert werden. Das Telefonbuch erlaubt die Verwaltung von den persönlichen Kontakten. Um nichts zu vergessen, können in der Aufgabenliste Tasks kreiert und Kategorien zugewiesen werden. Die Bookmarks-Sektion realisiert die Organisation von Lesezeichen.

Nach der Fertigstellung des Prototyps erfolgte die Evaluation in zwei Phasen. Zuerst wurde die Applikation anhand der zehn gängigsten Usability Heuristiken von Jacob Nielsen getestet. Diese Heuristiken sind eher oberflächliche Faustregeln als speziel-

le Guidelines, trotzdem liefern sie wertvolle Erkenntnisse. Dabei ist heraus gekommen, dass beim Personal Knowledge Desk ein Context-Menü bzw. Shortcuts fehlen, um erfahrenen UserInnen das schnellere Arbeiten zu ermöglichen. Der zweite fehlende Aspekt ist die Tatsache, dass die "Rückgängig machen"- und "Wiederherstellen"-Funktion des Browsers nicht implementiert wurde.

Um eine tiefere Überprüfung der Benutzerfreundlichkeit zu erreichen, wurde außerdem Thinking Aloud Usability Testing durchgeführt. Hierbei wurden fünf Zweier-Teams spezielle Aufgaben zugeteilt, die sie im System lösen und ihre Gedanken dabei laut aussprechen müssen. Das größte Problem für die PartizipantInnen wurde dabei durch die Icons verursacht, da deren Zweck nicht unmittelbar klar war. Außerdem wurden einige kleinere Schwachstellen gefunden, wie zum Beispiel das Aktivieren des Editors beim Wörterbuch oder wo genau der Name beim Telefonbuch eingefügt werden muss. Erfreulich ist allerdings, dass die Benutzung innerhalb der Testdurchführung schneller und flüssiger wurde, da die TeilnehmerInnen bereits die Benutzung erlernt hatten.

Ausblick

Weiterführende Arbeiten würden zuerst die Eingliederung der aus den Testmethoden gewonnenen Erkenntnisse erfordern, um die vorhandenen Schwachstellen zu beheben. Obwohl der Personal Knowledge Desk über das Web zugänglich und somit plattformunabhängig ist, wäre eine Offline-Applikation von Vorteil, welche automatisch mit dem Webdesktop synchronisiert wird. Dabei sollten vor allem die Gestaltungsrichtlinien des Responsive Design berücksichtigt werden, um die Benutzung und eine einheitliche Gestaltung auf verschiedenen Geräten mit verschiedenen Größen zu ermöglichen.

Glossar

Aufgaben

Die Aufgaben sind ein Modul des Personal Knowledge Desks, wo verschiedene Tasks in entsprechenden Kategorien verwaltet und erstellt werden können.

Cloud Computing

Beim Cloud Computing werden IT-Dienste an externe HerstellerInnen ausgelagert. Somit müssen sich Unternehmen nicht mehr um die IT-Infrastruktur kümmern und Ressourcen können nach Bedarf freigegeben werden. Es werden drei verschiedene Services unterschieden: Infrastructure as a Service (Computerhardware wird freigegeben), Platform as a Service (Programmier- und Laufzeitumgebungen werden ausgelagert), Software as a Service (Anwendungsprogramme können in der Cloud genutzt werden).

Dateisystem

Das Dateisystem ist ein Modul des Personal Knowledge Desks. Hierbei ist es möglich, verschiedene Dateien hochzuladen, zu organisieren und auch herunterzuladen.

Kalender

Der Kalender ist ein Modul des Personal Knowledge Desks. Es können neue Einträge erstellt, bearbeitet und gelöscht werden.

Lesezeichen

Die Lesezeichen sind ein Modul des Personal Knowledge Desks. Hierbei können Bookmarks ähnlich wie bei modernen Browsern organisiert werden.

Modal message

Eine Modal message ist eine Textbox, welche Auskunft über gerade passierte Ereignisse gibt. Modal messages werden gerne eingesetzt, da sie UserInnen Feedback zum momentanen Stand im System geben.

Personal Knowledge Desk

Personal Knowledge Desk ist der Name des in der Diplomarbeit erstellten Prototypen.

Pinnwand

Die Pinnwand wird beim Personal Knowledge Desk eingesetzt, um Fotos und Sticky-

Notes abzulegen.

proprietäre Software

Proprietäre Software wird auch als unfreie Software bezeichnet und darf nicht beliebig durch Dritte adaptiert werden. Weiters wird der Quelltext nicht öffentlich gemacht.

Telefonbuch

Das Telefonbuch ist ein Modul des Personal Knowledge Desk und ermöglicht das Anlegen, Editieren und Löschen von Kontakten.

Usability

Usability oder Benutzerfreundlichkeit bezeichnet eine Qualitätseigenschaft und sagt aus, wie einfach ein System von BenutzerInnen verwendet werden kann.

Wörterbuch

Das Wörterbuch ist ein Modul des Personal Knowledge Desks. Es können dabei verschiedene Begriffe abgelegt und erklärt werden. Um die Übersichtlichkeit zu steigern, können Erklärungen schön formatiert werden.

Literaturverzeichnis

- [1] Muhammad Aliif Bin Ahmad and Noorminshah A. Iahad. Websites usability instrument validation using think-aloud method. In *2013 International Conference of Information and Communication Technology (ICoICT)*, pages 208–212. IEEE, March 2013.
- [2] Gary Anthes. HTML5 leads a web revolution. *Communications of the ACM*, 55(7):16–17, July 2012.
- [3] Steven E. Boone and Sk Semwal. A 3D virtual desktop. *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia - MUM '05*, page 163, 2005.
- [4] GE Burnett and D Ditsikas. Personality as A-Criterion for Selecting Usability Testing Participants. In *ITI 4th International Conference on Information and Communications Technology 2006*, number December, pages 10–12, 2006.
- [5] D. Dayton. Audiences involved, imagined, and invoked: trends in user-centered interactive information design. In *IEEE International Professional Communication Conference, 2003. IPCC 2003. Proceedings*. IEEE, 2003.
- [6] Erik Frøkjær and Kasper Hornbæk. Metaphors of human thinking for usability inspection and design. *ACM Transactions on Computer-Human Interaction*, 14(4):1–33, January 2008.
- [7] Mariella Di Giacomo. MySQL : Lessons Learned on a digital library. *Software, IEEE*, 22(3):10–13, 2005.

- [8] Rich Halstead-Nussloch. Utilizing web server middleware for development environments and student projects. In *Proceedings of the 49th Annual Southeast Regional Conference*, pages 348–349, New York, New York, USA, 2011. ACM Press.
- [9] Johan Harjono, Gloria Ng, Ding Kong, and Jimmy Lo. Building Smarter Web Applications with HTML5. In *CASCON '10 Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research*, pages 402–403, 2010.
- [10] Marcel Heerink. Exploring the influence of age, gender, education and computer experience on robot acceptance by older adults. *Proceedings of the 6th international conference on Human-robot interaction - HRI '11*, page 147, 2011.
- [11] M Hills, P Klint, and J Vinju. An Empirical Study of PHP Feature Usage. In *SSTA 2013 Proceedings of the 2013 International Symposium on Software Testing and Analysis*, pages 325–335, 2013.
- [12] Gan Keng Hoon, Saravadee Sae Tan, and Bryan Gan. MICE3: An Information Desktop on the Web. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 1075 – 1076, 2006.
- [13] Mike Hughes. Talking Out Loud Is Not the Same as Thinking Aloud :: UXmatters. <http://www.uxmatters.com/mt/archives/2012/03/talking-out-loud-is-not-the-same-as-thinking-aloud.php>.
- [14] Caroline Jarrett. 7 Basic Best Practices for Buttons. <http://www.uxmatters.com/mt/archives/2012/05/7-basic-best-practices-for-buttons.php>.
- [15] Yanwen Ju. VirtHome: A Web-Like Mobile Personalized Virtual Desktop Computing Space. In *2008 International Symposium on Information Science and Engineering*, pages 192–196. IEEE, December 2008.
- [16] Tom Kenny. How to Design Buttons to Help Improve Usability. <http://inspectelement.com/tutorials/how-to-design-buttons-to-help-improve-usability/>.
- [17] Michael O. Leavitt and Ben Shneiderman. *Research-based web design & usability guidelines*. U.S. Dept. of Health and Human Services, 2006.
- [18] Horst Lichter, Heinz Zullighcwen, and Matthias Schneider-hufschmidt. Prototyping in Industrial - Bridging the Gap Between Software Projects Theory and

Practice -. In *ICSE '93 Proceedings of the 15th international conference on Software Engineering*, pages 221–229, 1993.

- [19] Jakob; Nielsen. 10 Heuristics for User Interface Design: Article by Jakob Nielsen. <http://www.nngroup.com/articles/ten-usability-heuristics/>.
- [20] Jakob Nielsen. Durability of Usability Guidelines. <http://www.nngroup.com/articles/durability-of-usability-guidelines/>.
- [21] Jakob Nielsen. F-Shaped Pattern For Reading Web Content. <http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>.
- [22] Jakob Nielsen. Thinking Aloud: The #1 Usability Tool. <http://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>.
- [23] Jakob Nielsen. Usability 101: Introduction to Usability. <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [24] Jakob; Nielsen. Usability ROI Declining, But Still Strong. <http://www.nngroup.com/articles/usability-roi-declining-but-still-strong/>.
- [25] Jakob Nielsen. Why You Only Need to Test with 5 Users. <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [26] Jakob; Nielsen and Hoa Loranger. *Prioritizing Web Usability.pdf*. New Riders, 1st edition, 2006.
- [27] Pamela Ravasio, Sissel Guttormsen Schaer, and Helmut Krueger. In Pursuit of Desktop Evolution : User Problems and Practices With Modern Desktop Systems. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 11(2):156–180, 2004.
- [28] S. Rush, S. Halko, B. Masker, E. Cuddihy, J. Halko, K. Mobrand, and J. Spyridakis. Internet-based research for the desktop and beyond: Building a foundation of excellence for information design on the web. In *IEEE International Professional Communication Conference IPCC 2009*, pages 1–5. IEEE, July 2009.
- [29] Eric Savitz. Gartner: Top 10 Strategic Technology Trends For 2013. <http://www.forbes.com/sites/ericsavitz/2012/10/23/gartner-top-10-strategic-technology-trends-for-2013/>.

- [30] Lydia Schneidewind and S Horold. How personas support requirements engineering. In *Usability and Accessibility Focused Requirements Engineering (UsARE), 2012 First International Workshop on*, pages 1–5, 2012.
- [31] Haifeng Shen, Zhonghua Yang, and Chengzheng Sun. Collaborative web computing: From desktops to webtops. In *IEEE Distributed Systems Online*, volume 8, pages 4–7. IEEE, 2007.
- [32] Ashok Sivaji, Azween Abdullah, and Alan Giffin Downe. Usability Testing Methodology: Effectiveness of Heuristic Evaluation in E-Government Website Development. In *2011 Fifth Asia Modelling Symposium*, pages 68–72. IEEE, May 2011.
- [33] DH Sova and Jakob Nielsen. How to Recruit Participants for Usability Studies. 2003.
- [34] StatCounter. Top 10 Desktop, Tablet & Console Screen Resolutions from June 2013 to June 2014 | StatCounter Global Stats. <http://gs.statcounter.com/#resolution-ww-yearly-2011-2014>.
- [35] K Steve. *Don't make me think*. New Riders Publishing Thousand Oaks, 2009.
- [36] Antero Taivalsaari, Tommi Mikkonen, Matti Anttonen, and Arto Salminen. The Death of Binary Software: End User Software Moves to the Web. In *2011 Ninth International Conference on Creating, Connecting and Collaborating through Computing*, number 978, pages 17–23. IEEE, January 2011.
- [37] M.J. van den Haak and M.D.T. de Jong. Analyzing the interaction between facilitator and participants in two variants of the think-aloud method. In *IPCC 2005 Professional Communication Conference*, pages 323–327. IEEE, 2005.
- [38] L van Velsen. Testing the usability of a personalized system: comparing the use of interviews, questionnaires and thinking-aloud. In *Professional Communication Conference, 2007. IPCC 2007. IEEE International*, pages 1 – 8, 2007.
- [39] Om Prakash Verma, Rahul Katarya, Vishal Bhargava, and Nikhil Maheshwari. Use of semantic web in enabling desktop based knowledge management. In *2011 3rd International Conference on Electronics Computer Technology*, pages 190–193. IEEE, April 2011.
- [40] R. VidyaBanu. Implementation of financial system using EyeOS in the cloud environment. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, volume 3, pages 656–660. IEEE, June 2011.

- [41] Umi Kalsom Yusof, Leng Khang Khaw, Hui Yang Ch'ng, and Boon Jiann Neow. Balancing between usability and aesthetics of Web design. In *2010 International Symposium on Information Technology*, volume 1, pages 1–6. IEEE, June 2010.