

1. Studienprojekt

Front- und Backend einer Webanwendung für das Requirements-Engineering

Studienjahrgang: 2013
Fachbereich: Duales Studium Wirtschaft - Technik
Studiengang: Informatik
Modul: IT3161 Studienprojekt I
Betreuer Hochschule: Prof. Dr. Dagmar Monett Diaz
Anzahl der Wörter: ?

Gemeinschaftsprojekt von:

Name, Vorname: Kujat, Sven-Erik

Name, Vorname: Hartmann, Marvin

Name, Vorname: Kujat, Sven-Erik
Matrikelnummer: xXxxXx
Ausbildungsbetrieb: xXx
Betreuer Unternehmen: xXx. xXx. xXx. xXxxXx xXxxXx

Vom Ausbildungsleiter zur Kenntnis genommen:

.....
(Datum/Unterschrift)

.....
(Datum/Unterschrift der/des Studierenden)

Name, Vorname: Hartmann, Marvin
Matrikelnummer: 653861
Ausbildungsbetrieb: MSA - The Safety Company
Betreuer Unternehmen: Dipl. Ing. Frank Scheidemann

Vom Ausbildungsleiter zur Kenntnis genommen:

.....
(Datum/Unterschrift)

.....
(Datum/Unterschrift der/des Studierenden)

Inhaltsverzeichnis

Begriffserklärung	II
1 Einleitung	1
2 Beschreibung der Aufgabenstellung	3
3 Motivation	5
4 Entwicklung	6
4.1 Konzept	7
4.2 Realisierung	12
4.3 Entwicklungsprozess	15
5 Webanwendung <i>Red:wire</i>	18
5.1 Oberfläche	19
5.2 Features	24
6 Ausblick und Fazit	25
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Literaturverzeichnis	A- 1
Anhang	A- 2

Begriffserklärung

- XAMPP

XAMPP ist eine Zusammenstellung von Software und Werkzeugen, die ein maximal einfaches, betriebssystemunabhängiges Aufsetzen eines Servers für Entwicklungszwecke erreichen soll. Dazu gehören unter anderem der Webserver **A**pache mit der Datenbank **M**ySQL und den Skriptsprachen **P**erl und **P**HP.

- MySQL

MySQL ist eines der am weitesten verbreiteten Open-Source-Datenbankmanagementsysteme. Es beinhaltet zahlreiche Funktionen beispielsweise zur Datenverwaltung, Speicheroptimierung, **Anfragengeschwindigkeitsverbesserung** und Syntaxüberprüfung.

- PHP

PHP ist eine Skriptsprache, welche vor allem auf Servern genutzt wird. Sie ermöglicht das Ausführen von **Operationen** bevor ein html-Dokument erzeugt wird. Dadurch ist es möglich Websites dynamischer zu Gestalten.

- CSS

Ein **C**ascading **S**tyle **S**heets beinhaltet eine Gestaltungssprache, die vor allem in Zusammenhang mit HTML und XML genutzt wird. Durch Einbindung von CSS-Dateien kann das Aussehen beispielsweise einer Website maßgeblich beeinflusst werden. Mit aktuellen CSS Versionen kann außerdem die Darstellung abhängig von den Geräteeigenschaften optimiert werden. Dies betrifft vor allem die Darstellung für Mobilgeräte.

- API

Ein **A**pplication **P**rogramming **I**nterface ist eine Programmierschnittstelle, die von einer Software zur Verfügung gestellt wird, um ihre Funktionen Nutzen zu können. Meistens werden diese fertigen Funktionen genutzt, um komplizierte Vorgänge mit wenigen Zeilen Code ersetzen zu können. **Außerdem können sie zusätzliche Features ermöglichen.**

- JQuery

JQuery ist die meist verwendete Javascript-Bibliothek. Es bietet Funktionen, welche die Manipulation von Website-Elementen vereinfacht und erweitert.

- AJAX

Das AJAX Konzept ermöglicht eine asynchrone Kommunikation zwischen Browser und Server, sodass eine Website verändert und Content nachgeladen werden kann, ohne die komplette Seite neu laden zu müssen.

1 Einleitung

Will ein Unternehmer ein neues Produkt entwickeln, so sollte er zunächst überlegen „Warum“ Kunden dieses brauchen. Anschließend sollte er überlegen „Wie“ es entwickelt werden soll und zuletzt muss er ganz genau beschreiben „Was“ er haben will.

Dies ist natürlich eine sehr kurz gefasste Beschreibung, aber so sollte ein Unternehmer im Optimalfall ein Produkt entwickeln, angelehnt an den *Golden Circle* des Requirements Engineering [3] von innen nach außen. Leider wird häufig mit dem „Was“ begonnen und es kann vorkommen, dass die Anforderungen an das Produkt übereilt und unprofessionell vertraglich festgehalten werden. Das sollte nicht geschehen, denn Anforderungen sind sehr wichtig. Wenn sie nicht professionell behandelt werden, können daraus viele Probleme resultieren. Im worst case entsteht ein anderes Produkt, als von den Stakeholdern gewünscht und das Projekt wird abgebrochen. Auch kann es vorkommen, dass das Produkt zum Zeitpunkt der Fertigstellung bereits veraltet ist. Dabei kann schon zu Beginn ein schwerwiegender Fehler bei der Anforderungsaufnahme genügen, um das gesamte Projekt scheitern zu lassen. Dieser Fehler kann seinen Ursprung allerdings an unterschiedlichen Stellen haben. Seien es unvollständige, falsch interpretierte oder auch einfach nur unrealistische Anforderungen.

Zur Verbesserung der Anforderungsqualität existieren folgende Grundsätze [2]:

Anforderungen müssen...

- ...vollständig sein
- ...korrekt sein
- ...prüfbar sein
- ...eindeutig sein
- ...verstehbar sein
- ...gültig und aktuell sein
- ...realisierbar sein
- ...notwendig sein
- ...verfolgbar sein
- ...bewertet sein

Eine weitere Möglichkeit um dem Entwicklungsprozess zu verbessern und Fehlern entgegen zu wirken ist die agile Softwareentwicklung. Bei dieser müssen nicht direkt am Anfang alle Anforderungen in einem Pflichtenheft festgehalten werden, sondern sind dynamisch änderbar in einem Backlog festgehalten. Je nach genutztem System kann beispielsweise einmal im Monat ein Review der bisherigen Arbeit geschehen und anschließend eine neue Liste von Anforderungen für die folgende Entwicklungsphase erstellt werden (Beispiel Scrum [5][9]). Ein solches System erhöht außerdem die Chance, dass mögliche Fehler frühzeitig erkannt und behoben werden können. Das ist wichtig, da die Kosten um einen Fehler zu beheben von der Analyse und Design Phase (ca. 400 €), über die Test- und Abnahmephase (ca. 4800 €), bis hin zur Veränderung einer bereits in Betrieb befindlichen Software (ca. 9000 €) rapide zunehmen. Letzteres kann sogar Imageschäden mit sich bringen [4].

Es ist also zu sehen, dass die Wahl eines Entwicklungsprozesses zusammen mit dem Umgang mit Anforderungen sich entscheidend auf das Projekt auswirken können. Um dies einfacher zu gestalten, soll nun eine Software entwickelt werden, welche sowohl bei der Aufnahme, als auch der Verwaltung von Anforderungen eine sehr gute Unterstützung darstellt.

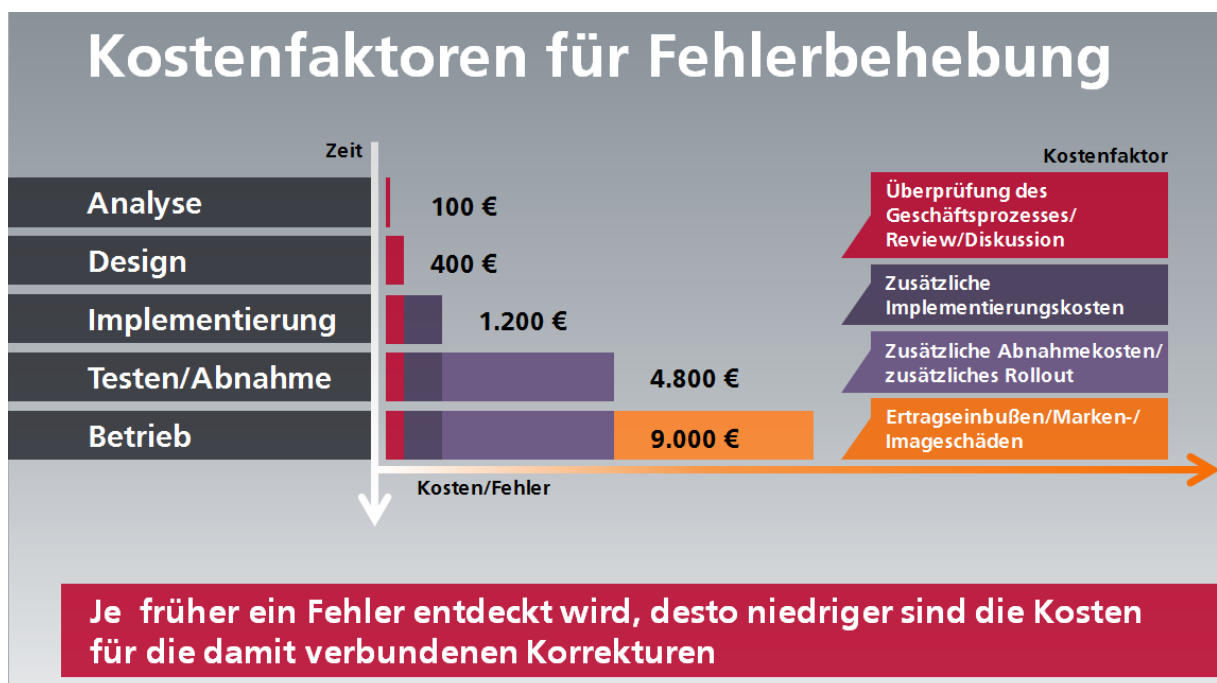


Abbildung 1.1: Kostenfaktoren für Fehlerbehebungen [4]

2 Beschreibung der Aufgabenstellung

Aufgabenstellung

Die Aufgabenstellung des hier beschriebenen Studienprojektes sieht vor, eine Online-Applikation bzw. Webapplikation zur Anforderungsaufnahme zu entwickeln. Anforderungen sollen mithilfe der „Rupp-Schablone“ von Chris Rupp formuliert werden. Die Idee hinter dem Online-Applikationskonzept ist die einfache Möglichkeit des zentralen Managements der Software und der guten Zugänglichkeit für mehrere Anwender auf die gleiche Datenbasis.

Als Hilfestellung zur Bewältigung dieses Projekts liegt eine Kopie des vorangegangenen Studienprojektes vor. Dieses behandelt die Aufnahme von Anforderungen auf Basis der „Rupp-Schablone“ in einer Offline-Applikation und steht zu Analyse des softwareseitigen Aufnahmeprozesses zur Verfügung.

Rupp-Schablone

Die „Rupp-Schablone“ dient der vereinfachten Aufnahme komplexer Anforderungen. Sie nutzt dazu nur eine Teilmenge der natürlichen Sprache. Genutzt wird die Schnittmenge, welche gleichermaßen Effizienz in der Bearbeitung, sowie Verständlichkeit für den Anwender darstellt.

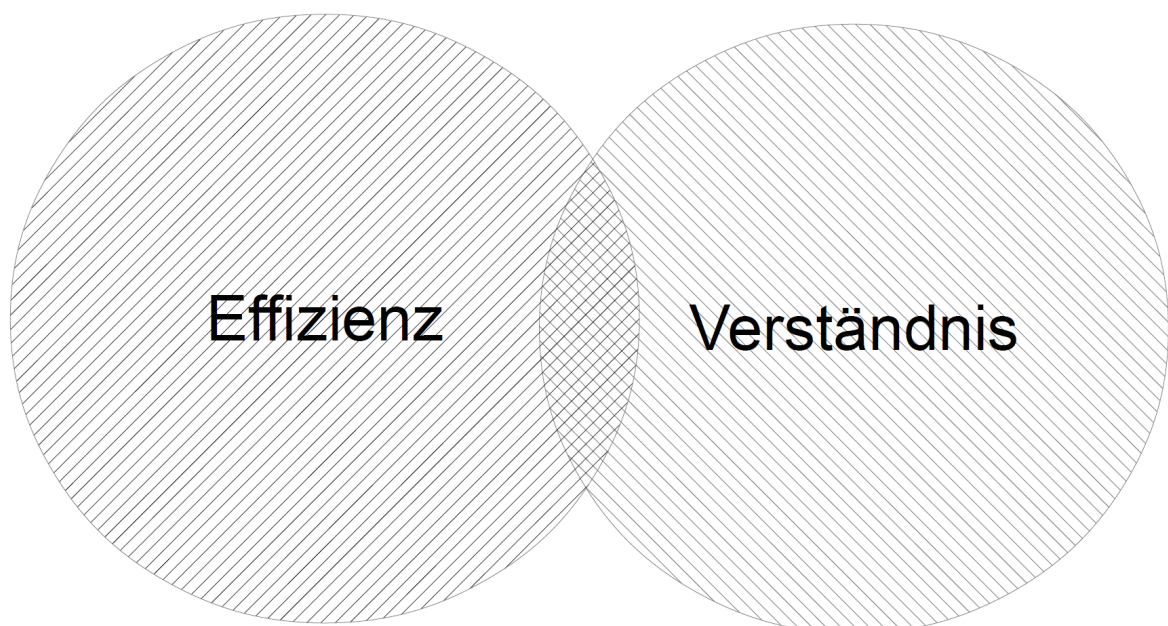


Abbildung 2.1: Schnittmenge

Der Aufbau der Schablone ist an den deutschen Satzbau angepasst. Dies kann mit einer eingeschränkten Form der Subjekt-Prädikat-Objekt Logik betitelt werden. Das erste Element des Aufbaus ist ein Subjekt. Dieses Subjekt stellt den Satzgegenstand dar. Folgend wird die Wichtigkeit der auszuführenden Tätigkeit auf Basis eines helfenden Prädikats festgelegt. Dieses wird von Rupp als rechtlich bindend bezeichnet (auf Vertragsbasis). Das dritte Element ist ein optionales Dativ-Objekt zur Bestimmung des Empfängers bzw. des Nutznießers der Tätigkeit. Mit dem vierten Element wird zwischen einer selbstständigen oder einer nutzerinitiierten Durchführung unterschieden. An fünfter Stelle steht das vom Subjekt zu behandelnde Objekt und abschließend das eigentlich auszuführende Prädikat. Das Prädikat beschreibt den Umgang bzw. die Beeinflussung des Objekts durch das Subjekt.

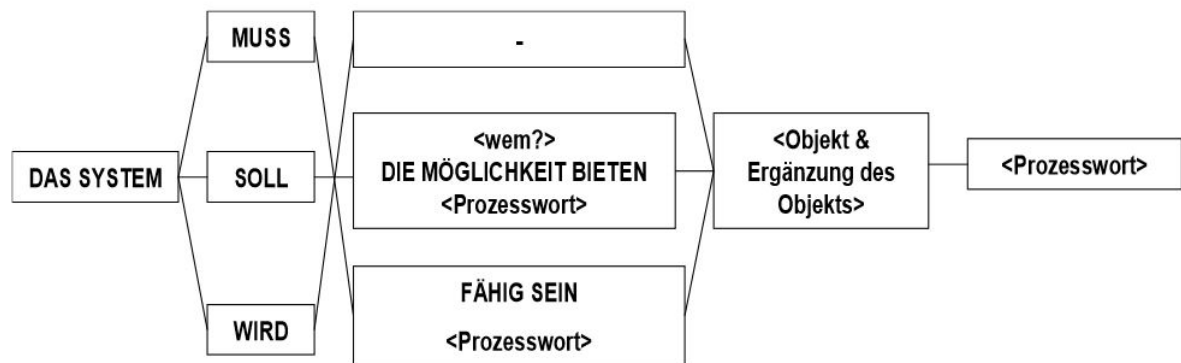


Abbildung 2.2: Schablone

Erweiterungen

Als wünschenswerte Entwicklungen für die Webapplikation seien im Gegensatz zur derzeitigen offline Lösung des vorigen Studienprojekts genannt: ein Organisationssystem zur Zusammenarbeit mehrerer Anwender an einem Anforderungsprojekt, ein Loginsystem zur Authentifizierung der einzelnen Anwender, eine Exportfunktion für die Anforderungen zur Weiterbearbeitung offline und einige Zusatzattribute zu jeder Anforderung (Priorität, Status, Identifikationsnummer, Abhängigkeiten).

3 Motivation

Die treibende Kraft zur Entwicklung einer Requirements-Engineering-Software hat ihren Ursprung in dem Leitgedanken, die Formulierung und Verwaltung von Anforderungen zu vereinfachen und zu verbessern. Wenn etwaige Verantwortliche unachtsam oder ungeübt mit Anforderungen agieren, wird dieser suboptimale Umgang leicht zur Gewohnheit. Dies können Fälle sein, wie etwa die Notation von Hand oder das Ablegen in einer einfachen Tabellensoftware. Hierbei wird weder die Formulierung kontrolliert noch eine gute Struktur gewährleistet. Infolgedessen entstehen schnell Redundanz und Konfusion. Dies bedeutet einen großen Verwaltungsaufwand, was die Kosten und Geschwindigkeit der eigentlichen Produktentwicklung maßgeblich beeinflusst. Um jenen Problemen entgegen zu wirken, wird eine Software entwickelt um ein einfaches Anforderungsmanagement zu ermöglichen. Durch die Eigenentwicklung bietet sich zusätzlich die Gelegenheit individuelle Wünsche und Features umzusetzen.

Infolge des Aufbaus als Webanwendung können Nutzer jederzeit über einen gewöhnlichen Browser auf ihre Anforderungen zugreifen. Aufgrund der steigenden Anzahl von Smartphone Nutzern, soll es auch ermöglicht werden, unterwegs über das Smartphone Zugriff zu erhalten. Dabei kann durch ein „Responsives Webdesign“ die Nutzererfahrung weiter verbessert werden.

Später kann die Software neben dem Einsatz in Projekten auch als Lernsoftware fungieren. Besonderen Vorteil bietet hierbei eine integrierte Formulierungsschablone. Jene soll dem Nutzer durch einen vorgegebenen Satzbau die Formulierung erleichtern.

Außerdem soll es ermöglicht werden, mit mehreren Personen an der selben Anforderungsliste zu arbeiten. Durch die resultierende Teamarbeit wird die Effizienz gesteigert und das Arbeitsklima verbessert. Nach Abgabe des Studienprojektes besteht zudem die Möglichkeit, die Software unter Open-Source-Lizenz zu veröffentlichen und somit unabhängige Weiterentwicklung zu ermöglichen.

4 Entwicklung

Die Entwicklung der Webapplikation wird in mehreren Schritten durchgeführt. Vorangehend ist eine ganzheitliche Planung und Strukturierung für den erfolgreichen Abschluss des Projekts unabdinglich. Vorausschauend muss die vorgegebene Zeitspanne zur Erfüllung der Aufgabe eingeteilt und unterteilt werden. Es muss ein Konzept und ein Leitfaden entstehen, auf welchem die Entwicklung basiert und durchgeführt wird. Weiterhin müssen Überlegungen zu möglichen Technologien und deren Vor- und Nachteile getätigt werden. Sobald diese Grundbausteine ausgearbeitet sind, ist es wichtig, eine funktionierende Infrastruktur zur Softwareentwicklung zu entwerfen. Innerhalb dieser muss auch die Lauffähigkeit der zu entwickelnden Applikation gewährleistet sein. Damit kann schon jetzt später möglicherweise auftretenden Fehlern vorgebeugt, bzw. diese beschränkt werden. Nach der darauffolgenden Erstellung der Aufgaben werden diese aufgeteilt, um paralleles Arbeiten effizient zu ermöglichen.

4.1 Konzept

Anders als bei einer offline Applikation verhalten sich die Daten bei einer Webanwendung wie folgt: Ein Benutzer schickt Daten von einem Rechner (Home-PC) an einen anderen (Server). Dieser schickt eine Antwort mit Daten zurück. Aber das Internet und die Internetseiten werden zunehmend komplexer. Vergleicht man ältere Versionen einer Website, mit deren Version von heute, werden viele Unterschiede ersichtlich. Als auffällige Änderungen seien hier beispielsweise das Layout, die Schrift und die Menüstruktur zu nennen. Dies wird deutlich im Vergleich der Abbildungen 4.1 und 4.2.

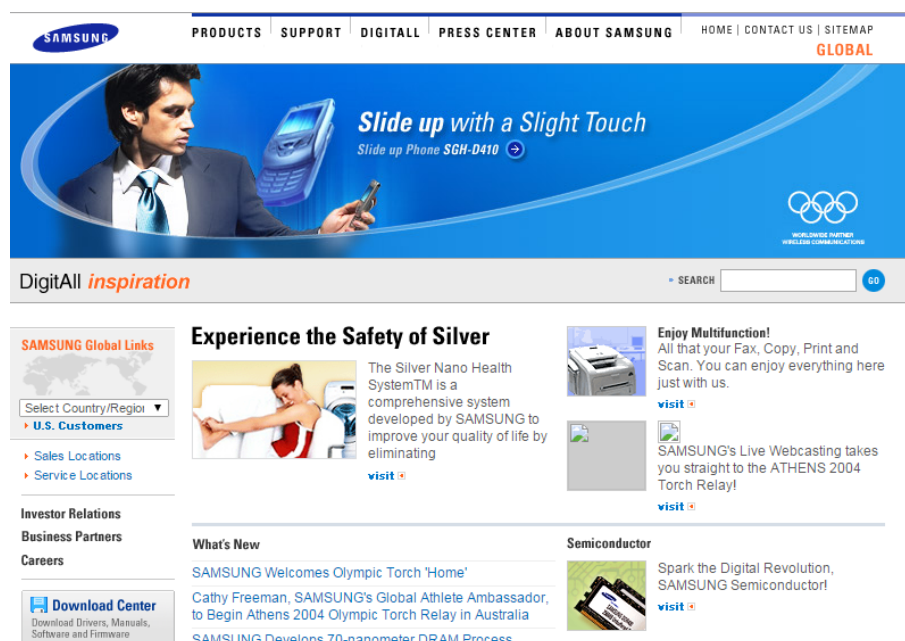


Abbildung 4.1: www.samsung.com Juni 2004 (Screenshot Wayback Machine)

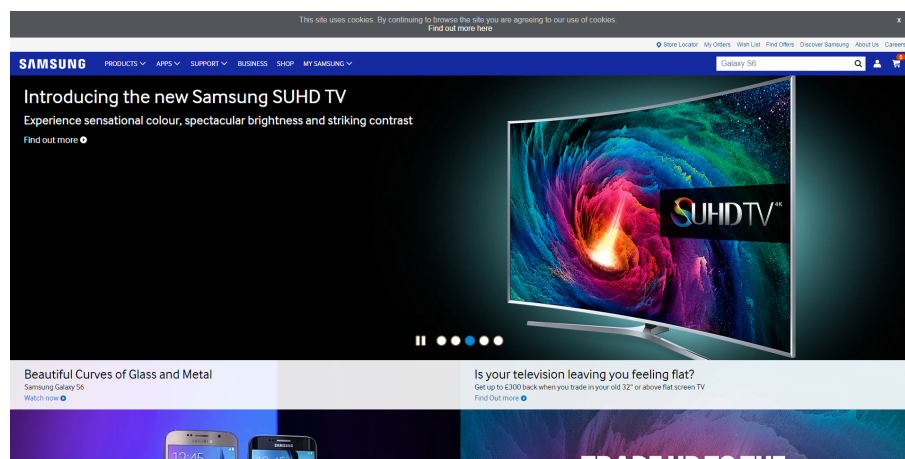


Abbildung 4.2: www.samsung.com Juni 2015 (Screenshot)

Im Verlauf der Entwicklung des Internets gab es immer wieder neue Herausforderungen. Neue Technologien und ein verändertes Nutzerverhalten, machen es unmöglich mit statischen Websites erfolgreich zu bleiben. Aktuell spielt die Verbreitung von Smartphones und steigende mobile Internetnutzung eine große Rolle. Für Suchanfragen bei Google, die von mobilen Geräten (Smartphones) durchgeführt werden, wurde bereits im Juni 2013 eine Änderung am Ranking-Algorithmus bekannt gegeben [8]. Am 26.02.2015 kündigte Google an, das Ranking für „Mobile-friendly“ Websites mit in die Gesamtbewertung dieser Suchanfragen einfließen zu lassen [6]. Diese Änderung wurde planmäßig am 21.04.2015 implementiert und veröffentlicht [7] und trägt dazu bei, dass viele Unternehmen ihre Internetseiten für Smartphones anpassen.

Durch diesen Schritt seitens Google wird einmal mehr verdeutlicht, dass es heutzutage notwendig ist auf einem hohen Technologischen Stand zu sein. Infolgedessen wurde beschlossen, bei der Entwicklung der Webanwendung in diesem Projekt verschiedene Technologien zu verbinden, um dem Benutzer ein positives Nutzungserlebnis zu bieten. Bootstrap, JQuery und MySQL sind als Beispiele der genutzten Tools und Framework zu nennen. Nach der Überprüfung unterschiedlicher Konzeptionen wurde schließlich folgendes Konzept festgehalten:

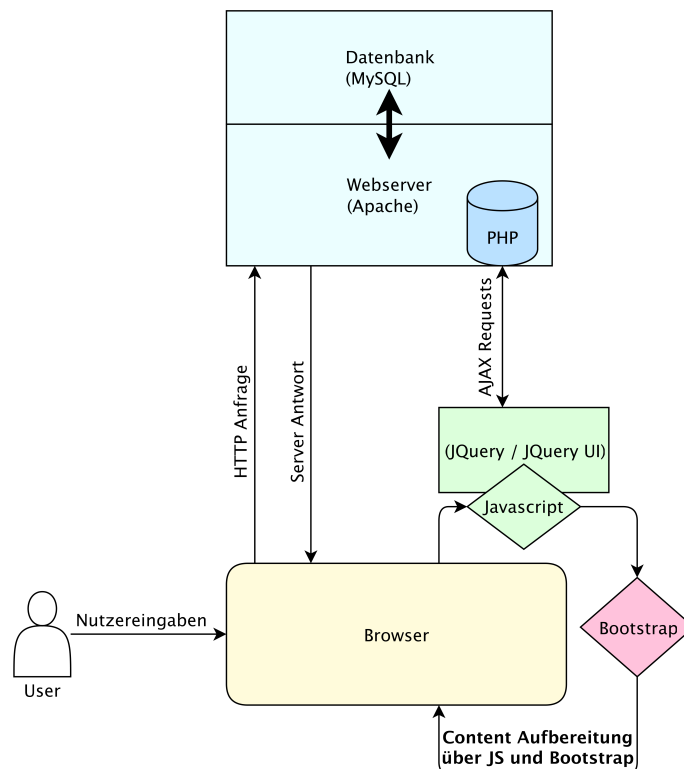


Abbildung 4.3: Webanwendung Konzept

Die Nutzereingaben werden vom Browser erfasst. Zunächst schickt dieser einen gewöhnlichen HTTP-Request an den Server. In der Antwort sind anschließend HTML, CSS und Scriptdaten enthalten. Diese benutzt der Browser um den Inhalt aufzubereiten und die Darstellung anzupassen. Außerdem kann der Funktionsumfang der Website mittels JQuery und Bootstrap weiter erweitert werden.

Dem Benutzer werden mehrere Funktionen geboten, ein Überblick gibt die Abbildung 4.4. Auf die Funktionsweise der einzelnen Use-Cases wird in **Punkt ??** näher eingegangen. Neben dem Registrieren und Einloggen, sollte der Benutzer sich natürlich am Ende seiner Arbeit ausloggen können. Nach der Anmeldung, sollte er sich auf einer Übersichtsseite (dem Dashboard) befinden, welche im eine Reihe von möglichen Aktionen bietet. Es soll ihm möglich sein Anforderungen zu verfassen und anzuzeigen, sowie diese Anzeige zur besseren Übersicht zu manipulieren. Entsprechende Aktionen bezüglich Anforderungen sind in der Abbildung 4.4 leicht grün hinterlegt. Jene Anforderungen sollen Teamgebunden und vom gesamten Entwicklerteam einsehbar sein, das heißt für jeden der Mitglied im entsprechenden Team ist. Mitglieder eines Teams sollten außerdem erweiterte Möglichkeiten haben das Team zu bearbeiten. Entsprechende geplante Aktionen bezüglich der Teams sind in der Abbildung 4.4 hellblau hinterlegt. Neben den funktionellen Aktionen, sollte ein Nutzer aber auch selbständig seine Accountinformationen wie Passwort und Email ändern können. Etwaige nutzerbezogene Aktionen sind in der Abbildung 4.4 hellgelb hinterlegt. Um das System zu Warten, sollte ein Administrator zumindest die Möglichkeit haben Nutzer zu löschen.

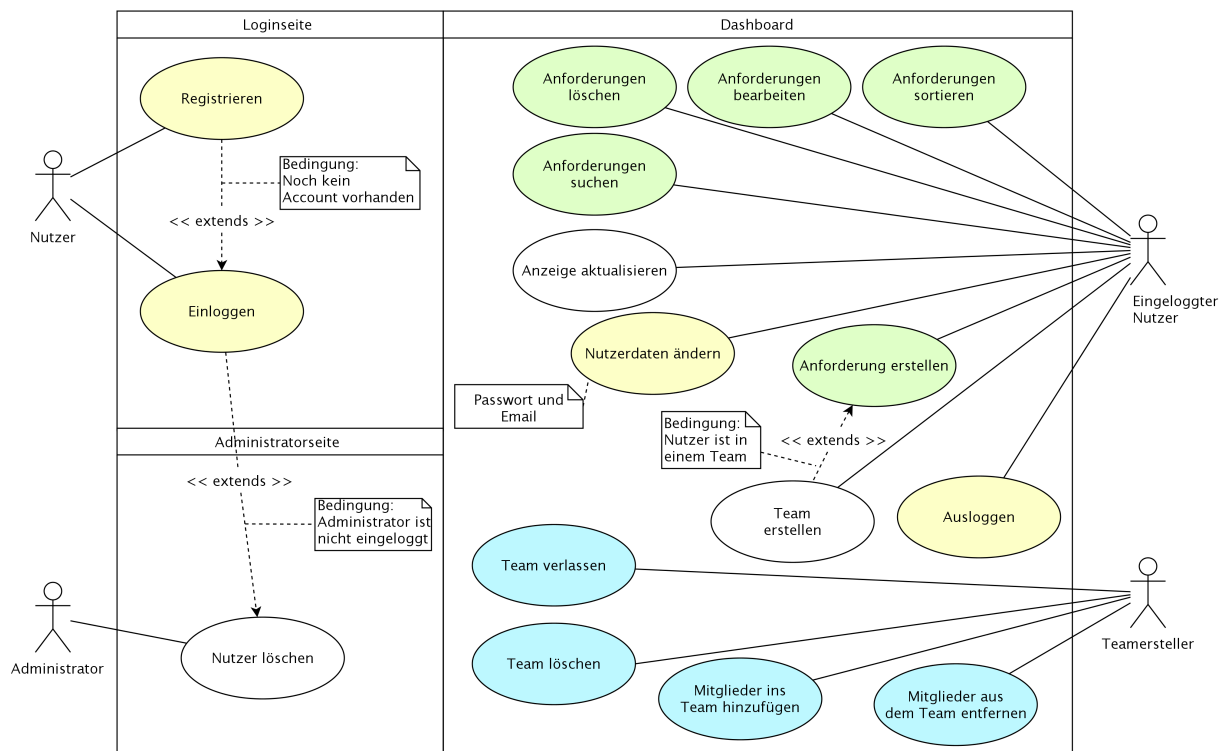


Abbildung 4.4: Use-Case Diagramm Nutzeraktionen

Folgende Meilensteine, sowie Zeitrahmen sind der Entwicklung angedacht: Zunächst ist das Ziel ein funktionierendes Anmeldesystem inklusive Datenbank zu entwickeln. Das ist notwendig, da die Software direkt auf Basis dieses Multi-User-Systems weiter entwickelt werden soll. Um dieses Grundgerüst zu schaffen und möglichst viele Fehlerquellen schon bei der Entwicklung zu finden und zu beheben, wurde zunächst ein Zeitrahmen von etwa 1,5 Monaten gesetzt. Die Entwicklung wurde parallel zur Theoriezeit des Dualen Studiums an der Hochschule gestartet. Ebenso ist dieser Schritt ein erster Meilenstein.

Nachdem das Benutzersystem auf seine ordnungsgemäße Funktion getestet wurde, richtet sich die Entwicklung auf die Anforderungen. Es muss gewährleistet werden, dass Anforderungen in die Datenbank geschrieben und ausgelesen werden können. Dazu ist vor allem ein Separator notwendig, damit die Satzglieder der Anforderungen später rekonstruiert werden können. Auch muss es ermöglicht werden, dass Nebeninformationen wie etwas die Ersteller ID oder ein Zeitstempel mit gespeichert werden können. Ist dieses Ziel erreicht muss ein geeignetes Ausgabeformat gewählt, und die korrekte Darstellung der Anforderungen überprüft werden. Zusätzlich muss man die Übersichtlichkeit der Anforderungen betrachten, um diese nicht zu vernachlässigen sind Sortierfunktionen notwendig. Funktionierte alles wie gewünscht, so sollte der zweite Meilenstein nach weiteren 2 Monaten erreicht sein.

Der dritte Meilenstein besteht hauptsächlich aus dem Teamsystem. Jenes soll gleichzeitiges Arbeiten an der selben Anforderungsliste ermöglichen. Um dies zu erreichen, muss es zunächst möglich sein Teams zu erstellen. Auch Verwalten von Teams, das heißt das Hinzufügen und Löschen von anderen Nutzern ist ein essentieller Bestandteil des Systems. Besonders bei diesem Teil des Systems muss kontrolliert werden, dass der Nutzer keine Fehler erzeugen kann. Vor allem betrifft dies das hinzufügen von Nutzern und die Namen von Teams. Um Komplikationen bei den Datenbankabfragen zu vermeiden sollte jeder Nutzer nur in einem Team, und die Teamnamen einzigartig sein. Neben dem Teamsystem liegen auf dem Weg zum dritten Meilenstein auch optische Anpassungen. Dazu gehören beispielsweise Tooltips, um somit eine angenehme Mensch-Maschine Interaktion zu schaffen. Zu diesem Zweck sollten auch Dialoge und Komplexität der Menüführung kontrolliert werden. Zur Überprüfung der Intuitivität empfiehlt es sich, die Software von Dritten, wie etwa Freunden oder Kollegen, testen zu lassen. Aufgrund von immer umfassenderem Testen, wird der Zeitaufwand hier deutlich mehr betragen als bei den vorherigen Meilensteinen. Ist alles zufriedenstellend, so kann der nächste Meilenstein als erreicht betrachtet werden. Allerdings werden hier weiterhin Anpassungen und eventuell Bugfixes von Nöten sein. Die verbleibende Zeit wird genutzt, um Fehler zu beheben und das Programm zu verbessern. Außerdem können weitere Ideen und Features, wie der News-Feed, einfließen.

4.2 Realisierung

Einleitung

Am Anfang der Realisierung steht die Planung sowie die Auswahl der Technologien, Programmiersprachen, Tools bzw. Tool-Chain. Diese sollten sorgsam gewählt werden, da sie die gesamte Entwicklungsdauer über genutzt werden. Durch die Analyse der benötigten Architektur und Infrastruktur lassen sich die verfügbaren Tools und Programmiersprachen eingrenzen, welche mit der Umgebung kompatibel sind.

Vision

Damit das Projekt erfolgreich abgeschlossen werden kann, ist es notwendig, sich zuerst eine Idee bzw. Vision des künftigen Produkts zu erschließen. Darunter fallen insbesondere die Funktionalitäten, das Design und die Bedienung. Zu diesen drei Kategorien werden alle Ideen dokumentiert. Durch Revision der einzelnen Vorschläge resultieren nun folgende Ergebnisse:

- Das Tool muss Anforderungen aus Eingabefeldern auslesen können
- Das Tool muss ein Authentifikationssystem bieten
- Das Tool muss ein Teamsystem bieten
- Das Tool muss ein Securitysystem bieten
- Das Tool muss Anforderungen exportieren können
- Das Tool muss das Bearbeiten vorhandener Anforderungen ermöglichen
- Das Design soll Aktualität widerspiegeln
- Das Design soll Simplizität darstellen
- Das Design soll Offenheit implizieren
- Die Bedienung soll intuitiv sein
- Die Bedienung soll einfach gehalten sein

Durch die exakte Umsetzung dieser Punkte sollte das Projekt erfolgreich abschließen. Durch Recherchen und Evaluationen bestimmter Techniken zum Erreichen positiver Nutzererfahrungen soll ein ansprechendes Design erstellt werden, welches den Nutzer durch visuelles Feedback bei der Arbeit unterstützt.

Architektur

Eine Webapplikation zeichnet sich dadurch aus, dass sie im Netzwerk arbeitet. Das bedeutet, dass sie für alle authentifizierten Anwender in diesem Netzwerk zugänglich sein muss. Aus Sicht des Systems ist der Anwender ein Client, welcher Datenpakete an den Server sendet und abfragt. Basierend auf dieser Tatsache wird abgeleitet, dass es hier um eine Client-Server Architektur handelt. Der Server ist deshalb ein Webserver, der die Dokumente für den Client bereitstellt und für Interaktionen mit dem Anwender bereit steht. Da Daten persistent gespeichert werden müssen, so dass über einen längeren Zeitraum mit ihnen gearbeitet werden kann, wird eine Datenbank benötigt. Diese beiden Komponenten (Webserver, Datenbank) sind für die Datenverarbeitung zuständig, während der Client (Browser) die verarbeiteten Daten empfängt und neue Daten / Anfragen an den Server sendet.

Sprachen

Die oben beschriebene vorausgesetzte Umgebung lässt sich am einfachsten mittels XAMPP (Windows, OSX), oder LAMP (Linux) realisieren. XAMPP steht für Apache, MySQL, PHP, Perl und LAMP für Linux, Apache, MySQL, PHP. In beiden dieser Pakete sind jeweils schon ein Webserver (Apache) und eine Datenbank (MySQL, relational) enthalten. Damit die Datenbankanbindung gewährleistet ist, bieten diese Pakete PHP-Unterstützung mit an. Nur so lässt sich die Datenbank ansprechen und es können Daten ausgelesen und abgespeichert werden. Das Frontend wird dementsprechend mittels der Struktursprache HTML geschrieben. HTML erlaubt das Einbetten von (u.A.) PHP Code, falls die PHP-Unterstützung des Webserver aktiviert ist. Die Formatierung der HTML-Struktur wird durch CSS realisiert. Zur intuitiveren Bedienung der Webapplikation wird zusätzlich das Javascript genutzt. Es erlaubt DOM-Manipulationen (Document Object Model) einfach durchzuführen.

Frameworks

Frameworks stellen schon vorgefertigte Funktionen einer bestimmten Sprache zur Verfügung. In einem größeren Softwareprojekt lässt sich so ohne Mehraufwand, im Vergleich zur puren Programmiersprache, ein funktionales Code-Grundgerüst bauen. Das Design der Website wird mittels CSS erstellt und setzt auf ein schon strukturiertes HTML-Dokument auf. Derzeit werden allerdings immer mehr Websites über mobile Endgeräte aufgerufen, welche kleinere Displays und/oder Auflösungen als aktuelle Home Computer besitzen. Um das Design trotzdem für alle Geräte zu optimieren, eignet sich CSS nativ nur bedingt. Deshalb wird das CSS-Framework „SASS“ genutzt, welches eine ähnliche Syntax

besitzt, jedoch mit weniger Code auskommt. Außerdem wird eine Klassenbibliothek für CSS-Klassen genutzt. Diese heißt „Bootstrap“ und ist in der Webentwicklung weit verbreitet. Bootstrap bietet viele responsive Klasse und Layouts, die z.B. alte nicht-responsive Tabellen-Layouts sehr einfach ersetzen können. Viele Klassen besitzen zusätzliche Effekte bei Hover (Mauszeiger über Element), oder Klickanimationen. Für einige bestimmte Effekte wird weiterhin das von jQuery stammende „jQuery UI“ genutzt, insbesondere für Klickanimationen. Javascript erlaubt DOM-Manipulationen. Das ermöglicht eine intuitivere Bedienung der grafischen Weboberfläche. Um diese Manipulationen, wie z.B. Fades und Slides einfacher implementieren zu können, bietet das Framework jQuery eine vorgefertigte Funktionsliste, welche nur aufgerufen werden muss.

Tools

Um eine grundlegende Planung durchzuführen, wird das Projektmanagementtool „Trello“ genutzt. Dieses ermöglicht die Erstellung von Story Cards, teambasiertes Arbeiten an sogenannten „Boards“, Zuweisungen von Aufgaben an Teammitglieder und Sortieren aller Aufgaben. Das Tool ist webbasiert, kostenlos und kann unter www.trello.com erreicht werden.

Zur Installation der Softwareumgebung wird das XAMPP-Paket unter Windows installiert. Dieses bietet einen Webserver, eine relationale Datenbank und PHP-Anbindung für den Webserver.

Das eigentliche Programmieren erfolgt in „Notepad++“. Ein Texteditor mit Syntax-Highlighting für viele verschiedene Sprachen. Genau diese Funktionalität macht es zu einem einfach nutzbaren und konfigurierbaren Entwicklungstool. Alle in dem Projekt genutzten Sprachen werden vom Highlighting unterstützt. Zum automatisierten Kompilieren von SASS „.scss“ Dateien in „.css“ Dateien wird das Command-Line Tool „Compass“ genutzt. Dieses besitzt eine eigene Ruby Command-Line, mit welcher die Kompilierung getriggert wird. Es kann außerdem einen Verzeichnispfad beobachten und kompiliert alle dort befindlichen Dateien, sobald eine Änderung der Checksumme eben dieser wahrgenommen wird.

Für Debug-Zwecke dient die Konsole im Firefox-Browser. Über diese können beispielsweise Variablenwerte ausgelesen, Funktionsaufrufe durchgeführt und Elemente untersucht werden.

4.3 Entwicklungsprozess

Datenbank

Um die Daten der Applikation zu speichern, wird eine Datenbank aufgesetzt. Diese enthält drei Tabellen. Eine Tabelle für die Nutzerinformationen, eine für die Anforderungsinformationen und eine für die Teaminformationen. Der Zusammenhang dieser besteht mittels Fremdschlüsselvergabe an jeweilige Attribute. Zur eindeutigen Identifikation eines Datensatzes werden Identifikationsnummern vergeben.

Grundstruktur

Nach allen Vorbereitungen werden die manifestierten Ideen umgesetzt. Als Hauptelement dient ein HTML - Template. Das Template ist eigens für diese Applikation erstellt worden und umfasst die Seitenaufteilung in mehrere Bereiche (Layout). Alle Bereiche sind gekapselte Objekte und nicht voneinander abhängig, um eine optimale Modularität zu gewährleisten. Das Template besteht aus Kopf- sowie Fußbereich, aus Inhaltsbereich und aus links positionierter Menüleiste, ist zunächst rein statisch und dient zur Demonstration der Optik.

Für das Authentifizierungssystem wird ein weiteres HTML - Template erstellt, welches als Login-Seite genutzt wird. Auch dieses ist statisch und wird nach und nach im Funktionsumfang erweitert.

Für die Administrationsseite wird eine abgewandelte Form des Applikations-Haupttemplates verwendet, welches nur die notwendigen Funktionen zur Administration enthält, jedoch das gleiche Layout besitzt.

Mittels CSS werden die Seiten formatiert. Die Admin- und Hauptseite greifen auf die selbe CSS-Datei zu, um Einheitlichkeit zu behalten. Für die Loginseite wird eine eigene CSS-Datei erstellt.

Einbau von Funktionalitäten

Loginseite

In die nun fertigen HTML-Seiten werden die Funktionalitäten einzeln eingebaut. Zuerst werden die Authentifizierungsfunktionen implementiert. Dazu zählen:

- Einlesen der Nutzerdaten (Name, Passwort)
- Abgleichen der Nutzerdaten mit den in der Datenbank hinterlegten
- Weiterleitung an die Hauptseite
- Fehlermeldung bei Falscheingabe

Zum Testen des korrekten Logins wird ein Datensatz manuell in die Datenbank eingetragen. Mit diesen eingetragenen Daten wird sich eingeloggt. Um jedem die Möglichkeit zu bieten, das Tool zu nutzen, wird außerdem eine Registrierungsmöglichkeit geboten. Diese erweitert den Funktionsumfang der Seite um:

- Einlesen neuer Daten
- Speicherung der Daten in der Datenbank

Diese Funktionen werden mit dem Registrieren eines neuen Nutzers und dem anschließenden Login mit dessen Daten getestet.

Hauptseite

Die Applikations-Hauptseite muss einen sehr großen Umfang an Funktionalitäten bieten. Grundlegende Funktionen sind hier:

- Aufnahme von Anforderungen
- Darstellung von Anforderungen
- Bearbeiten / Löschen von Anforderungen
- Erstellen von Teams
- Beitreten in Teams / Austreten aus Teams
- Ändern der Profildaten
- Logout aus der Applikation

Zum Testen dieser Funktionen werden Anforderungen erstellt und ausgelesen. Sobald dieser Test erfolgreich ist, wird das Bearbeiten sowie Löschen der Funktionen getestet. Es wird ein Team erstellt, Nutzer diesem Team zugewiesen und das Team gelöscht. Weiterhin werden die Profildaten neu eingegeben und es wird sich aus der Applikation ausgeloggt.

Adminseite

Die Adminseite ist nur dem Administrator zugänglich und bietet einen eingeschränkten Funktionsumfang. Es lassen sich lediglich Nutzer und/oder Teams löschen. Die Nutzerdaten des Admins können außerdem geändert werden.

Zu Testen ist dies, indem ein bestehender Nutzer gelöscht, sowie dessen Team an einen anderen Nutzer übertragen wird.

5 Webanwendung *Red:wire*

- Namenswahl, weil..? (Roter Faden)
- Browser Voraussetzungen (Browser/Javascript/Cookies)
- (- Server Voraussetzungen?)
- Tests

5.1 Oberfläche

Die Loginseite der Webanwendung besteht nur aus den unbedingt benötigten Komponenten (Abbildung 5.1). Unter dem Namen „Red:wire“ befinden sich zwei Inputfelder für den Benutzernamen und das Passwort. Darunter zwei Buttons, einer zum Einloggen und der andere zur Registrierung, sollte man einen neuen Account anlegen wollen. Durch diese einfache Aufteilung funktioniert der Login schnell und intuitiv. Nach der Anmeldung gelangt man direkt auf die Hauptseite.

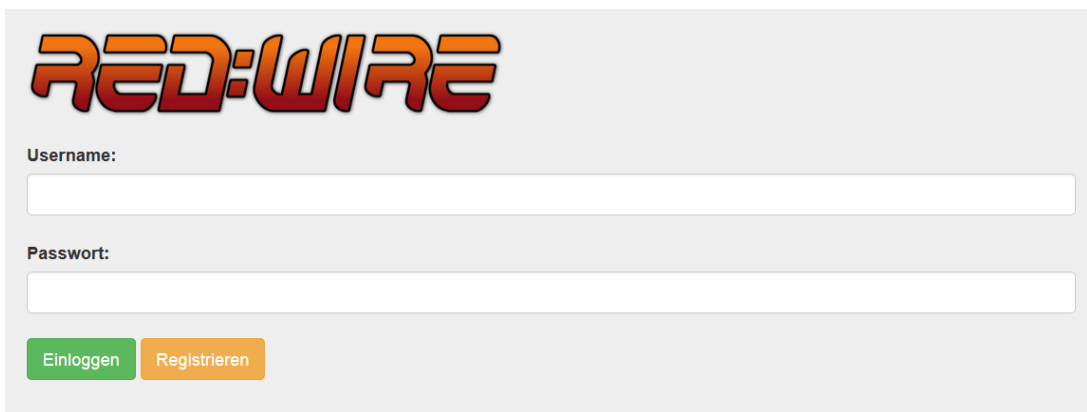
The screenshot shows a login interface for 'Red:wire'. At the top is the 'RED:WIRE' logo in a stylized, orange-to-red gradient font. Below the logo, the text 'Username:' is followed by a white input field. Below that, the text 'Passwort:' is followed by another white input field. At the bottom, there are two buttons: a green one labeled 'Einloggen' and an orange one labeled 'Registrieren'.

Abbildung 5.1: Screenshot der Loginseite

Die Struktur der Hauptseite setzt sich aus 4 Bereichen zusammen (Abbildung 5.2). Dieses Layout bietet ausreichend Platz und ist gleichzeitig noch sehr Übersichtlich. Oben im Header befindet sich der Name „Red:wire“, sowie unten im Footer Hyperlinks zu Kontakt und Versionsverlauf. Sollten noch weitere Nebenspunkte benötigt werden, können diese ohne Probleme im Footer hinzugefügt werden. Zwischen Header und Footer liegt vertikal geteilt der restliche Inhalt. Dabei beinhaltet der große Content-Bereich auf der rechten Seite die eigentlichen, aktuell abgerufenen Informationen, während sich in dem verbliebenen Platz am linken Bildschirmrand sowohl das Menü, als auch ein News-Feed befinden. Somit wird es möglich die Navigationsleiste als Orientierung dauerhaft sichtbar zu lassen. Außerdem kann der News-Feed ohne Kollisionen mit anderen Elementen nach unten wachsen.

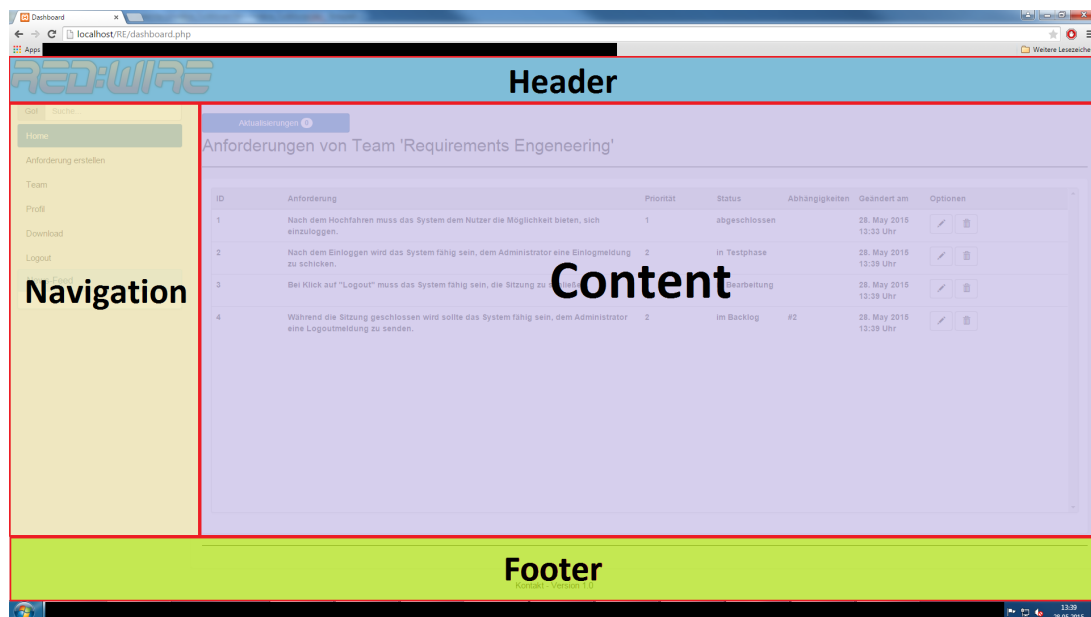


Abbildung 5.2: Layout der Website Aufteilung

Da die Webanwendung die Arbeit erleichtern und nicht mehr Aufwand bedeuten soll, steht bei dem Design der Benutzeroberfläche besonders die Übersichtlichkeit im Vordergrund. Diese wird von der Menüstruktur, den einzelnen funktionalen Bereichen sowie der Positionierung von allein stehenden Elementen beeinflusst. Dafür ist es besonders wichtig, dass Zusammenhänge der Menüpunkte und die aktuelle Position in den Ebenen der Website immer zu erkennen ist. Außerdem sollten die Bezeichnungen der Menüpunkte selbsterklärend oder zumindest eindeutig zuordenbar sein. Zu diesem Zweck sind die Bezeichnungen der Menüpunkte möglichst kurz gehalten (Abbildung 5.3). Die Hauptseite und gleichzeitig Startseite ist „Home“, gefolgt von der häufig benötigten Funktion „Anforderung erstellen“. Dem folgend befindet sich die zwei Verwaltungsmenüs „Team“ und „Profil“. Jene ermöglichen eine Verwaltung der Teams sowie die Bearbeitung der eigenen Accountinformationen. Anschließend gibt es einen Button zum Download, also Export der Anforderungen, gefolgt von dem letzten Menüpunkt „Logout“. Mit diesem wird die aktuelle Sitzung beendet und zur Loginseite zurückgekehrt. Der aktuelle Bereich ist zu jeder Zeit der im Menü blau hinterlegte Punkt. Dadurch genügt ein kurzer Blick um zu wissen, auf welcher Ebene der Website man sich gerade befindet. Direkt über dem Menü befindet sich ein Suchfeld. Dort kann der Nutzer zu jeder Zeit Anforderungen suchen und auf die Hauptseite springen. Unter dem Menü befindet sich ein passiver News-Feed. Mit diesem kann nicht interagiert werden, da der Inhalt automatisch aktualisiert wird und von den Aktionen anderer Teammitglieder abhängt.

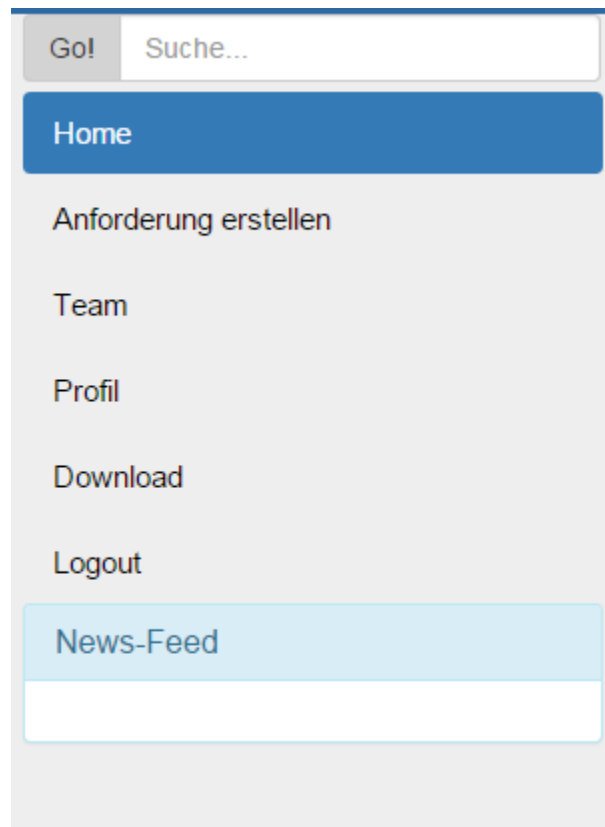


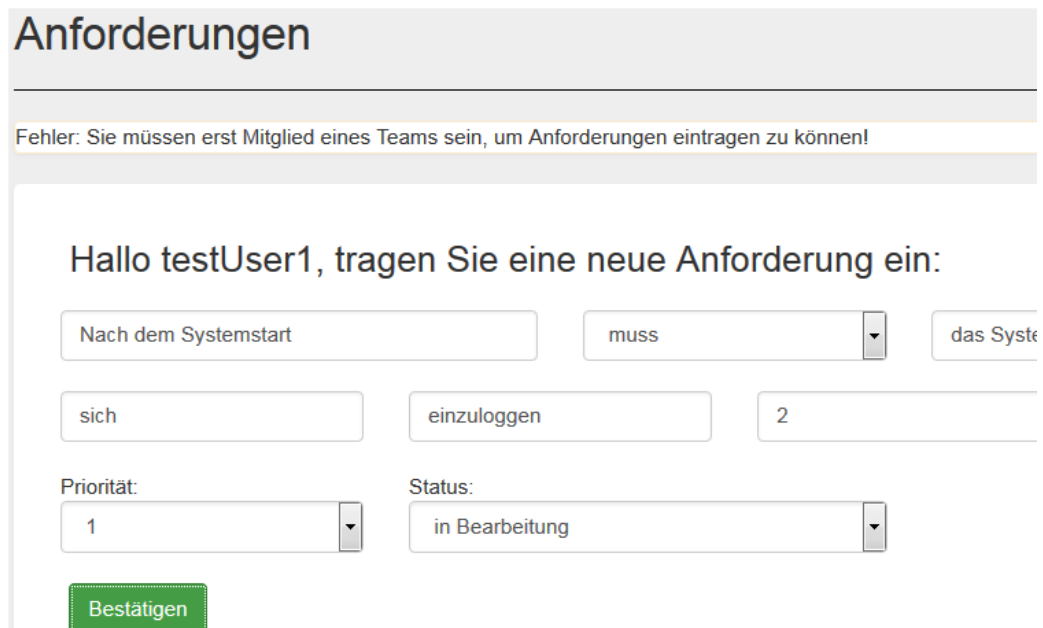
Abbildung 5.3: Screenshot der Navigationsliste

Der Contentbereich verändert sich je nach Nutzeraktionen und wird dynamisch nachgeladen. Dynamisches Laden verhindert, dass jedes Mal die gesamte Seite neu übertragen werden muss und schon die Bandbreite. An oberster Stelle des Bereichs befindet sich ein blauer Button. In diesem wird die Anzahl gefundener Änderungen durch andere Teammitglieder als kleine, bei mehr als 0 Änderungen auffällig rote, Zahl angezeigt. Ein Klick auf jenen aktualisiert den Content und setzt die Zahl auf 0 zurück, wodurch diese weiß wird. Darunter ist in einer Überschrift stets der Teamname des aktuellen Teams zu sehen. Anschließend sind im größten Teil alle Anforderungen, zur besseren Übersicht tabellarisch, dargestellt. Die Spaltennamen hierbei lauten wie folgt: ID, Anforderung, Priorität, Status, Abhängigkeiten, Geändert am, Optionen. In jeder Spalte werden die entsprechenden Daten zu jeder Anforderung dargestellt. Ausnahme ist die Spalte „Optionen“. In dieser sind gut zu erreichen immer zwei Buttons zum Bearbeiten oder Löschen der entsprechenden Anforderung. Sobald diese Tabelle die Grenzen der Content-Bereichs erreicht wird der Inhalt scrollbar, um den Footer nicht unnötig nach unten zu verschieben.

Befindet man sich unter dem Punkt „Anforderung erstellen“, so wird im Content-Bereich die Anforderungsschablone gezeigt. Entsprechend der Anforderungsschablone nach Rupp helfen die Art und Beschreibung der Eingabefelder dabei, das Muster einzuhalten. Zusätzlich bietet sich die Möglichkeit eine eigene ID festzulegen (numerisch), sowie optional Abhängigkeiten anzugeben. Sollte bei der Priorität oder dem Status nichts ausgewählt werden, so haben diese jeweils einen Standardwert (Priorität „0“ und Status „im Backlog“) festgelegt. Dies dient dazu eine gewisse Hochwertigkeit der Anforderungen sicherzustellen.

Die Verwaltungsoptionen „Team“ und „Profil“ des Menüs werden separat in einem modalen Dialogfenster geöffnet. Dies verhindert eine Manipulation der Anforderungen, während man Teams wechselt oder seine Nutzerdaten ändert. Die letzten Optionen „Download“ und „Logout“ öffnen keine extra Fenster, da diese hier nicht benötigt werden. Download speichert direkt eine .csv Datei, sofern der Browser nicht nach einem Speicherort fragt. Diese kann anschließend in z.B. Microsoft Excel importiert werden.

Die gesamte Webanwendung ist größtenteils selbsterklärend. Viele Funktionen werden durch Tooltips beim Mouseover näher erklärt. Außerdem sind nur bekannte Icons, wie etwa ein Mülleimer zum Löschen, verwendet worden. Versucht der Nutzer eine ungültige Aktion, so wird er durch eine kurze Fehlermeldung darauf hingewiesen und sieht, welche Schritte zunächst erfolgen müssen. Dies ist beispielsweise der Fall, bei dem Erstellen einer Anforderung ohne sich in einem Team zu befinden (Abbildung 5.4).



Anforderungen

Fehler: Sie müssen erst Mitglied eines Teams sein, um Anforderungen eintragen zu können!

Hallo testUser1, tragen Sie eine neue Anforderung ein:

Nach dem Systemstart muss das System

sich einzuloggen 2

Priorität: 1 Status: in Bearbeitung

Bestätigen

Abbildung 5.4: Screenshot einer Fehlermeldung

Zur Teamverwaltung gibt es den Menüpunkt mit der eindeutigen Bezeichnung: „Team“. Wie auch beim Punkt „Profil“ öffnet sich ein Dialogfenster. In diesem sind drei Unterpunkte zur Teamverwaltung in einem Accordiondesign so angeordnet, dass stets nur ein oder kein Punkt ausgeklappt ist (Abbildung 5.5). Dadurch wird ein dynamisches Nachladen des Inhalts der einzelnen Accordionpunkte bei einem Klick auf ebensolche ermöglicht. Die Eingabe- und Select Felder, sowie die Buttons sind im selben Design wie beim Punkt „Anforderung erstellen“ gehalten, um ein einheitliches Websitedesign zu behalten. Gleiches gilt für das Dialogfenster des Punktes „Profil“, wobei hier Aufgrund des zur Verfügung stehenden Platzes auf ein Accordiondesign verzichtet wurde.

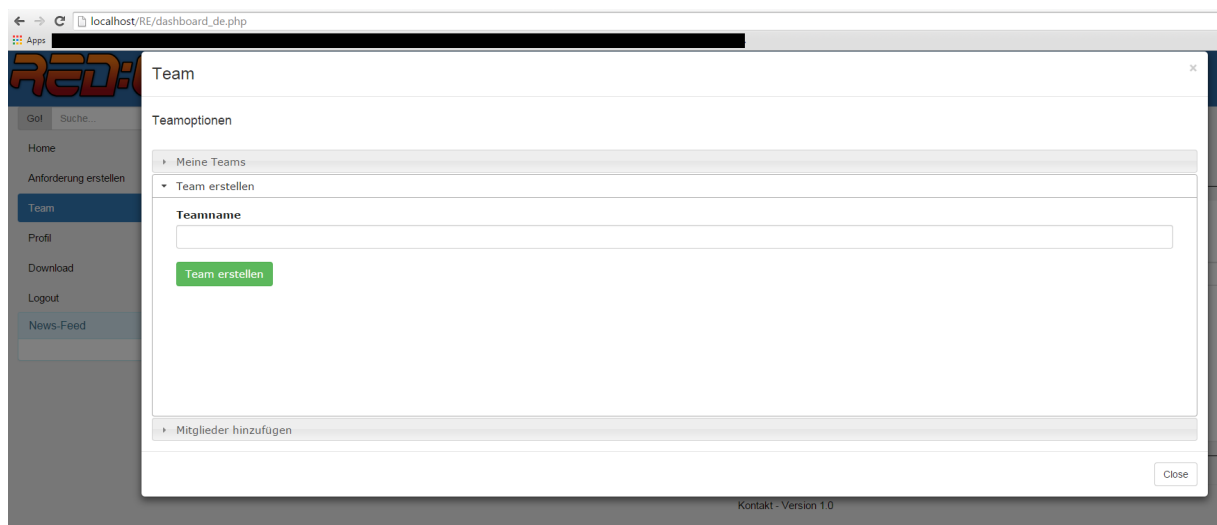


Abbildung 5.5: Screenshot des Team-Dialogfensters

Im Footer der Webseite befindet sich als eine Art Impressum ein Link zu Kontakt, sowie ein Link zum Versionsverlauf der Webanwendung. Das Anklicken jener öffnet jeweils ein modales Dialogfenster. Hier ist es möglich Kontaktdaten der Entwickler oder auch Betreiber zu betrachten oder sich Neuerungen der Webanwendung anzuschauen. Da diese Informationen nicht ständig benötigt werden, befinden sie sich in einem kleinen Teil der Website im Footer.

5.2 Features

- Teams
- > Funktionsweise
- Profil
- Such- und Sortierfunktion
- > Suchfeld
- > Sort by: ID, Name, Priority, Status, Date
- .csv-export (und Anleitung)
- News-feed
- > Benachrichtigung bei Hinzufügen / Ändern / Löschen (momentan alle x Sekunden)
- Administratorbereich (was kann man hier machen?)

6 Ausblick und Fazit

- Zusammenfassung
- > abschließendes review / retrospective
- Entwicklungsherausforderungen
- > Probleme, Schwierigkeiten
- > Was könnte man nächstes Mal besser machen?
- Mögliche zukünftige Features
- > z.Z. User in mehreren Team, PDF export/druck

Abbildungsverzeichnis

1.1	Kostenfaktoren für Fehlerbehebungen [4]	2
2.1	Schnittmenge	3
2.2	Schablone	4
4.1	www.samsung.com Juni 2004 (Screenshot Wayback Machine)	7
4.2	www.samsung.com Juni 2015 (Screenshot)	7
4.3	Webanwendung Konzept	8
4.4	Use-Case Diagramm Nutzeraktionen	10
5.1	Screenshot der Loginseite	19
5.2	Layout der Website Aufteilung	20
5.3	Screenshot der Navigationsliste	21
5.4	Screenshot einer Fehlermeldung	22
5.5	Screenshot des Team-Dialogfensters	23

Tabellenverzeichnis

Literaturverzeichnis

- [1] Authornamen: *Titel*(, *Seite*). URL: <http://www.google.com>.
- [2] Chris Rupp, Matthias Brenner: *Nur qualitativ hochwertige Anforderungen erzeugen hochwertige Produkte*. URL: http://www.sophist.de/index.php?id=180&tx_ttnews%5Byear%5D=2011&tx_ttnews%5Bmonth%5D=11&tx_ttnews%5Bday%5D=09&tx_ttnews%5Btt_news%5D=514 [siehe S. 1].
- [3] Michael Schenkel: *Der Golden Circle im Requirements Engineering*. URL: <http://www.microtool.de/requirements/golden-circle-im-requirements-engineering/> [siehe S. 1].
- [4] Michael Schenkel: *Warum der Umgang mit Anforderungen so wichtig ist*. URL: <http://www.microtool.de/requirements/warum-der-umgang-mit-anforderungen-wichtig-ist/> [siehe S. 2].
- [5] ScrumGuides: *The Scrum Guide*. URL: <http://www.scrumguides.org/scrum-guide.html> [siehe S. 2].
- [6] Takaki Makino, Chaesang Jung, Doantam Phan: *Finding more mobile-friendly search results*. URL: <http://googlewebmastercentral.blogspot.de/2015/02/finding-more-mobile-friendly-search.html> [siehe S. 8].
- [7] Takaki Makino, Doantam Phan: *Rolling out the mobile-friendly update*. URL: <http://googlewebmastercentral.blogspot.de/2015/04/rolling-out-mobile-friendly-update.html> [siehe S. 8].
- [8] Yoshiaki Kato, Pierre Far: *Changes in rankings of smartphone search results*. URL: http://googlewebmastercentral.blogspot.de/2013/06/changes-in-rankings-of-smartphone_11.html [siehe S. 8].
- [9] it-agile GmbH: *Scrum*. URL: <http://www.it-agile.de/wissen/methoden/scrum/> [siehe S. 2].

Anhang

- Installationsanleitung
- Nutzerhandbuch?