

Inhaltsverzeichnis

Begriffserklärung	II
1 Einleitung	1
2 Beschreibung der Aufgabenstellung	2
3 Motivation	5
4 Entwicklung	6
4.1 Konzept	7
4.2 Realisierung	11
5 Webanwendung <i>Red:wire</i>	14
5.1 Oberfläche	15
5.2 Features	18
6 Ausblick und Fazit	19
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

- Grundsätze zu Anforderungen
 - > Anforderungen müssen eindeutig sein
 - > Anforderungen müssen korrekt sein
 - > Anforderungen müssen vollständig sein
 - > Anforderungen müssen konsistent sein
 - > Anforderungen müssen bewertet sein
 - > Anforderungen müssen verifizierbar sein
 - > Anforderungen müssen verfolgbar sein
 - (-> Anforderungen müssen modifizierbar sein)

- > etc. (kennen wir ja)

- Wichtigkeit von Anforderungen
 - > Viele Software Projekte scheitern, weil zu Beginn schon... (hatten wir auch)

2 Beschreibung der Aufgabenstellung

Aufgabenstellung

Die Aufgabenstellung des hier beschriebenen Studienprojektes sieht vor, eine Online-Applikation bzw. Webapplikation zur Anforderungsaufnahme zu entwickeln. Anforderungen sollen mit Hilfe 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.

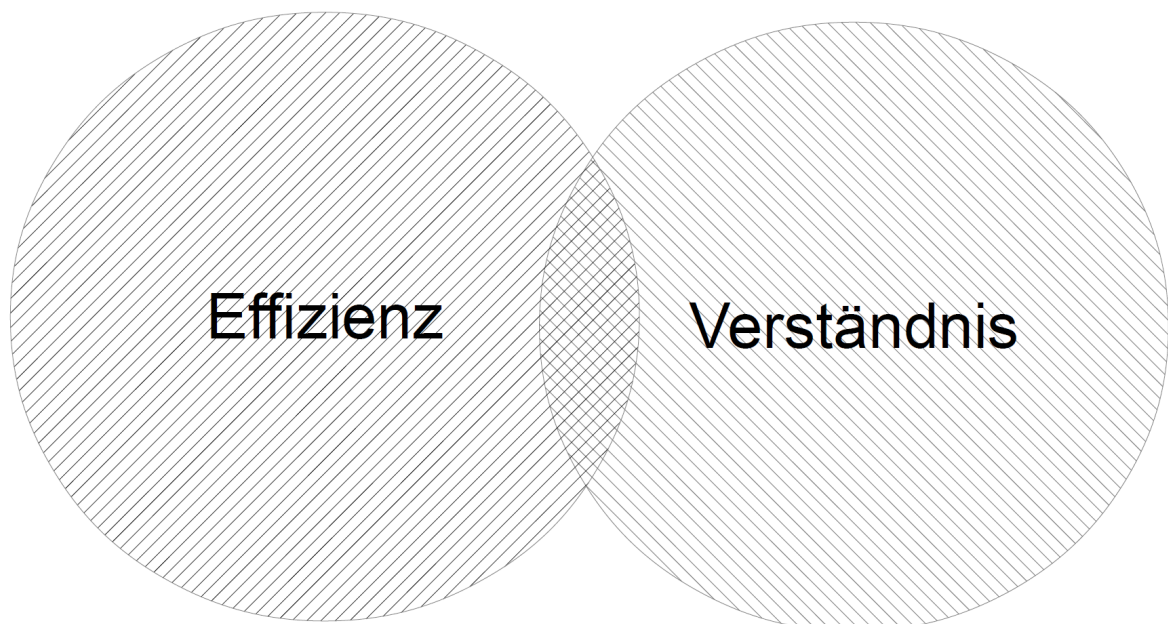


Abbildung 2.1: Schnittmenge

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.

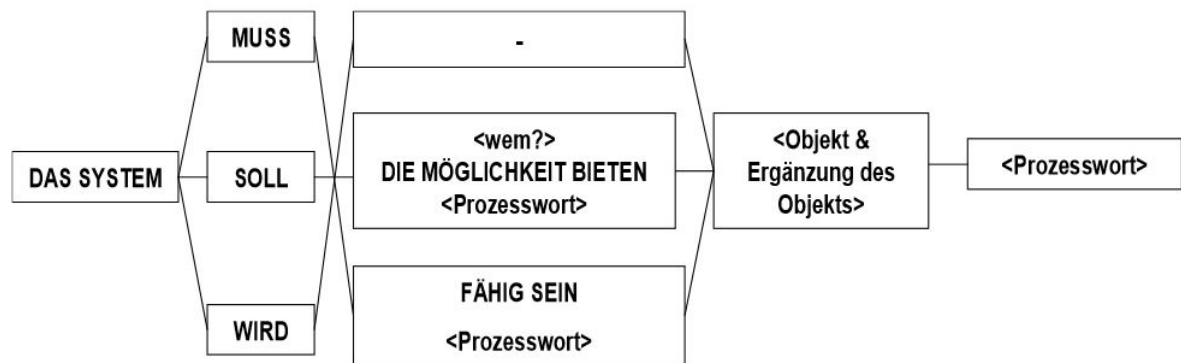


Abbildung 2.2: Schablone

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).

- Entwicklung einer Webanwendung zur Anforderungsformulierung
- Basierend auf vorherigem Studienprojekt[...]
- Formulierungsschablone für Anforderungen
- > (Chris) Rupp Schablone

- Erweiterung um eigene Ideen
- wie kam die idee? wie ist die idee? vision?

3 Motivation

Der Anstoß zur Entwicklung einer Requirements-Software entstammt der Tatsache, dass die Formulierung und Verwaltung von Anforderungen verbessert werden soll. Wenn die Verantwortlichen unachtsam oder ungeübt mit den 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 entsteht schnell Redundanz und Konfusion. Dies bedeutet einen großen Verwaltungsaufwand, was die Kosten und Geschwindigkeit der eigentlichen Produktentwicklung maßgeblich beeinflusst. Um diesen Problemen entgegen zu wirken, wird eine Software entwickelt um Anforderungsmanagement zu vereinfachen. Durch die Eigenentwicklung bietet sich zusätzlich die Gelegenheit individuelle Wünsche und Features umzusetzen.

Infolge des Aufbaus als Webanwendung können Nutzer jederzeit über jeden Browser auf die 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 genutzt werden. 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 solche Teamarbeit wird die Effizienz gesteigert und das Arbeitsklima verbessert. Nach Beendigung des Projektes besteht zudem die Möglichkeit, die Software unter Open-Source-Lizenz zu veröffentlichen und somit unabhängige Weiterentwicklung zu ermöglichen.

- (- wie kam die idee? wie ist die idee? vision?
- Weiterentwicklung vorhandener Software
- > Online zugänglich
- > Multi User (Accountsysteem)
- Arbeit in Gruppen ermöglichen (Teams)
- Möglichkeit für Erweiterungen bieten
- > Open source?)

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.

- beispielsoftware (ähnliche)
- Anforderungen an das System:[...]
 - > Anforderungsformulierung
 - > User Interface
 - > Performance?
 - > Sicherheit?
- Komplette Neuentwicklung, weil [...]
 - > Funktionen einer Webanwendung in diesem Fall günstiger als Java
 - > Bessere Kenntnis über php/javascript
- Arbeitsumgebung: Trello, Github
- tools / tool chain

4.1 Konzept

Zur Entwicklung wurde beschlossen, Software und Programmiersprachen zu verwenden, bei denen Vorkenntnisse vorhanden sind. Eine Hauptkomponente der Webanwendung ist der Server, auf dem das System läuft. Als Server wird der Webserver Apache mit PHP und der Datenbank MySQL aus dem XAMPP Softwarepaket verwendet. XAMPP wurde gewählt, da es eine leichte Installation der benötigten Services ermöglicht. Aus diesem Grund ist XAMPP ein weit verbreitetes Mittel in der Webanwendungsentwicklung. Aufgrund fehlender Sicherheitseinstellungen, folgend aus der automatischen, schnellen Konfiguration, eignet sich dieser Server nicht für den öffentlichen Einsatz. Um direkten Zugriff auf die Datenbank zu erhalten wird das in XAMPP mitgelieferte Werkzeug phpMyAdmin genutzt. Dieses ermöglicht direkten Datenbankzugriff und Manipulation über den Browser. Später erfolgt der Datenbankzugriff über AJAX-Requests mithilfe der zugehörigen JQuery API.

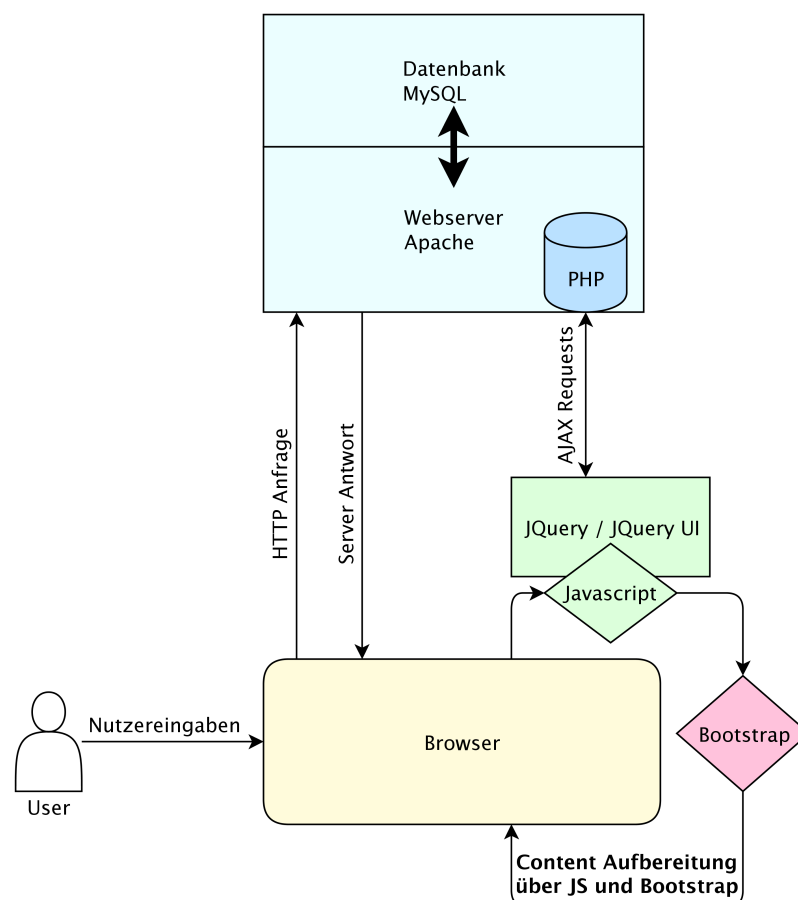


Abbildung 4.1: Systemaufbau

Mithilfe von AJAX können asynchrone Anfragen an den Server gesendet werden. Diese Requests können auf dem Server abgelegte PHP-Dateien ansprechen und ermöglichen es zur Laufzeit Content nachzuladen. Die PHP-Dateien regeln serverseitig Datenbankabfragen und Zugriffe. Ausgelöst werden Anfragen vom Hauptcode der Anwendung aus, bei dieser Software bestehend aus Javascript. Der Javascriptcode ist maßgeblich für die Funktionalität der Website verantwortlich und steuert jegliche Reaktionen auf User-Aktionen. Er ermöglicht beispielsweise die Navigation durch die Seite ohne erneutes Laden der kompletten Website. Um den Funktionsumfang zu erweitern, wird die Bibliothek JQuery eingebunden.

JQuery ist die meist verwendete, frei verfügbare Javascript Bibliothek mit zahlreichen Funktionen und Erweiterungsmöglichkeiten. Es bietet neben DOM-Manipulation ein erweitertes Event-System und eigene Hilfsfunktionen. Zusätzlich werden in diesem Projekte Animationen und Effekte, sowie die Erweiterung JQuery UI genutzt. JQuery UI bietet einen großen Funktionsumfang zur Gestaltung der Benutzeroberfläche. Neben JQuery UI wird zur Gestaltung der Website das CSS-Framework Bootstrap verwendet. Jenes ist eine große Hilfe bei der Nutzung und Gestaltung von Website-Komponenten. Das beschriebene Konzept ist in Abbildung 4.1 dargestellt

Dem Benutzer werden mehrere Funktionen geboten, ein Überblick gibt die Abbildung 4.2. 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 dem Eingeloggen, 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 4.2 leicht grün hinterlegt. Jene Anforderungen sollen Teamgebunden und vom gesamten Entwicklerteam einsehbar sein, das heißt für jeden der Mitglied im entsprechenden Team ist. Ist der Benutzer der Ersteller oder Mitglied eines Teams, so sollte er erweiterte Möglichkeiten haben das Team zu bearbeiten. Entsprechende Aktionen bezüglich der Teams sind in der Abbildung 4.2 hellblau hinterlegt. Ein Nutzer sollte selbständig seine Accountinformationen wie Passwort und Email ändern können. Jene Nutzerbezogenen Aktionen sind in der Abbildung 4.2 hellgelb hinterlegt. Um das System zu Warten, sollte ein Administrator zumindest die Möglichkeit haben Nutzer zu löschen.

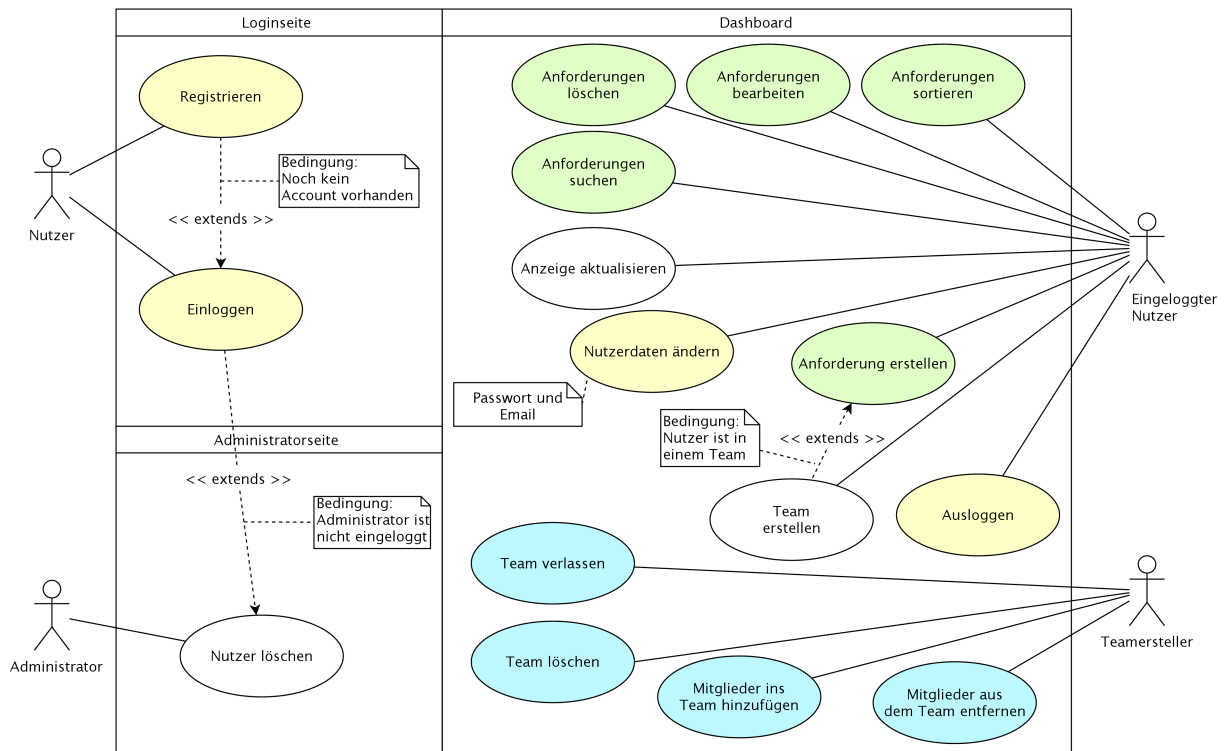


Abbildung 4.2: Use-Case Diagramm Nutzeraktionen

Zunächst ist es Ziel ein funktionierendes Anmeldesystem zu entwickeln. Dies ist notwendig, da die Software direkt auf Basis eines **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 Zeitaufwand von **32 Arbeitsstunden** veranschlagt. Ebenso ist dieser Schritt ein erster Meilenstein. Nachdem das Benutzersystem auf seine ordnungsgemäße Funktion getestet wurde, richtet sich die Entwicklung zunächst 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. Funktioniert alles wie gewünscht, so ist der zweite Meilenstein erreicht.

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.

(- zeitplanung / milestones -> aufwandsschätzung

- Entwicklung mit uns bekannten Sprachen/Frameworks (MySQL, PHP, JS, JQuery, JQueryUI, Bootstrap), weil[...]

-> bisschen was zu den Sprachen (Nutzen, Anwendung)

- Use Cases

- mensch-maschine interaktion (aufgreifen der vorlesungsinhalte) -> was soll der User tun können?)

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

- UML-Diagramme (Codestruktur / Sequenzdiagramme / ...)
- Zusammenhänge der Dateien
- kommunikation / datenaustausch
- Abläufe (Login / DB-Zugriffe / Content erzeugen / update())
- Tests

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.

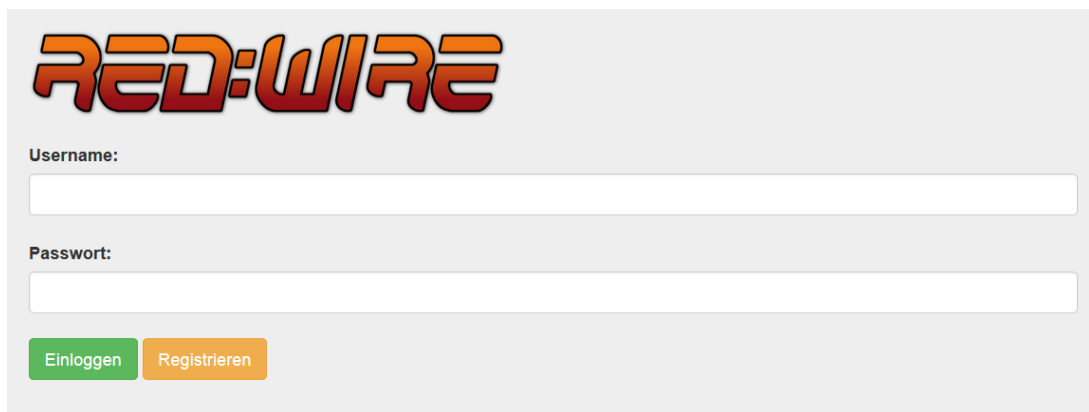


Abbildung 5.1: Screenshot der Loginseite

Die Struktur der Hauptseite setzt sich aus 4 Bereichen zusammen (Abbildung 5.2). Dieses Schema 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 Nebepunkte 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 die Navigationsliste, 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.

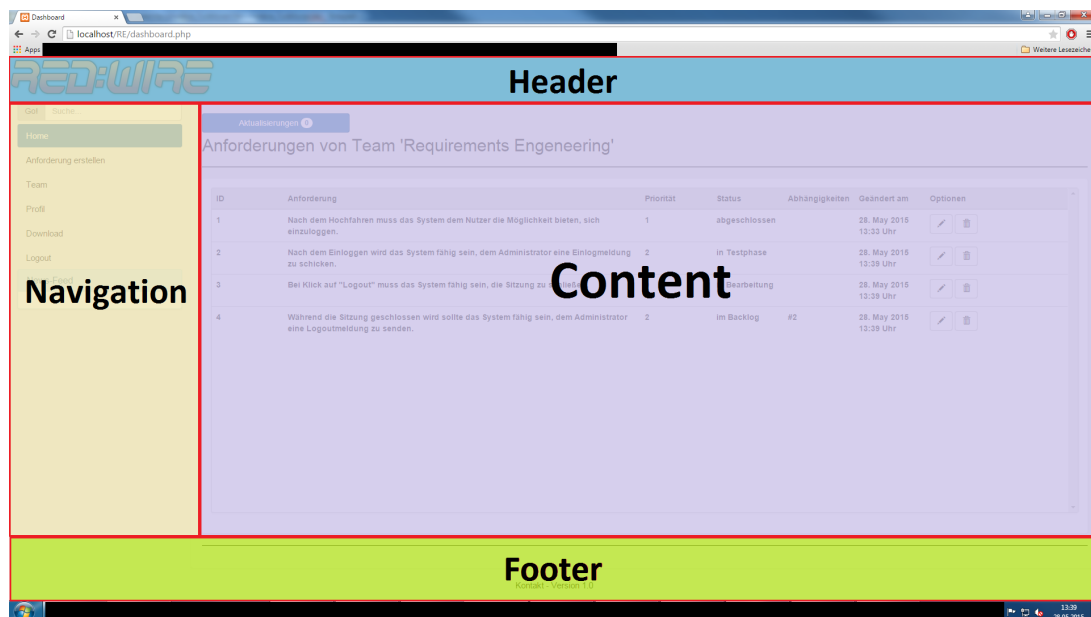


Abbildung 5.2: Schema der Website Aufteilung

Damit 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 Navigationslistenstruktur, den einzelnen funktionalen Bereichen sowie der Positionierung von allein stehenden Elementen beeinflusst. Dafür ist es besonders wichtig, dass Zusammenhänge der Navigationslistenpunkte und die aktuelle Position in den Ebenen der Website immer zu erkennen ist. Außerdem sollten die Bezeichnungen der Navigationslistenpunkte selbsterklärend oder zumindest eindeutig zuordenbar sein. Zu diesem Zweck die Bezeichnungen der Navigationslistenpunkte 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 Navigationslistenpunkt „Logout“. Mit diesem wird die aktuelle Sitzung beendet und zur Loginseite zurückgekehrt. Der aktuelle Bereich ist zu jeder Zeit blau hinterlegt. Dadurch genügt ein kurzer Blick um zu wissen, auf welcher Ebene der Website man sich gerade befindet. Direkt über der Navigationsliste befindet sich ein Suchfeld. Dort kann der Nutzer zu jeder Zeit Anforderungen suchen und auf die Hauptseite springen. Unter der Navigationsliste 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.

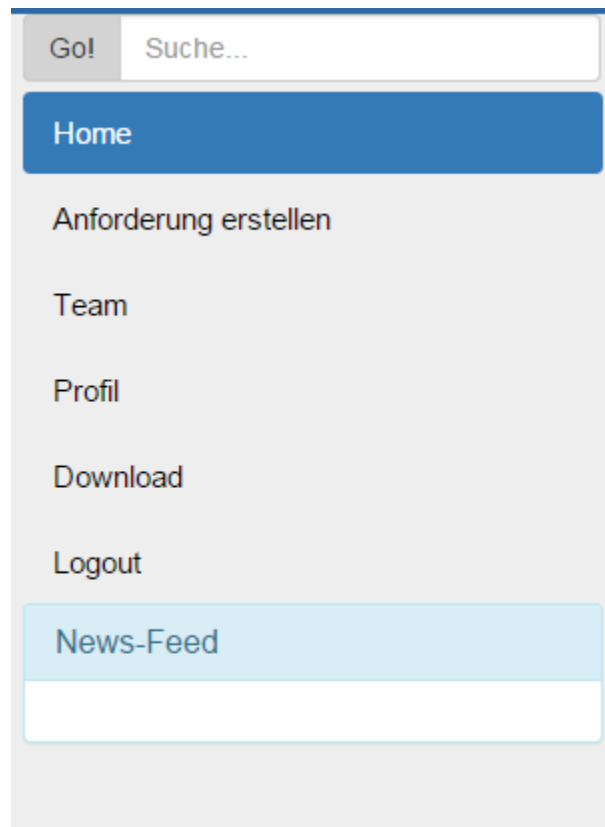


Abbildung 5.3: Screenshot der Navigationsliste

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 zu sehen, in welchem Team man gerade arbeitet. Anschließend sind im größten Teil alle Anforderungen zur guten 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. Eine Priorität und ein Status sind immer ausgewählt, da diese ein Schritt in Richtung „Gute Anforderungen“ sind.

Die Verwaltungsoptionen „Team“ und „Profil“ der Navigationsliste werden separat in einem modalen Dialogfenster geöffnet. Dies verhindert eine Manipulation der Anforderungen, während man Teams wechselt oder seine Nutzerdaten ändert.

- Grafisches Layout
- > Farben, Buttons, Platzierung
- Menüführung
- Nutzerfreundlichkeit / Nutzeraktionen

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

2.1	Schnittmenge	2
2.2	Schablone	3
4.1	Systemaufbau	7
4.2	Use-Case Diagramm Nutzeraktionen	9
5.1	Screenshot der Loginseite	15
5.2	Schema der Website Aufteilung	16
5.3	Screenshot der Navigationsliste	17

Tabellenverzeichnis

Literaturverzeichnis

[1] Authorname: *Titel*(, *Seite*). URL: <http://www.google.com>.

Anhang

- Installationsanleitung
- Nutzerhandbuch?