Projektmanagement

SE-Welcome App

# Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc439502482)

[1 Projektbeschreibung 4](#_Toc439502483)

[1.1 Ausgangsituation 4](#_Toc439502484)

[1.2 Zielsetzung 4](#_Toc439502485)

[1.2.1 Grobkonzept 4](#_Toc439502486)

[1.2.2 Zielgruppe 4](#_Toc439502487)

[1.2.3 Funktion und benötigte Rechte der Benutzergruppen 4](#_Toc439502488)

[1.2.4 Allgemeine Beschreibung der Funktionen 4](#_Toc439502489)

[1.2.5 Genutzte Plattformen 5](#_Toc439502491)

[1.2.6 Rahmenbedingungen 5](#_Toc439502492)

[1.2.7 Technische Umgebung 5](#_Toc439502493)

[1.2.8 Einsatzbedingungen 5](#_Toc439502494)

[1.2.9 Rechtliches: 5](#_Toc439502495)

[1.2.10 Vorkenntnisse 5](#_Toc439502496)

[1.2.11 Einschränkungen 6](#_Toc439502497)

[1.3 Meilensteine 6](#_Toc439502498)

[2 Organisation 6](#_Toc439502499)

[2.1 Aufbauorganisation 6](#_Toc439502500)

[2.1.1 Projekt Hierarchie 6](#_Toc439502501)

[2.1.2 Taskforces 7](#_Toc439502502)

[2.2 Skillsheet 7](#_Toc439502503)

[3 Ablaufsteuerung 9](#_Toc439502504)

[3.1 Phasenmodell 9](#_Toc439502505)

[3.1.1 Erste Planung 9](#_Toc439502506)

[3.1.2 Zweite Planung 9](#_Toc439502507)

[3.2 Scrum 10](#_Toc439502508)

[3.2.1 Scrum im Projekt 10](#_Toc439502509)

[3.3 Extreme Programming 10](#_Toc439502510)

[4 Risikomanagement 11](#_Toc439502511)

[4.1 Stakeholderanalyse 11](#_Toc439502513)

[4.1.1 Intern 11](#_Toc439502514)

[4.1.2 Extern 11](#_Toc439502515)

[4.2 Umweltanalyse 11](#_Toc439502516)

[5 Das Softwaresystem 11](#_Toc439502517)

[5.1 Anwendungsfall Beschreibung 11](#_Toc439502518)

[5.2 Anwendungsfalldiagramm 11](#_Toc439502519)

[5.3 Aktivitätsdiagramm 11](#_Toc439502520)

[5.4 Klassendiagramm 11](#_Toc439502521)

[5.5 Sequenzdiagramm 11](#_Toc439502522)

[5.6 Zustandsdiagramm 11](#_Toc439502523)

[5.7 Auswertung der Diagramme 11](#_Toc439502524)

[6 Anhang 11](#_Toc439502525)

[7 Literaturverzeichnis 12](#_Toc439502526)

# Projektbeschreibung

## Ausgangsituation

## Zielsetzung

### Grobkonzept

Mithilfe der mehrsprachigen Web-Plattform soll es Flüchtlingen und besonders Asylsuchenden vereinfacht werden, sich in Deutschland zurecht zu finden. Um dies zu erreichen sollen möglichst alle benötigten Informationen für ein schnelles Zurechtfinden und gute Integration gesammelt und aufbereitet werden, und dazu weitere hilfreiche Funktionen integriert werden. Diese sollten möglichst genau auf die Anforderungen der Flüchtlinge zugeschnitten sein, unter anderem eine sehr einfache Benutzeroberfläche, die sofort verständlich ist und intuitiv bedienet werden kann. Im Zuge der Projektarbeit nutzten wir bereits bestehende Anwendungen anderer Webseiten oder verlinkten auf diese, bspw. Open Street Map oder ein Wiki.

Darüber hinaus stellt es die Prüfungsleistung der DWI 13 in dem Kurs „Softwareengineering“ dar. Ein Anspruch an die Software wie vergleichbare Projekte muss gegeben sein, um ein entsprechend gutes Ergebnis präsentieren zu können.

### Zielgruppe

Die Zielgruppe der zu erstellenden Website/App sind zum einen die Asylbewerber und Asylsuchende, zum anderen die Helfer.

### Funktion und benötigte Rechte der Benutzergruppen

Nicht eingeloggter normaler Benutzer (z.B. Asylbewerber): keine besonderen Rechte. Kann alles anzeigen, aber nichts bearbeiten. Benutzt die Website genauso wie das Wikipedia. Gespeicherte Daten: keine notwendig für den Hauptprozess

Eingeloggter Benutzer (Helfer): erweiterte Benutzerrechte. Kann das Wiki bearbeiten. Benutzt die Website genauso wie das Wikipedia. Gespeicherte Daten: Die Session des Nutzers, die Benutzerrechte bzw. Rolle und den Verein für den er tätig ist (bestimmt die Bearbeitungsrechte).

Administrator: vollständige Rechte. Er darf alles bearbeiten, neu anfügen, löschen usw. Benutzt alle bereitgestellten Funktionen, auch welche über die Website hinaus. So z.B. um einen Neuen Build auf den Application Server zu pushen. Gespeicherte Daten: Die Session des Nutzers, die Benutzerrechte bzw. Rolle.

### Allgemeine Beschreibung der Funktionen

Wir nutzen verschiedene Funktionen aus bereits existierenden Webseiten und Anwendungen, dabei spezifizieren wir den Umfang nach unseren Anforderungen. Hier folgen die zu implantierenden Funktionen:

* Übersichtliche, thematische, einfach zugängliche Informationssammlung
  + Behördenkontaktdaten zu den jeweiligen Standorten
  + Rechtliches Einmaleins
  + Anlaufstationen für z.B. kurzfristige Unterkünfte / Tafel / DRK
  + Infos der lokalen Refugee Law Clinics / andere lokale Vereine
  + häufig genutzte Rufnummern (Notrufnummern) + deren Aufgabengebiete
  + Gesundheitssystem Kostenlose Ärzte Wo finde ich einen in meiner Nähe
  + Verknüpfungen zu anderen Hilfeseiten und Funktionalitäten
  + …
* Sofern möglich – Ermittlung des Standortes und Darstellung der gesuchten Informationen auf einer Karte („wo bin ich und wo muss ich hin“)
* Ein Wikipedia – dort kann gemeinschaftlich dran gearbeitet werden. Inhaltlich dreht es sich um tiefergehende Infos zu Standorten (Freizeitaktivitäten bspw.)
* Ein Forum, aufgebaut wie ein schwarzes Brett und standortspezifisch, um Dienste anzugeben. Also eine Plattform für Helfer und hilfesuchende – was brauche ich genau? Wer kann mir helfen?
* Freizeitbeschäftigung: Map/Wiki/Forum -> was kann ich in meiner Umgebung machen? Fußball etc.
* Eine Funktion die erlaubt, herauszufinden welche Materialien ich in welchem Laden finde. Was gibt es im Baumarkt, was finde ich im Supermarkt. Wie heißen die Supermärkte? Und dann direkt eine Integration „auf Karte anzeigen“ – dort werden dann die benötigten Kategorien eingeblendet. (fließender Übergang)
* Suchfunktion: Einfaches Schlagwortfiltern. Ich möchte mich nicht durchklicken? Okay – ich such danach
* Formular Template: Deutschland hat ein Bürokratischer Djungel der allein für deutsche nicht immer durchschaubar ist – für Menschen die dann nicht einmal deutsch sprechen eine noch viel größere Herausforderung. Eine Standortbezogene Auflistung der benötigten Formularen mit Ausfüllhilfe: Was bedeutet das Feld oder das? Was muss ich hier eintragen?

### Genutzte Plattformen

In der ersten Version fertigen wir eine Webseite an, die man aus den gängigen Browser abrufen kann. Diese Webseite wird immer weitergepflegt und erweitert. Zusätzlich soll in den nächsten Version (Version 2 und höher) auch eine App für Smartphones entwickelt werden.

### Rahmenbedingungen

### Technische Umgebung

Bei den technischen Lösungen sollen keine monetären Kosten anfallen, daher werden wir überwiegend auf Open-Source Software nutzen. Das verwendete Modell soll eine Tier-3 Architektur sein – somit sind Skalierungsmöglichkeiten, falls ein durchschlagender Erfolg eintritt, gegeben.

Als Client kommen alle Geräte in Frage, welche Webseiten darstellen können. (Desktop PC, Laptop, Tablet, Smartphone).

### Einsatzbedingungen

Geltende Sicherheitsstandards müssen erfüllt sein. Insbesondere sobald Nutzerdaten gespeichert werden, ist dies unumgänglich.

Die Webseite sollte innerhalb üblicher Reaktionszeiten agieren (max. 3 Sekunden beim Kunden).

### Rechtliches

### Vorkenntnisse

* Die einzigen Vorkenntnisse besitzen wir aus den Inhalten, des im Rahmen des dualen Studiums der Wirtschaftsinformatik angeeignet haben. Darunter fallen Programmierkenntnisse, wie das Programmieren mit u. a. Java, SQL, sowie Projektmanagementkenntnisse (unterstützt mit Microsoft Project).

### Einschränkungen

* Durch fehlendes tiefergehendes Know how an vielen Stellen, ist eine 80% Lösung ausreichend. Es muss kein technisches Meisterwerk sein – vor allem da nur sehr begrenzte Zeit vorhanden ist im Rahmen des Projektes. Weiterentwicklung nach Abschluss der Ersten Phase steht derzeitig nicht fest. Aufgrund begrenzter Ressourcen wird eine fertige Lösung nicht möglich sein. Die erste Version (diese wird Frau Wieland als Abschlussarbeit eingereicht) bedarf weitere Arbeit, um alle Anwendungen zu implementieren und eine App zu erstellen. Die Weiterentwicklung nach der ersten Version ist noch nicht festgelegt.

(Klodinski, S. 1-3)[[1]](#footnote-1)

## Meilensteine

https://docs.google.com/spreadsheets/d/1Ssdn-gzXDYjQ3ldQSXaKPz8bGzdTRwECd5rJbsxSwsw/edit?usp=sharing

# Organisation

## Aufbauorganisation

### Projekt Hierarchie

Das Projekt wurde mit drei Teilprojektgruppen initialisiert. Diese sollten die unterschiedlichen Aspekte des Projekts vorbereiten und eine thematisch untergliederte Basis bilden. In den späteren Phasen (siehe 3.1) werde diese Gruppen funktional weitestgehend aufgelöst.

|  |  |  |  |
| --- | --- | --- | --- |
| Projektlead | Philipp Staats | | |
|
|
|  | Team 1 | Team 2 | Team 3 |
| Lead | Philipp Staats (134147) | Stefan Ludowicy (134131) | Nico Remus (134152) |
|  | Aras Mousavii (134242) | Philipp Kneist (134159) | ~~Tobias Karhof  (134124)[[2]](#footnote-4)~~ |
|  | Sheldon Jesudasan (134158) | Sebastian Treske (134153) | Jasmin Clos  (134108) |
|  | Christoph Ramp (134136) | Jan Bickmann  (134102) | Jean Klodzinski (134125) |
|  | Eric Skowronski  (134145) |  |  |

[[3]](#footnote-5)

#### Projektleiter - PL

Die Gesamtprojektleitung übernimmt Philipp Staats.

Die Rolle des Projektleiters dient der Gesamtkoordination des Projekts in Abstimmung mit den Teilprojektleitern. Der Schwerpunkt liegt hierbei im Controlling.   
Zu den Controllingaufgaben werden die Steuerung der Teams sowie der Gruppenmeetings, aber auch Einhaltung von Terminen zählen. Dabei wird der Projektleiter in direkter Kommunikation mit dem PO (Fr. Wieland) stehen und über projektspezifische Missstände aber auch Erfolge in Kenntnis setzen. Er ist für das Risikomanagement verantwortlich, so dass bei projektgefährden Ereignissen der Handlungsbedarf von Ihm ausgehen muss.

Fachliche Fragestellungen werden sind im Detail von den Know-How Trägern in den Task-Forces (siehe 2.1.2) beschlossen.

#### Teilprojektleiter – TPL

Die Unterteilung des Projekts in drei Teilprojektgruppen dient der Entlastung des Projektleiters. Jede Teilprojektgruppe hat einen thematischen Schwerpunkt, nach welchem diese dem Projekt zuarbeitet. Gemeinsam leiten sie das Gesamtprojekt.

1. Research - Philipp Staats
2. Technische Entwicklung - Stefan Ludowicy
3. Solution Design - Nico Remus

#### Projektmitglied - PM

### Taskforces

*(Aufbau nach Einsetzen der Scrum-Methodik )*

#### Anfoteam

Das komplette Anforderungsmanagementteam bestand aus folgenden Personen:

Jasmin Clos, Stefan Ludowicy, Philipp Staats, Jean-Pierre Klodzinski, Nico Remus, Tobias Karhof (der uns leider verlassen hat)

Das Anfo-Team ist dafür zuständig neue Anforderungen in Form von Stories zu definieren und die Ergebnisse zu überprüfen. Darüberhinaus kümmert es sich um die Steuerung des Teams, falls es intern Probleme gibt. Probleme sind unter anderem aufkommende Fragen oder wenn es einen Disput zwischen den Mitgliedern geben sollte.

#### Scrum Masters[[4]](#footnote-6)

Die beiden Scrum Master lauten:

Stefan Ludowicy und Jean-Pierre Klodzinski

Die Scrum Master sind dafür zuständig das Scrum Board zu pflegen, Fragen und Probleme zu lösen wie der Disput zwischen Mitgliedern, fehlende Ausarbeitungen oä. und haben in letzter Instanz die höchste Entscheidungsgewalt bezüglich Stories.

#### Translators

#### Development

#### Redaktion und Frontendlayout

*(Frontendtest, Usability)*

## Skillsheet

*(Ermittlung von Know-How-Trägern zur Bildung von Self-teaching Teams->1erfahren+1semierfahren)*

Mit einem Gesamtprojekt übergreifenden Skillsheet werden besondere Fähigkeiten und Vorkenntnisse der einzelnen Projektmitglieder ermittelt. Daraus Ergibt sich ein Kompetenzenpool, auf welchem die weitere Konzeption des Projekts beruht.

# Ablaufsteuerung

## Phasenmodell[[5]](#footnote-7)

*Warum Phasenmodell*

*Warum wechsel auf neues Modell*

### Erste Planung

* 1. 1.Phase (2.Wochen)
     1. Team 1 Staats Recherchiert/ Erfasst Content  
        [Erste Recherche nach Drittdienstleister]
     2. Team 2 Ludowicy Einarbeitung in die Entwicklungsplattformen
     3. Team 3 Remus AS(AnforderrungsSpezifikation) - Erstspezifikation

-> Eine Woche später FSL (Fachspezifsche Lösung) nach vorstellung des Konzepts

* 1. 2. Phase (2.Wochen)
     1. Team 1 Staats Recherche Zielgerichtet nach FSL/ Beginn Übersetzungen  
        Recherche und Abstimmung mit Team 2, wie Drittanbieter eingebunden werden können
     2. Team 2 Ludowicy Anwendungs-Gerüst
     3. Team 3 Remus FSS mit CoOp Team 2
  2. 3. Phase (2.Wochen)
     1. Team 1 Staats Hauptphase der Übersetzungen  
        Kontakt mit Drittanbietern herstellen
     2. Team 2 Ludowicy Anwendungs-Skelet
     3. Team 3 Remus Graphische Spezifikation und Elemente/ Zuarbeit zu Team 2
  3. 4. Phase...

### Zweite Planung

* 1. 1.Phase (2.Wochen)
     1. Team 1 Staats Recherchiert/ Erfasst Content  
        [Erste Recherche nach Drittdienstleister]
     2. Team 2 Ludowicy Einarbeitung in die Entwicklungsplattformen
     3. Team 3 Remus AS(AnforderrungsSpezifikation) - Erstspezifikation

-> Eine Woche später FSL (Fachspezifsche Lösung) nach vorstellung des Konzepts

* 1. 2. Phase (2.Wochen)
     1. Team 1 Staats Recherche Zielgerichtet nach FSL/ Beginn Übersetzungen  
        Recherche und Abstimmung mit Team 2, wie Drittanbieter eingebunden werden können
     2. Team 2 Ludowicy Anwendungs-Gerüst
     3. Team 3 Remus FSS mit CoOp Team 2
  2. 3. Phase (2.Wochen)
     1. Team 1 Staats Hauptphase der Übersetzungen  
        Kontakt mit Drittanbietern herstellen
     2. Team 2 Ludowicy Anwendungs-Skelet
     3. Team 3 Remus Graphische Spezifikation und Elemente/ Zuarbeit zu Team 2
  3. 4. Phase...

## Scrum

Scrum als agile Vorgehensmethode beruht auf so genannten Sprints. Jeder Sprint beginnt mit einem Planning, in dem die Aufgaben an das Team verteilt verteilt werden und endet mit dem Review, in dem die verteilten Stories vorgestellt und vom Product Owner abgenommen werden. Die Sprints dauern zwischen einer und meiste vier Wochen, basierend auf der Komplexität der Aufgaben. Mehr oder weniger ist selbstverständlich möglich.

Scrum als sehr bekannte Form der agilen Vorgehensmethoden kam auch für unser Projekt in Frage. Als Alternativen boten sich Extreme Programming und das Wasserfallmodell. Wir entschieden und gemeinschaftlich für Scrum weil es verschiedene Vorteile resultierend aus dem Vorgehen bietet. So musste nicht von Anfang an das komplette Softwaresystem ausspezifiziert worden sein. Es war uns möglich durch die Sprints nach und nach die Dinge zu durchdenken. Darüberhinaus hatten wir nach jedem Sprint ein besseres Gefühl bis wann das Projekt etwa fertig sein wird, bzw. „wie weit“. Auch lässt sich mit Scrum besser die Last im Team verteilen, da jede Woche neu „gewählt“ wurde.

### Scrum im Projekt

Die Scrum vorgehensweise wurde in unserem Team gemeinschaftlich beschlossen und fast vollständig umgesetzt. Wir verstehen Scrum als Richtlinie wonach wir die mehrheitlichen Regeln befolgen. So haben wir eine Menge fester Rollen haben, welche ein festes Aufgabengebiet übernehmen. Jedoch kann in unserem Team eine Person mehrere Rollen gleichzeitig wahrnehmen und wirkt entsprechend anders an dem Fortschritt des Projektes mit. Darüber hinaus ist auch das Vorgehen, mit Sprint, Planning, Review, Retrospektive (war sehr wichtig für das Anfo-Team) dem Scrum Vorbild nachempfunden. Jedoch verzichteten wir auf Daylies und haben stattdessen eine selbstständige Eintragung in unserer Story Bord in Form von Kommentaren bevorzugt. Dies reduzierte die extreme organisationslast auf Seiten der Scrum Master ein wenig und jeder konnte sich einteilen wann er welche Aufgaben macht. Wichtig war uns das die Aufgaben erledigt werden innerhalb des Zeitraums – nicht aber auf welche Art und Weise. Das Ergebnis musste stimmen. Als weiteren Unterschied wurden die Stories kategorisiert. Dies half uns bei der Verteilung den jeweilig richtigen Personen die Aufgaben zu überreichen. Selbstständiges Wählen der Aufgaben und raten durch die Mitglieder des Teams hat leider nicht wie erwartet funktioniert.   
Zum Ende des Projektes, also über die Weihnachtsfeiertage, als die Zuarbeit des Teams gegen null lief, wurde das Vorgehen auf extreme Programming umgestellt. Eine kleine Gruppe setzte dann alle notwendigen Funktionen um.

Hier eine kurze Zusammenfassung unserer Rahmenwerte:

#### Agil

* Sprintlänge: 1 Woche

Story - Creation Team

siehe 2.1.1 Anfo Team.

#### Kategorisierung der Stories

Unterscheidung der Stories in:

* + Entwicklung
  + Research
  + Design
  + Visual Design
  + Pre-Dev

Storypoints

= Arbeitstunden (60min reale Arbeit)

#### Scrum-Storyboard[[6]](#footnote-9)

Das Story Board ist das zentrale Fortschritts-Messinstrument gewesen. Auf dem Story Bord sind alle Aufgaben hinterlegt. Es zeigt den aktuellen Status (Fertig, in Bearbeitung und offen) sowie welche Personen zuständig sind und wie komplex die Aufgabe scheint. Die Idee zum Story Bord kam während des ersten Sprints zur Erfassung des Fortschritts und um Motivation bei den Gruppenmitgliedern zu fördern. Auf dem Board sieht man sehr gut wie viel Arbeit noch getan werden muss. Hauptverantwortlich für die Anforderungsdefinition war Jean-Pierre Klodzinski (in Zusammenarbeit mit Stefan Ludowicy und Philipp Staats) und letztendlich umgesetzt wurde es von Nico Remus (in Kooperation mit Philipp Staats)

## Extreme Programming

*Funktionalität der CoOp Arbeit von 2-3 Entwicklern + Backup*

*[[7]](#footnote-10)*

# Risikomanagement



## Stakeholderanalyse

### Intern

#### Projektmitglied fällt temporär aus

#### Projektmitglied fällt dauerhaft aus

### Extern

#### Frau Dr.Wieland

#### Cooperation mit T-Systems allgemein

#### Cooperation mit telekom.refugee.de

## Umweltanalyse

#### Flüchtlingskriese könnte sich auflösen

#### Server brechen weg

#### Software Lizenzen ändern sich (/laufen aus)

#### Eingriff des Telekomkonzerns in die Projekttätigkeit

#### Technisches Know How unzureichend

#### Verlust der GitHub Plattform

#### Supporteinstellung von Softwareanbietern

# Das Softwaresystem

In diesem Kapitel werden die für die Prüfungsleistung benötigten Dokumente und UML Diagramme hinterlegt. Sofern ein bestimmter Anwendungsfall gewünscht war, wird im folgenden die Funktionalität „Wo Gibt Es Was“ benutzt. Folgend eine Wörtliche Beschreibung aus dem Solution Design Dokument:

*Mit der Funktion "Wo gibt es was" soll dem Benutzer ermöglicht werden auf einfache Art und Weise herauszufinden, in welchem Laden er welche Dinge kaufen kann.*

*Um das möglichst einfach zu gestalten, wählt der Nutzer als erstes ein Bild einer Produktkategorie aus (Beispielsweise Nahrung). Neben dem Bild findes sich noch der Kategoriename.*

*Anschließend hat er die Möglichkeit eine Unterkategorie zu wählen, oder direkt sich die Shops anzeigen zu lassen.*

*Falls er eine Unterkategorie auswählt, wird er zu einer Übersichtsseite weitergeleitet. Auf dieser wird die Shopkategorie angezeigt*

*(zb. Supermarkt) und die dazugehörenden Shopnamen (Lidl, Rewe, Aldi), sowie eine Übersicht von verfügbaren Produkten (begrenzt auf x Werte). Zu den Shops gibt es dann noch eine grobe Richtung in welcher Preisklasse, Angebotsvielfalt und Qualität dieser liegt: Beispiel: Alid 1, Lidl 2, Rewe und Edeka 3.*

*Falls er jedoch direkt wissen möchte wo er beispielsweise Nahrung bekommt, dann klickt er auf einen Button wie zb. "Wo bekomme ich das?" und bekommt gleich die Übersichtseite samt Shopliste angezeigt und einer*

*Menge von Beispielprodukten (Nudeln, Reis, Tomaten, ...).*

*Darüberhinaus hat er auf der Übersichtsseite die Möglichkeit eine Verlinkung zur Karte zu öffnen über einen Klick auf den Button "In meiner Nähe"/"der nächste in meiner Nähe"/"Wo finde ich diese". Damit öffnet sich ein Fenster*

*mit einer Karte gefiltert nach der ausgewählten Shop-Kategorie. Er kann dort nun seine Adresse eingeben und ein Ziel auswählen und die Navigation starten.*

*Alternativ ist es dem Benutzer möglich, direkt nach Produkten zu Suchen (Bspw. Reis). Falls ein Eintrag gefunden wird, wird die Übersichtsseite angezegit. Falls nicht eine Fehlermeldung: "Hupps - das gewünschte Produkt konnte keiner*

*Kategorie zugeordnet werden" Klicke hier um einen erneuten Suchvorgang zu starten. Wenn mehrere Einträge gefunden werden, dann gibt es eine Übersicht mit den Ergebnissen welche gefunden wurden. Daraus kann der Benutzer dann auswählen und gelangt dann zur Übersichtsseite.*

*Die suche soll partiell und vollständig und familiär sein. und alles irendwie anders. Beispiel:*

*DB: Paprika und Nüsse*

*Suche: Paprikaschoten -> Ergebnis: Paprika*

*Suche: Erdnüsse -> Nüsse[[8]](#footnote-11)*

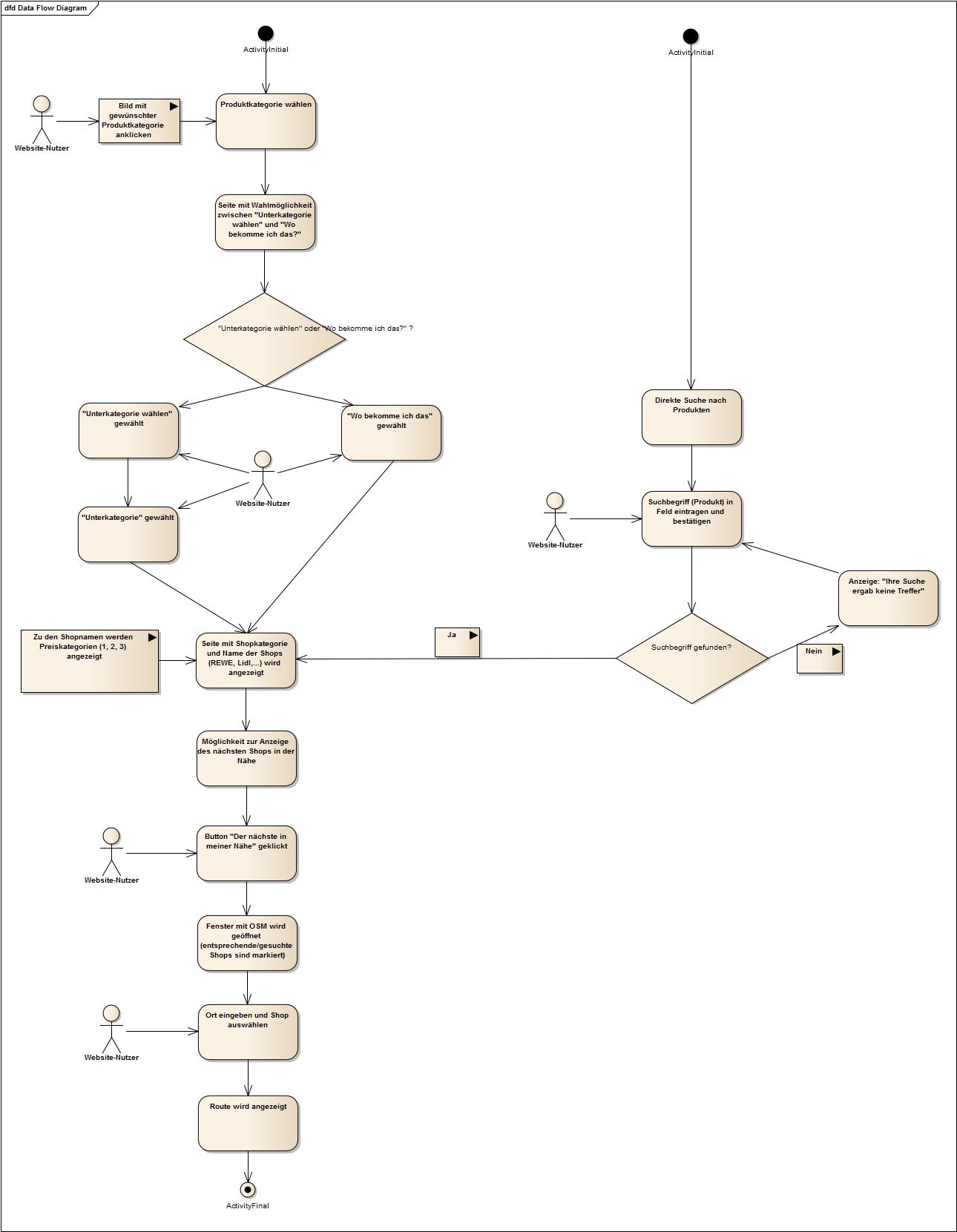
## Anwendungsfall Beschreibung

|  |  |
| --- | --- |
| Use Case | Wo gibt es was |
| Actors | Webseiten Besucher |
| Pre Conditions | * Die Webseite ist online * Der Datenbankserver ist online und gepflegt |
| Main flow | 1. Benutzer besucht die Webseite 2. Benutzer wählt „Wo finde ich was?“ in der Menuleiste aus 3. Der Benutzer wählt eine Produkthauptkategorie aus 4. Der Benutzer wählt eine Produktunterkategorie aus 5. Der Benutzer sieht Produktbeispiele der Produktunterkategorie und in welcher Shopkategorie er das Produkt kaufen kann.   Auf der angezeigten Seite wird darüber hinaus eine eingebunde Karte von Open Streetmap angezeigt. |
| Alternative flow (noch nicht implementiert) | 1. Benutzer besucht die Webseite 2. Benutzer wählt „Wo finde ich was?“ in der Menuleiste aus 3. Der Benutzer trägt in das Suchfeld einen Suchbegriff ein 4. Der Benutzer erhält alle gefundenen Ergebnisse übersichtlich angezeigt |
| Post conditions | * Der Benutzer muss auf die Ausangsseite zurückkehren können |
| end | Wo gibt es was |

## Anwendungsfalldiagramm

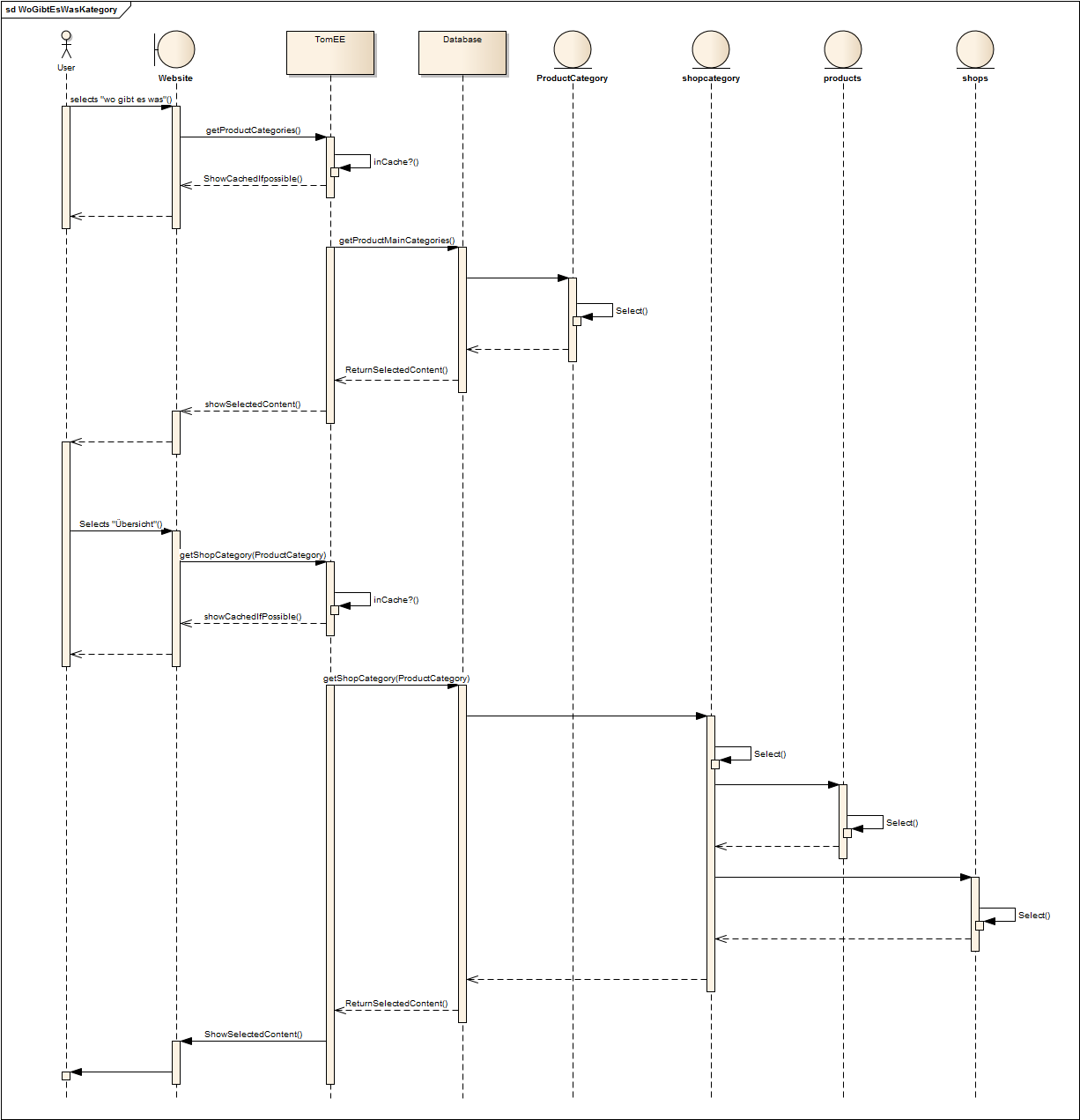
## 

## Aktivitätsdiagramm



## Klassendiagramm

## Sequenzdiagramm



## Zustandsdiagramm

## Auswertung der Diagramme

## Auswahl des Vorgehensmodells und Umsetzung

Im Projekt wurde sich gemeinschaftlich für Scrum entschieden. Eine ausführliche Beschreibung welche Vorteile wir in Scrum sehen und wie gut das funktioniert hat, entnehmen sie bitte Punkt 3.

## Auswahl der Architektur und Umsetzung

### Architecture

Das Softwaresystem basiert auf einer 3 Tier Architektur.

* Einen HTML Kompatiblen Webbrowser als Präsentationsschicht. Der Foundation Framework und die AngularJS werden eine Vereinfachung der Entwicklung auf Frontendseite herbeiführen.
* Als zweite Schicht, die Application Server Schicht wird ein “TomEE web profile” (basierend auf dem Apache Tomcat webserver/ servlet container, Java EE featuresets und OpenEJB
* Als dritte und letzte Schicht, die so genannte Datenschicht wird die MariaDB benutzt. Darüberhinaus wird diese Datenbank für einfache konsistenz und validierungs Checks und einfache authorisierungs Checks für den Zugriff auf die gespeicherten Daten genutzt.

### Security

The finished application will comply to the Deutsche Telekom Group [privacy and security assessment process](https://www.telekom.com/psa-process) and thus fulfill the security requirements available at [DE](https://www.telekom.com/static/-/155996/18/technische-sicherheitsanforderungen-si) [EN](https://www.telekom.com/static/-/246042/6/security-requirements-si).

### Allgemeines zu 3-Tier-Architektur

3-Tier-Architektur, oder auch Schichtenarchitektur genannt, ist ein Strukturierungsprinzip für die Architektur von Softwaresystemen. Den einzelnen Schichten (Layern) werden einzelne Aspekte des Softwaresystems konzeptionell zugeordnet, wobei zu beachten ist, dass Aspekte einer Schicht nur durch höhere Schichten verwendet werden können.

Vorteile:

Komplexität der Abhängigkeiten innerhalb des Systems werden reduziert und vereinfacht.

Zyklen im Abhängigkeitsgraphen werden vermieden (leichteres Verständnis und leichtere Wartung

einzelne Schichten können ausgetauscht werden ohne das gesamte Softwaresystem ändern zu müssen

Nachteile:

Ausführgeschwindigkeit der Anwendung kann aufgrund der Datenverarbeitung in den einzelnen Schichten verschlechtert werden

Die drei Schichten der 3-Tier-Architektur sind:

Präsentationsschicht (Front-End): Schnittstelle zwischen Benutzer und System, Datenein- und ausgabe

Logikschicht (Middle-Tier): Verarbeitungsmechanismen

Datenhaltungsschicht (Back-End): enthält die Datenbank und ist verantwortlich für das Laden und das Speichern der Daten

## Auswahl der Versionsverwaltung und Umsetzung

## Installationshandbuch

Die Installationshinweise finden sie vollständig als Dokument im Anhang. (Installationshandbuch.docx)

## Benutzerhandbuch

Die Benutzung der Webseite ist denkbar einfach. Wie auf jeder anderen Webseite auch, kann man mit linksklick einzelne Menüpunkte anwählen und somit auf diese Seite gelangen. Zurück gelangt man über den Browser. Mehr besondere Funktionen gibt es nicht – alles bringt ein herkömmlicher Internet Browser mit sich.  
Wir empfehlen dennoch den Chrome Browser von Google, da bestimmte Funktionalitäten (die Einbindung der Karte) bisher lediglich in diesem funktioniert.

# Anhang

<Hier einfügen: Benutzerhandbuch, Installationsanleitung,…>

RA-60 Dokumentation Sicherheitsarchitektur (PSA Verfahren der Telekom):



# Literaturverzeichnis

Klodinski, J. (kein Datum). Konzept 1.0 SoftwareEngineering.

1. Konzept1.0 SoftwareEngineering [↑](#footnote-ref-1)
2. Herr Karhof war bis zum Sprint 4 Teil des Teams. Jedoch stieg er aus dem Projekt aus, da er das Unternehmen verließ. [↑](#footnote-ref-4)
3. [Aufbauorgnisation g+](https://docs.google.com/spreadsheets/d/1W5Od9s6aarCx1sFPQoNwrKGn5htl_ddexCt8joMCvyU/edit?usp=sharing) [↑](#footnote-ref-5)
4. RA-17 [↑](#footnote-ref-6)
5. Meetingprotokoll vom 20.10.2015 [↑](#footnote-ref-7)
6. [Scrumboard](https://docs.google.com/spreadsheets/d/16N5eu0yM38uJgUU9UVnepE90Qff0_ujf9t0Hzrlt9jo/edit?usp=sharing) [↑](#footnote-ref-9)
7. Extreme Programming Präsentation von Sheldon Jesudasan & Nico Remus 28.10.2015 [↑](#footnote-ref-10)
8. (Jean-Pierre Klodzinski, 2015) [↑](#footnote-ref-11)