

Technische Hochschule Köln  
Fakultät 10  
-  
Institut für Ingenieurwissenschaften und Informatik

# Projektdokumentation

## „InTime“

**Gruppe 45**  
Stefan Geier, 11112826  
Vadim Demizki, 11112832

Studiengang Medieninformatik  
Entwicklungsprojekt interaktive Systeme  
WS17/18

**Dozenten**  
Prof. Dr. Gerhard Hartmann  
Prof. Dr. Kristian Fischer

**Betreuer**  
Robert Gabriel

---

**Inhaltsverzeichnis**

|   |           |
|---|-----------|
| <b>1. Exposé</b>  | <b>3</b>  |
| <b>2. Domänenrecherche</b>                              | <b>4</b>  |
| <b>3. Konkurrenzanalyse</b>                             | <b>5</b>  |
| 3.1 Konkurrenz identifizieren                           | 5         |
| 3.2 Eigenschaften des Projektes „InTime“                | 5         |
| 3.3 Google Now  | 6         |
| 3.4 Hound   | 6         |
| 3.5 Alexa (Amazon Echo)                                 | 7         |
| 3.6 Alleinstellungsmerkmale des Projektes „InTime“      | 8         |
| <b>4. Vorgehensmodell</b>                               | <b>9</b>  |
| <b>5. Projektplan</b>                                   | <b>11</b> |
| <b>6. Zielhierarchie</b>                                | <b>11</b> |
| 6.1 Strategische, taktische und operative Ziele         | 11        |
| <b>7. User Needs (Erfordernisse)</b>                    | <b>13</b> |
| <b>8. Recherche zu „predictive analytics“</b>           | <b>14</b> |
| 8.1 Was ist „predictive analytics“?                     | 14        |
| 8.2 „Predictive Analytics Process“                      | 14        |
| 8.3 API (Application Programming Interface)             | 15        |
| <b>9. API Analyse</b>                                   | <b>16</b> |
| 9.1 Was ist VRS ÖPNV?                                   | 16        |
| 9.2 OSM (Open Street Map)                               | 17        |
| <b>10. Stakeholder Analyse</b>                          | <b>18</b> |
| <b>11. Benutzermodelle</b>                              | <b>20</b> |
| 11.1 User Profiles                                      | 20        |
| 11.2 Personae   | 25        |
| <b>12. Benutzungsmodelle</b>                            | <b>33</b> |
| 12.1 Use Cases  | 33        |
| <b>13. Anforderungen</b>                                | <b>38</b> |
| 13.1 Funktionale Anforderungen                          | 38        |
| 13.2 Nicht-funktionale Anforderungen                    | 38        |
| 13.2.1 Produktqualität & Organisatorische Anforderungen | 38        |
| <b>14. Kommunikationsmodell</b>                         | <b>40</b> |
| 14.1 Deskriptives Kommunikationsmodell                  | 40        |
| 14.2 Präskriptives Kommunikationsmodell                 | 41        |
| <b>15. Architekturdiagramm</b>                          | <b>42</b> |
| <b>16. Risiken</b>                                      | <b>45</b> |
| <b>17. REST Modellierung</b>                            | <b>48</b> |
| <b>18. Proof of Concepts</b>                            | <b>49</b> |
| <b>19. UI</b>   | <b>51</b> |
| 19.1 Maße   | 51        |
| 19.2 Positionierung der Interaktionsobjekte             | 52        |
| 19.3 Farbgebung   | 52        |
| 19.4 Farbgebung der Interaktionsobjekte                 | 53        |
| <b>20. Verzeichnisse</b>                                | <b>55</b> |
| 20.1 Literaturverzeichnis                               | 55        |
| 20.2 Abbildungsverzeichnis                              | 57        |
| 20.3 Abkürzungsverzeichnis                              | 58        |
| 20.4 Tabellenverzeichnis                                | 59        |

## 1. Exposé

Im Zuge der Erarbeitung von Problemdomänen stießen wir auf die Domäne des Zeitmanagements. Nach weiterer Vertiefung entschieden wir uns für den Problemfall der Aktivitätsplanung mit darauffolgender Nutzung eines Busses als zeitliche Deadline.

Bis zu dem Zeitpunkt in dem ein Ablauf von Aktivitäten noch nicht zur Routine geworden ist, ist es für viele Menschen schwierig den Tagesabschnitt so zu planen, dass alle notwendigen Aktivitäten absolviert werden können und dennoch in passender Zeit die Wohnung verlässt um rechtzeitig an der Bushaltestelle zu stehen. Der momentan von vielen Leuten verfolgte Ansatz ist jener, dass alle paar Minuten auf die Uhr geschaut wird um abzuschätzen, ob man noch gut in der Zeit liegt oder sich beeilen sollte. Diese ständige Kontrolle möchten wir mit unserem System obsolet werden lassen.

Der Nutzer übergibt dem System die zu erledigenden Aufgaben und den Zielort, an dem er sich zu einem bestimmten Zeitpunkt zu befinden hat. In der Initialisierung des Systems wurde vom Nutzer bereits die durchschnittlich beanspruchte Zeit für oft vorkommende Tätigkeiten angegeben, sodass das System nun abschätzen kann wann der Nutzer die Aktivitäten voraussichtlich abgeschlossen hat. Daraufhin spricht der Dienstinutzer den Dienstgeber an, welcher von externen Quellen die nächste Bushaltestelle und die Laufdauer der Person dorthin ermittelt. Diese Daten werden schließlich an den Dienstinutzer überreicht, welcher dadurch der nutzenden Person mitteilen kann, wenn es in wenigen Minuten notwendig ist die Wohnung zu verlassen. Des Weiteren ist es möglich, dass das System nach jedem Zeitpunkt, in dem der Nutzer eine Aktivität abgeschlossen haben sollte, diesen darauf aufmerksam zu machen. Dadurch ist die nutzende Person auch während der Erledigung der Aufgaben in der Lage zu registrieren, ob es notwendig ist sich zu beeilen oder gar gewisse Aktivitäten zu verwerfen.

Heutzutage ist es immer häufiger notwendig zu einem gewissen Zeitpunkt an einem bestimmten Ort zu sein. Da es allerdings in den meisten Fällen erforderlich ist kurz davor explizite Aufgaben zu bearbeiten, gestaltet es sich oft schwierig den Termin dennoch rechtzeitig wahrzunehmen. Ob nun im Beruf, im Studium oder bei anderweitigen Terminen, diese Pünktlichkeit ist in vielen Aspekten des alltäglichen Lebens unentbehrlich, sodass sich nahezu jede Person vor dem genannten Problem regelmäßig wiederfindet.

Im Hinblick darauf, dass unser System eine mögliche Lösung des wiederkehrenden Problems bereitstellt und ein Großteil der Bevölkerung sich mit diesem zu befassen hat, lässt sich unserem Lösungsansatz dadurch eine gesellschaftliche Relevanz einräumen.

## 2. Domänenrecherche

Gemäß der von uns erarbeiteten Projektidee wird das System „InTime“ vorangehend in der Domäne des alltäglichen Zeitmanagements genutzt. Anders als Systeme dessen Domäne sich auf ein sehr spezielles Teilgebiet beschränkt, ist durch das sehr weite Nutzungsbereich nahezu jede Person ein potentieller Nutzer.

Aus dem Grund, dass im Hinblick des Zeitmanagements gewisse Aktivitäten in einem vorgegebenen zeitlichen Rahmen durchgeführt werden müssen, bildet das Konzept der strukturierten Ausführung von Handlungen eines der zwei wichtigsten Aspekt dieser Domäne. Das zweite bildet die möglichst optimale Nutzung des verfügbaren Zeitraums.

So ist es in Anbetracht der Gestaltung unseres Projektes notwendig, dass die vom Nutzer ausgehende Interaktion mit dem System während der Ausführung der im Zeitplan beschriebenen Aktivitäten möglichst wenig Zeit in Anspruch nimmt, sodass das System selbst kein Hindernis zur Einhaltung des Zeitplanes bildet. Des Weiteren ist es von hoher Priorität, eine strukturierte Planung der Aktivitäten zu ermöglichen, in der den einzelnen Handlungen ein verfügbarer Zeitraum eingeräumt wird, sodass die vorab gesetzte Deadline eingehalten wird und dennoch alle notwendigen Tätigkeiten ausgeführt werden konnten.

Ein nicht unerhebliches Paradigma des Zeitmanagements bildet zudem die, während der Abarbeitung des Zeitplans, möglich oft auftretende Einsparung von Zeit. Aufgrund dessen ist es stets anzustreben, dass der einer Aktivität vorgegebene Zeitrahmen unterschritten wird, sodass den darauffolgenden Tätigkeiten ein entsprechend vergrößerter Zeitraum zugewiesen werden kann.

Des Weiteren ist die Einplanung eines oder mehrerer zeitlichen Puffer ein Paradigma der Domäne. So ist es bei der Erstellung eines Zeitplans empfehlenswert den tatsächlich für eine Aktivität benötigten Zeitrahmen geringfügig zu vergrößern, sodass unberechenbaren externen Ereignissen, welche die Einhaltung des Zeitplans gefährden könnten, entgegengewirkt werden kann.

### 3. Konkurrenzanalyse

Mit der Erkenntnis über die Domäne in der sich das fertige Produkt befinden soll ist es von Nöten Konkurrenzprodukten aufzuspüren. Dieser Schritt hat eine sehr hohe Wichtigkeit für das ganze Projekt, da damit die Alleinstellungsmerkmale des eigenen Produkts gefunden werden können. Hinzukommend wird ersichtlich wie erfolgreich das Unternehmen mit dieser Geschäftsidee sein kann.

*„Die Konkurrenzanalyse ist ein systematischer Prozess der Beschaffung, Archivierung, Auswertung und Weitergabe von Informationen mit dem Ziel rechtzeitig Bedrohungen oder Chancen durch Wettbewerbsaktivitäten zu erkennen, um dann mit adäquaten Maßnahmen reagieren zu können.“ [1]*

Wie Peter Kairies im obigen Zitat bereits beschrieben hat, dreht sich diese Art von Analyse um das Sammeln von Informationen die dem eigenen Unternehmen ungemein helfen kann sich aus der Masse der Vielfalt von Produkten heraus zu kristallisieren.

#### 3.1 Konkurrenten identifizieren

Das Wichtigste ist einen Überblick über seine Konkurrenten zu haben. Um Konkurrenten schnell zu identifizieren muss das eigene Produkt erst analysiert werden. Erst durch die Vor- und Nachteile und der spezifischen Merkmale seines eigenen Produktes kann abgewogen werden welche Produkte der möglichen Konkurrenten überhaupt konkurrenzwürdig sind.

#### 3.2 Eigenschaften des Projektes „InTime“

Tabelle 1: InTime's Vor- und Nachteile

| Vorteile  | Nachteile  |
|---|--|
| <ul style="list-style-type: none"> <li>• Planen von Aktivitäten</li> <li>• Benachrichtigung bei Veränderung der Busabfahrtszeit</li> <li>• Geminderter Zeitdruck</li> <li>• Berechnung der Laufdauer von der aktuellen Position bis zur entsprechenden Bushaltestelle</li> <li>• <b>Berücksichtigt Aktivitäten bis zu dem ausgerechneten Zeitpunkt an dem das Haus verlassen muss und stellt einen Wecker zur Erinnerung.</b></li> <li>• Selbstständiges Lernen des Systems (möglich mit <b>Google Prediction API</b>)</li> </ul> | <ul style="list-style-type: none"> <li>• Keine Sprachfunktion</li> <li>• Begrenzung auf möglicherweise ein Endgerät</li> </ul> |

Dieses Produkt befindet sich in der Domäne des Zeitmanagements. Es ist ein Produkt zur Lösung des Problemfalls der Aktivitätsplanung mit darauffolgender Nutzung eines Busses als zeitliche Deadline. Eine besondere Funktion davon, ist das Auswählen von Aktivitäten, dass das System dann berücksichtigt und damit bestimmt wann der nächste Bus genommen werden soll. Diese Hybrid-App aus Wecker, Aktivitäten Planung und Nahverkehr App ist nicht nur sehr praktisch, sondern auch relativ einzigartig.

### 3.3 Google Now

Google Now ist eine Erweiterung des Google Search. Es verwendet sogenannte „Karten“ in denen spezielle Information abgespeichert werden. Diese Information kann um über verschiedene Themen wie z.B. Verkehr oder Politik handeln. In jeder dieser Karten sind personenbezogene Daten mit eingebunden (siehe auch [2]).

Darüber hinaus verfügt Google Now eine Sprachsuchfunktion, eine Ortssuchfunktion mit einer genauen Routenführung, eine Erinnerungsfunktion und die Funktion Informationen zu versenden bevor danach überhaupt gesucht wird (siehe auch [3]).

Tabelle 2: Google Now's Vor- und Nachteile

| Vorteile   | Nachteile  |
|--|--|
| <ul style="list-style-type: none"> <li>• Personen bezogene Datensammlung</li> <li>• Sprachfunktionen</li> <li>• Ortsuchfunktion mit Routenführung</li> <li>• Erinnerungsfunktion</li> <li>• <b>Information erhalten ohne Suchanfrage zu starten (Google Prediction API)</b></li> <li>• Auf allen Endgeräten</li> </ul> | <ul style="list-style-type: none"> <li>• Standort immer aktiv</li> <li>• Sammeln von persönlichen Daten</li> <li>• Akkuverbrauch bei mobilen Endgeräten</li> </ul> |

### 3.4 Hound

Hound ist wie Siri, Alexa oder Google Now mit einer Sprachsuchfunktion ausgestattet. Um die Konkurrenten auszusteichen kann Hound viel komplexere Suchanfragen in wenigen Sekunden bewältigen. Durch so eine Funktion können Benutzer viele Kriterien und Filterfunktionen in ihrer Hauptanfrage einbringen. Beispielsweise kann eine Anfrage an das System gestellt werden die wie folgt lautet: „*Suche mir eine Pizzeria in meiner Nähe*“. Siri und Co hätten auch bei dieser Anfrage kein Problem, diese schnell und akkurat zu beantworten. Wenn jetzt aber weitere Suchkriterien dazu genannt werden, wie zum Beispiel: „*Dieses Lokal soll auch Fritten auf der Speisekarte haben und bis 23 Uhr geöffnet sein*“, wird der Vorteil gegenüber den Rivalen ersichtlich (siehe auch [4]).

Diesen hohen Grad des sprachlichen Verständnisses verdankt Hound der hybriden Sprachfunktion, bestehend aus Sprachenerkennung und Sprachverstehen. Mit so einer Errungenschaft kann auch Umgangssprache gut identifiziert und richtig vom System gedeutet werden. Das einzige Hindernis für diese App im deutschen Raum ist die aktuelle Sprachunterstützung. Denn Hound ist zurzeit lediglich in englischer Sprachausgabe verfügbar (siehe auch [5]).

Tabelle 3: Hound's Vor- und Nachteile

| Vorteile  | Nachteile   |
|---|---|
| <ul style="list-style-type: none"> <li>• Sprachfunktion</li> <li>• Ortssuchfunktion mit Routenführung</li> <li>• Anruf tätigen</li> <li>• Textnachrichten senden</li> <li>• Informationen prüfen etc.</li> <li>• Interaktive Spiele mit Sprachsteuerung</li> <li>• <b>Spracherkennung mit Sprachverstehen (Speech-to-Meaning technology)</b></li> </ul> | <ul style="list-style-type: none"> <li>• Nur auf Englisch.</li> <li>• Keine Informationen zu lokalen Busverbindungen und Bushaltestellen</li> <li>• Zeitplanungsfeatures nur im niedrigem bis mittelmäßigem Maße vorhanden</li> </ul> |

### 3.5 Alexa (Amazon Echo)

Auch bei Alexa ist die Steuerung sehr sprachbasierend. Es ist wie Siri ein Sprach-Assistent mit sehr ähnlichen Funktionen. Alexas Hauptfunktionen sind das Managen der Musikkwiedergabe und der außergewöhnlichen Funktion der Einkaufsliste, die erstellt werden kann, wenn das Amazon-Premium Konto mit Alexa verbunden wurde (siehe auch [6]).

Hinzu kommt, dass sich mit Alexa auch das eigene Haus steuern lässt, falls die einzelnen Komponenten (Lampen, Fernseher, etc.) auch kompatibel sind. Es sind auch Anrufe und das Versenden von Nachrichten möglich. Aber auch viele weitere Funktionen die mit der Informationsbeschaffung, wie z.B. das Abfragen der Wetter- oder Verkehrsdaten, liegen im Rahmen des Möglichen (siehe auch [7]).

Tabelle 4: Alexa's Vor- und Nachteile

| Vorteile  | Nachteile   |
|---|---|
| <ul style="list-style-type: none"> <li>• Sprachfunktion</li> <li>• <b>Smart Home Geräte steuern</b></li> <li>• <b>Einkaufsliste erstellen und direkt bestellen.</b></li> <li>• Einfache Informationsbeschaffung.</li> </ul> | <ul style="list-style-type: none"> <li>• Preis</li> <li>• Raumgebunden</li> <li>• Nur eine einfache To-Do Liste die sich erstellen lässt.</li> <li>• Stetige Tonaufnahme über das Mikrofon, da</li> </ul> |

|   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Starke und viele Mikrofone zur eindeutigen Spracheingabe</li> <li>• Musikwiedergabe etc.</li> <li>• Taxi bestellen.</li> </ul> | durchgehend auf Kommandos gewartet werden muss (fraglicher Datenschutz). |
|---|--|

Nach dem Identifizieren der größten Gegner hinsichtlich unserem zukünftigem Produkts folgt das Nebeneinanderstellen der Nachteile der Konkurrenz. Durch diese war zu erkennen, dass die Nachteile in nur geringer Anzahl vorhanden waren. Auf Basis der bisher gefundenen Nachteile, begannen wir die Entwicklung des Alleinstellungsmerkmals.

Einerseits wird nur gutes von dem jeweiligen Produkt gesprochen, da kein Unternehmen sein Produkt in irgendeiner Weise schlecht darstellen würde andererseits wurde klar, dass einige Vorteile unseres Produkts Defizite bei der Konkurrenz aufweisen. Diese Funktionalitäten waren im Grunde vorhanden aber nicht so spezifisch entworfen um das Ziel unseres Produktes erreichen zu können. Meist kann mit den Konkurrenz Erzeugnissen nur eine einfache To-Do Liste erstellt werden, die mittels einer Benachrichtigung einen daran erinnern. Das kommt einer kontrollierten Aktivitäten Planung zwar etwas näher aber erfüllt es nicht in vollen Zügen. Daraus folgt ein eindeutiges Alleinstellungsmerkmal.

### 3.6 Alleinstellungsmerkmale des Projektes „InTime“

InTime's Alleinstellungsmerkmal ist die Tatsache, dass die Applikation einen weckt, anhand der Aktivitäten die morgens geplant sind und der damit verbundenen Zeit die benötigt wird, um anschließend den Bus zu erwischen.

Als eine Erweiterung dieser Funktionen können sich auch mehrere Accounts verbinden die im selben Haushalt Tätigkeiten ausüben wollen. Hier regelt dann das System, zum Beispiel die Reihenfolge der Benutzung der Dusche. Hinzukommend wäre da noch die Funktion des Vorhersagens vom System nach einer gewissen „Zeitspanne des Lernens“. Somit wäre es möglich die Bewältigungszeit der Aktivitäten dem System zu überlassen.

InTime's Alleinstellungsmerkmal setzt sich gewissermaßen aus 2 Unterpunkten zusammen. Eines dieser Punkte ist die Weckfunktion, welche anhand der Busdaten und der zuvor geplanten Aktivitäten einen optimalen Zeitpunkt des Weckens bestimmt. Das zweite, aber nicht minder wichtige Merkmal ist die mehrgliedrige Planung von Tätigkeiten, in der die zeitlich bestmögliche Ausführung eines gesamten Aktivitätenblocks ermittelt wird.



---

## 4. Vorgehensmodell

Nach Erarbeitung der Projektidee lag es in unserem Interesse, ein für unser Vorhaben angepasstes Vorgehensmodell zu finden, beziehungsweise zu erstellen. Da dieses den groben Erarbeitungsprozess des kommenden Projektes skizziert und aus ihm zusätzlich der Projektplan abgeleitet wird, lässt sich der Erarbeitung eines Verhaltensmodells ein hoher Stellenwert einräumen. Aus diesem Grund ist es ratsam das Vorgehensmodell zu Anfang des Projektes zu erarbeiten.

Im Zuge der eingehenden Recherche konventioneller Verhaltensmodelle ließ sich in Anbetracht der in der Veranstaltung „Entwicklung interaktiver Systeme“ gesetzten Rahmenbedingungen eine Mehrzahl dessen ausgliedern. So ist es notwendig zu Beginn der Projektarbeit formale und dokumentierende Arbeitsschritte durchzuführen, während beispielsweise die Methode des *Extreme Programming* diese nicht vorsieht und stattdessen den Ansatz der agilen Programmierung verfolgt. Aus demselben Grund waren wir in der Lage zusätzlich jegliche weiteren Modelle auszuschließen, welche den Ansatz der agilen Programmierung verfolgen.

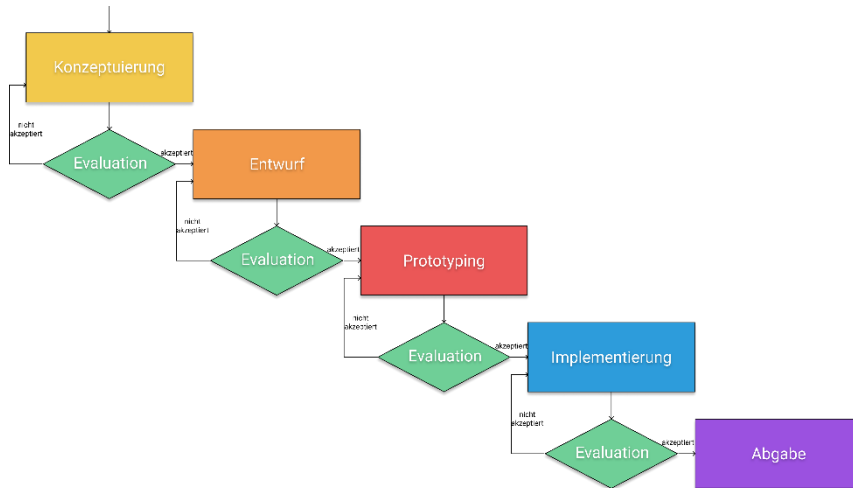
Als Alternativen boten sich stattdessen das V-Modell und das Spiralmodell an. Letzteres sieht eine vor der Implementierung Anfertigung von präskriptiven Artefakten vor, welches damit nicht den zuvor im Ansatz des *Extreme Programming* genannten Kritikpunkt aufweist. Allerdings beschreibt das Spiralmodell eine auf Zyklen basierte Vorgehensmethode, welche wiederum vorsieht, dass die verschiedenen Unterpunkte der Artefakte erarbeitet und, nach Erfassung der Risiken, direkt im Anschluss implementiert werden. Dies hat zur Folge, dass ein pseudoparalleler Entwicklungsprozess erreicht wird, welches jedoch nicht den vorgegebenen Rahmenbedingungen entspricht.

Das V-Modell wiederum birgt weder das Problem der fehlenden Artefakterarbeitung noch der zyklischen Anfertigung dieser. Stattdessen wird hier der Planung und der Vorarbeit ein gewisser zeitlicher Rahmen gegeben, indem eben diese Erarbeitet werden können. Begonnen wird dabei mit den abstraktesten Artefakten, während mit der Zeit immer detailliertere Bearbeitungen stattfinden. Dies gipfelt schließlich in der Codeimplementation, welches den detailliertesten Arbeitsschritt bildet und im zeitlichen Mittelpunkt der Projektbearbeitung anzusetzen ist. Darauffolgend wird das Erarbeitete System anhand der zuvor entworfenen Artefakte in Form von Test überprüft.

Auch wenn dieses Vorgehensmodell sehr attraktiv ist um die zuvor gesetzten Ziele korrekt und vollständig zu erreichen, so entschieden wir uns aus dem Grund dagegen, da wir sowohl zeitlich als auch im Hinblick auf nötige Ressourcen nicht in der Lage sind, die in dem V-Modell vorgesehenen Tests durchzuführen.

Zuletzt stießen wir in Anbetracht der Vorgabe, eine gewisse Struktur in der Planung beizubehalten, auf das klassische und weit verbreitete Wasserfallmodell. Dieses trennt die größten Bearbeitungsschritte des Projektes klar voneinander und setzt zudem eine klare aufeinanderfolgende Abarbeitung dieser Schritte voraus.

Eben jener Aspekt des Wasserfallmodells entspricht somit bereits in gewissem Maße der Meilensteine des Moduls, in dessen Rahmen das Projekt bearbeitet wird. Nichtsdestotrotz war eine leichte, jedoch gleichzeitig



gehaltvolle Modifikation, in Form von abschließender Evaluation der Arbeitsschritte, von Nöten (siehe Abbildung 1). So sind wir in der Lage, die zu Anfang notwendigen Artefakte zu formulieren um nach darauffolgender

Abbildung 1. Das für unser Vorhaben angepasste Wasserfallmodell

Evaluation in den nächsten Arbeitsschritt zu übergehen und zudem parallel den Abgabeplan des Moduls zu folgen. Um jedoch auch während der Entwicklung einer UI und des Programmcodes ein strukturiertes Vorgehen zu gewährleisten, entschieden wir uns auch für diese Arbeitsschritte entsprechende Modelle zu wählen. Für die Gestaltung der Nutzungsoberfläche wird von uns das Vorgehen des in ISO 9241-210 beschriebenen Mensch-zentrierten Designs genutzt. Sowohl die dort vorgesehene modulare Abarbeitung aller notwendigen Nutzungsanforderungen, als auch die regelmäßige Evaluation der entwickelten Prototypen unterstützt uns im Bearbeitungszeitraum durch einen strukturierten und sukzessiven Arbeitsablauf.

Zuletzt ist an dieser Stelle noch das Vorgehen innerhalb der Programmierung zu nennen. Anders als in den voran gegangenen Punkten verwenden wir an dieser Stelle kein Verhaltensmodell, sondern die Methode des Feature Driven Developments.

Da wir zu dem Zeitpunkt der letztendlichen Programmierung bereits eine Menge an Anforderungen vorliegen haben ist es uns möglich anhand des FDDs diese nach und nach zu implementieren. Hierbei ist jedoch zu erwähnen, dass wir aufgrund der Gruppengröße nicht auf einen zu besetzenden Posten zur Überblicksgewährleistung zurückgreifen. Stattdessen erfolgt dieser Querschnittseinblick in die Programmierphase durch regelmäßige, kurze Evaluation der Erarbeitung und möglichst optimale Dokumentation der Vorgänge und noch zu programmierenden Inhalts.

## 5. Projektplan

Auf Basis des Vorgehensmodells, der die einzelnen Entwicklungsphasen berücksichtigt, lassen sich Entwicklungsschritte ableiten und über die Projektdauer eine übersichtliche Darstellung des Verlaufs erstellen. Der Projektplan bezieht sich auf alle Entwicklungsschritte von der Projektidee bis hin zur Implementierung. Im Projektplan werden alle Aktivitäten, Nebenaktivitäten und Meilensteinen aufgelistet. Hinzukommend wird die Dauer der Bearbeitung der Aktivitäten festgehalten. Der Plan dient in erster Linie der Organisation der Bearbeitungsreihenfolge und bildet den Leitfaden der Entwicklung des Projektes. Da das Team von „InTime“ aus zwei Personen besteht ist eine Zeitanforderung von 600 Stunden zu erfüllen.

Der Projektplan ist unter [diesem Link](#) zu finden [16].

## 6. Zielhierarchie

Bevor es in die Entwicklung geht ist es wichtig die Ziele festzulegen, die das System am Ende erfüllen soll. Ohne ein konkretes Ziel vor Augen zu haben ist es schwer darauf hin zu arbeiten. Die Zielhierarchie hilft mit den strategischen, taktischen und operativen Zielen dabei auf verschiedenen Ebenen des Unternehmens Ziele zu definieren. Strategische Ziele sind globale und langfristige Ziele des Unternehmens für das Produkt, die in der Formulierung sehr grob und einfach gehalten werden. Taktische Ziele sind mittelfristig und beziehen sich auf die Ziele einer Gruppe. Die operativen Ziele hingegen schließen meist nur die Ziele einer Person mit ein und werden Detailreich beschrieben.

### 6.1 Strategische, Taktische und Operative Ziele

Tabelle 5: Zielhierarchie Teil 1

| Strategisches Ziel   | Taktisches Ziel                         | Operatives Ziel   |
|--|---|---|
| Das System soll eine große Anzahl von Aktivitäten sammeln. | Aktivitäten sollten gespeichert werden. | 1. Benutzer speichert verschiedene Aktivitäten, um diese für sich immer wieder zu nutzen. |
|  |   | 2. Andere Nutzer können diese Aktivitäten auch für sich auswählen.                        |

Tabelle 6: Zielhierarchie Teil 2

| Strategisches Ziel   | Taktisches Ziel  | Operatives Ziel   |
|--|--|---|
| Die Zeit für jede Aktivität des Benutzers zu einer sehr Hohen Wahrscheinlichkeit vorhersagen können. | Benutzer soll mit Prädiktive Analytik vorhersehbar werden. | Durch eine Probelauf speichert das System die Zeit von den Aktivitäten und nutzt diese für spätere vorhersagen. |

Tabelle 7: Zielhierarchie Teil 3

| Strategisches Ziel        | Taktisches Ziel                   | Operatives Ziel   |
|---------------------------|-----------------------------------|---|
| Benutzerdaten beschützen. | Verhindern von Verlust von Daten. | Angriffen von außen durch sicheren Transportkanal schützen. |

Tabelle 8: Zielhierarchie Teil 4

| Strategisches Ziel                             | Taktisches Ziel  | Operatives Ziel  |
|--|--|--|
| Der Benutzer kommt immer pünktlich am Ziel an. | 1. Das System muss schnell auf Veränderungen wie Fahrplanänderung reagieren.                               | 1.1. Anpassungen an den Aktivitäten und/oder der Weckzeit. |
|  | 2. Veränderte Aktivitätendauer darf sich nicht großartig auf den weiteren Verlauf des Zeitplans auswirken. | 2.1. Anpassen der anderen Aktivitäten.                     |
|  |  | 2.2. Entfernen von wenig priorisierten Aktivitäten.        |
|  |  |  |

## 7. User Needs (Erfordernisse)

Als entscheidender Schritt der die Richtung der Entwicklung maßgeblich beeinflusst, ist die der Benennung der Erfordernisse. Dafür ist es erforderlich die Nutzungssituation richtig einzuschätzen, verstehen und zu erkennen, was ein Benutzer in einer bestimmten Situation braucht.

### Ablauf:

#### 1. Voreinstellungen

- Der Benutzer möchte seine Aktivitäten mit der entsprechenden Zeit die er dafür benötigt in eine App eintragen.

#### 2. Aufstehen

- Benutzer möchte in irgendeiner Weise geweckt oder benachrichtigt werden.

#### 3. Aktivitäten (Vor der Deadline)

- Der Benutzer will ohne viele Apps zu bedienen wissen:
  - Wann der nächste Bus kommt.
  - Wie lange er noch hat bis der Bus kommt.
  - Eine Benachrichtigung bekommen falls der Bus eine andere Abfahrtszeit hat.

Ein Erfordernis beschreibt also:

- **WAS** in einer konkreten Situation
- für einen **BESTIMMTEN BENUTZER**,
- **VORHANDEN** bzw. **ERFÜLLT** sein muss,
- Damit ein Mensch ein bestimmtes **ZIEL** erreichen kann.

### User Needs

Als <Nutzer> muss man <die Abfahrtzeit> <wissen>, um <den Bus> <bekommen> zu können.

Als <Nutzer> muss man <die verfügbare Zeit> <wissen>, um <Aktivitäten vor der Deadline><in Ruhe ausführen> zu können.

Als <Nutzer> muss man <veränderte Abfahrzeiten> <wissen>, um <den Bus><nicht verpassen> zu können.

Als <Nutzer> muss man <ein Interaktionsobjekt> <verfügbar haben>, um <die Aktivitätsbearbeitungsdauer><anpassen> zu können.

---

## 8. Recherche Predictive Analytics

Für die Zielerreichung sind neue und unbekannte Technologien sehr wichtig. Diese lassen sich meist durch eine gezielte Recherche erkunden und liefern Erkenntnisse über neue Möglichkeiten der Entwicklung.

### 8.1 Was ist „Predictive Analytics“?

Bei der prädiktiven Analytik geht darum eine Statistik z.B. über das Verhalten einer Person zu erstellen. Die einzelnen Informationen werden dann aus den Datenmengen extrahiert und verglichen, um Zusammenhänge in diesen Datenmengen zu entdecken und eine Vorhersage des Verhaltensmusters der ausgewählten Person zu treffen. Meist sind es wiederkehrende Entscheidungen, die epochal sind bei der Prognose der Handlungsweise eines Nutzers.

Prädiktive Analytik ist eine statistische und analytische Technik um ein Modell zu entwickeln, dass die zukünftigen Ereignisse und Handlungsweisen darstellt. Das Modell unterscheidet sich stark nach welchen Vorhersagen sich gerichtet wird (aus [8]).

Das wichtigste bei dieser und wie bei vielen anderen analytischen Techniken ist es zuverlässige Daten mit hoher Qualität zu haben, um eine relative saubere Analyse durchzuführen. „Predictive Analytics“ ist in vielen Branchen gegenwärtig sehr ausgeprägt vorhanden. Von dem Vorhersehen von Flugpreisen, Kundenkaufverhaltens oder das Aufstellen von Prognosen hinsichtlich der Behandlungserfolge an Patienten gibt es ein breites Spektrum der Einsatzmöglichkeit dieses Verfahrens. Jedoch gibt es einige Bedenken bei der Ausführungsgewalt der „Predictive-Analytics-Software“. Falls die Software so entworfen ist, dass der Endbenutzer falsche Parameter eingeben kann, so wird es zur einer falschen Vorhersage kommen und einen gravierenden Schaden beim Unternehmen verursachen. Um dem entgegenzuwirken ist es unabdinglich die Software so zu gestalten, dass es keine falschen Parameter geben kann (aus [9]).

### 8.2 „Predictive Analytics Process“

Bei dem Ablauf der prädiktiven Analytik gibt es eine feste Reihenfolge, die jedes Mal abgearbeitet werden muss um eine vollwertige Analyse zu bekommen.

Der erste Schritt bezieht sich auf das „Bestimmen des Projekts“. Hier wird, wie der Name schon erahnen lässt das Projekt definiert und es werden Datensätze mit denen man arbeiten wird identifiziert und bereitgelegt. Der zweite Schritt ist dafür gedacht die Daten zu sammeln. Ein sehr wichtiger Begriff ist das „Data Mining“ was sich auf die große Daten Menge bezieht und durch Methoden diese verarbeitet. Bei diesem Schritt ist die Benutzerinteraktion unabdingbar mit einbezogen, wodurch sich ein Einblick in diese Interaktion ermöglicht wird.

Im dritten Schritt wird die eigentliche Datenanalyse durchgeführt. Durch die Inspektion von Daten werden wichtig eingestuft Informationen aufgefunden.

Beim vierten Schritt wird eine Statistik anhand der vorgefundenen Daten erstellt.

Die Statistik dient als fünfter Schritt dann als ausschlaggebendes Instrument zur Modellierung der Vorhersagungsmodelle.

Der vorletzte Schritt ist die Entwicklung. Dieser Abschnitt widmet sich völlig der Entwicklung der Vorhersagungsmodelle. Durch Entscheidungen und Verhaltensmuster die im Alltag getroffen werden, können diese als großes Ganzes dafür genutzt werden, um die Modellentwicklung zu automatisieren. Die Modellüberwachung ist der siebte Schritt und bildet das Schlusslicht dieses Prozesses. Am Ende wird das Modell wie so gut wie jedes verwaltet und überwacht. Diese gilt der Überprüfung und beugt mögliche Fehler vor (siehe auch [10]).

### 8.3 API (Application Programming Interface)

Google selbst bietet die Google Cloud Prediction API an, die vor allem Funktionen der „Predictive Analytics“ ermöglicht. Diese RESTful-API ermöglicht es die Erstellung eines Modells für maschinelles Lernen und lässt damit zu Daten zu analysieren und verschiedene Funktionen zu Ihren Anwendungen hinzuzufügen. Leider ist die API veraltet und wird ab dem 30.04.2018 außer Dienst gestellt.

#### Welche Aufgaben erfüllt die Prediction API?

Die API führt laut dem offiziellen Entwicklerleitfaden der „Cloud Prediction API“ zwei Aufgaben durch:

1. *„Bei einem neuen Element wird ein numerischer Wert vorhergesagt, der für dieses Element gilt. Dieser Wert basiert auf ähnlich bewerteten Beispielen in den Trainingsdaten.“ [11]*
2. *„Bei einem neuen Element wird eine Kategorie gewählt, die es am besten beschreibt. Voraussetzung dafür ist ein Satz von ähnlich kategorisierten Elementen in den Trainingsdaten.“ [11]*

Der Nachfolge von Google Cloud Prediction API ist „Cloud Machine Learning Engine“. Wie Cloud Prediction erfüllt es alle gewünschten Funktionen und verfügt über das leistungsstarke TensorFlow-Framework, das hinter vielen Produkten von Google steht. TensorFlow selbst, ist eine Open Source Software-Bibliothek für numerische Berechnungen mit Hilfe von Datenflussgraphen.

Besonderheiten von „Cloud Machine Learning Engine“ (aus [12]):

1. Einfache Modelle für maschinelles Lernen anfertigen, die mit allen Datentypen und -größen arbeiten können.
2. TensorFlow-Framework (Deep Learning Funktionen).
3. Skalierbare prognostische Analysen.

---

## 9. API Analyse

Mit Hilfe der Ergebnisse der „user needs“ kann man die Funktionalitäten und damit verbundenen Systemschnittstellen die das System braucht errahnen. Für die notwendigen Details hilft eine Analyse in diese Richtung.

### 9.1 Was ist VRS ÖPNV?

Die VRS ÖPNV ist eine API welche Fahrplaninformationen beherbergt, die das Gebiet Rhein-Sieg umfassen. Durch externe VRS Schnittstellen (Open Service) wie dem ASS (Auskunft Service System) lassen sich Fahrplandaten, Haltestellen und sonstige für den Nahverkehr wichtigen Daten abfragen. Diese Funktion ist aber an eine maximale Anfragefrequenz pro Tag gebunden.

3 Schritte sind notwendig um diese Daten zu nutzen (aus [13]).

1. Die Nutzervereinbarung wird unterzeichnet und mit einer kurzen Projektbeschreibung (max. 500 Zeichen) an **api@vrsinfo.de** gesendet.

Diese Vereinbarung beinhaltet:

- Dass bei jeder präsentierten Ergebnisseite die Nennung der Datenquelle sichtbar platziert werden muss.
- Die Datensicherheitsstandards müssen eingehalten werden.
- Es muss verhindert werden das DDos Angriffe auf die API möglich sind.
- Und es dürfen Daten nicht an Dritte weitergeleitet werden.

2. Als nächstes wird das Ganze von der VRS GmbH geprüft.
3. Falls alles in Ordnung ist, wird ein Passwort für die Freischaltung des Zugriffs verschickt.

Mit dem Zugriff auf die API wird einem auch eine Dokumentation zugesendet, die Funktionalitäten und Anwendungsbeispiele für die ASS Schnittstelle enthält. Der ASS Server stellt eine XML-Schnittstelle bereit, die über eine URL angesprochen werden kann. Dieser Server stellt außer der Fahrplanauskunft oder Haltestellenabfahrtsplan auch viele andere Dienste zu Verfügung.

Diese sind:

- Die Objektsuche: Die Objektsuche beschreibt die allgemeine Suche von geografischen Objekten.
- Die Adresssuche: Die Adresssuche beschreibt die spezielle Suche nach Adressen und deren geografische Lage.
- Die POI-Suche: Die POI-Suche beinhaltet die Suche nach POI's (Point of Interest) und deren geografische Lage. Hierbei werden alle relevanten, bzw. verfügbaren Attribute ausgegeben.
- Die Kurzstrecken.



## Verwendung der API

Die Anfragen die vom Client zum Server werden über das http übertragen. Diese Informationen werden mittels der POST Methode zwischen den beiden kommunizierenden Komponenten ausgetauscht. Um eine erfolgreiche Anfrage und deren Beantwortung sicher zu stellen, stellt der ASS-Server URLs zur Verfügung, die XML Anfragen bearbeiten. Als XML Response werden vorbereitete Daten geordnet zurückgesendet.

### 9.2 OSM (Open Street Map)

Open Street Map ist eine öffentlich zugängliche API die sich unter der Berücksichtigung der Namensnennung kostenlos nutzen lässt. Die API stellt geografische Karten der Welt zu Verfügung wodurch Routen es leicht ist anhand der Position eine Route zum Ziel zu bestimmen. Leider ist OSM nicht dauerhaft zuverlässig, da die Karten sich nicht immer auf dem aktuellsten Stand befinden. Durch diesen möglichen Mangel sollte beim Einsatz der API berücksichtigt werden, dass daraus ein möglicher Personenschaden entstehen könnte (aus [14]).

Die API basiert auf dem Konzept der RESTful API. Wie in allen anderen RESTful APIs werden die http Verben (GET, PUT, POST, UPDATE und DELETE) verwendet um auf Ressourcen mit unterschiedlichen Anfragemethoden zugreifen zu können. Als Antwort wird eine XML Datei zurück geliefert (text/xml) (aus [15]).

## 10. Stakeholder Analyse

Auf die Phase der Untersuchung über die Domäne, der Erfordernisse des Benutzers und die Erfassung der Systemschnittstellen folgt die Phase der Identifizierung von direkten oder indirekten Personengruppen, die auf irgendeiner Art und Weise in Beziehung zu dem Produkt stehen.

### Was sind Stakeholder?

Laut der ISO-EN-DIN 9241, Teil 210 ist ein Stakeholder ein Individuum(Einzelperson) oder Organisation, die ein Anrecht, einen Anteil, einen Anspruch oder ein Interesse auf ein bzw. an einem System oder an dessen Merkmalen hat, die ihren Erfordernissen und Erwartungen entsprechen. Nach Berücksichtigung der User Needs erstellen wir eine Liste von Stakeholder mit ihren Prioritäten zum Produkt.

| Bezeichnung        | Beziehung | Objektbereich  | Erfordernis/Erwartung                            | Priorität (Gesamtsicht) | Priorität (Programmierersicht) | Priorität (Designersicht) |
|--------------------|-----------|--|--|-------------------------|--------------------------------|---------------------------|
| Nutzer             | Anspruch  | Korrekte Berechnung der notwendigen Zeit für Tätigkeiten | Erreichung des Projektziels/Lösung des           | 10                      | 10                             |                           |
|                    | Interesse | Intuitive Nutzung  | Anstrengungsfreie Nutzung                        | 7                       | 2                              | 10                        |
|                    | Anrecht   | Datenschutz  | Rechtsvorschrift                                 | 10                      | 10                             | 10                        |
|                    | Anspruch  | Problemfreie Nutzung des Systems (bug-/crashfrei)        | Möglichst Allzeit Nutzbar                        | 6                       | 8                              |                           |
|                    | Anspruch  | Erinnerung bei kritischen Zeitpunkten                    | Hilfestellung/Führung durch den Zeitplan         | 9                       | 9                              |                           |
|                    | Anspruch  | Geringe Nutzungszeit während in Ausführung des Zeitplans | Zeitersparnis                                    | 7                       | 6                              | 9                         |
|                    | Anspruch  | Verspätungen der Busse berücksichtigen                   | Durchgängige Korrektheit des Systems             | 8                       | 8                              |                           |
|                    | Interesse | Ansprechendes Aussehen des UI                            | Möglichst Intuitive Nutzung des Systems          | 8                       |                                | 9                         |
| Entwickler         | Interesse | Problemfreie Nutzung des Systems während der Laufzeit    | Fokus auf andere Aspekte der Entwicklung möglich | 5                       | 8                              |                           |
|                    | Interesse | Möglichst wenig Nachkallibrierung im Code notwendig      | Fokus auf andere Aspekte der Entwicklung möglich | 5                       | 8                              |                           |
|                    | Interesse | Erfolg des Systems                                       | Berufserfolg/Portfolio aufbauen                  | 7                       | 9                              | 9                         |
|                    | Anspruch  | System/Code ist Erweiterbar                              | Arbeitseinsparung, sobald Erweiterung laut       | 7                       | 8                              |                           |
|                    | Anspruch  | Korrekte Planung durch Management                        | Erfolg des Systems                               | 8                       | 7                              | 7                         |
| Manager/Management | Interesse | Möglichst große Nutzermenge                              | Durchsetzung gegenüber anderen Unternehmen       | 8                       | 4                              | 4                         |
|                    | Interesse | Hohe Nutzerzufriedenheit                                 | Kundenkreis aufbauen/Unternehmen                 | 9                       | 6                              | 8                         |
|                    | Anspruch  | Der Planung entsprechende Entwicklung                    | Möglichst Reibungsloser Bearbeitungsablauf       | 9                       | 8                              | 8                         |
|                    | Interesse | Erreichung des Projektziels                              | Berufserfolg/Portfolio aufbauen                  | 10                      |                                |                           |
|                    | Anspruch  | System ist Erweiterbar                                   | Spätere Anpassungen/ Rekalibrierung möglich      | 5                       | 8                              |                           |
|                    | Anspruch  | Geldverdienst  | Entwicklungskosten tragen können                 | 10                      | 8                              | 8                         |

Abbildung 2: Stakeholder Teil 1

|               |                  |   |   |    |    |    |
|---------------|------------------|---|---|----|----|----|
| Gesetzgeber   | Anrecht/Anspruch | Konformität des Systems mit Richtlinien und Gesetzen            | Rechtsvorschrift                              | 10 | 10 | 10 |
|               | Anrecht/Anspruch | Datenschutz der Nutzer (per se)                                 | Rechtsvorschrift                              | 10 | 10 |    |
|               | Anrecht/Anspruch | Einverständnis zur Benutzerspezifischen                         | Rechtsvorschrift                              | 10 | 10 |    |
| OpenStreetMap | Anrecht/Anspruch | Legale Nutzung des eigenen Teilsystems                          | Rechtsvorschrift                              | 10 | 10 | 5  |
|               | Anspruch         | Nennung von OpenStreetMap im System                             | Nutzungsbedingungen                           | 10 | 8  | 10 |
|               | Anrecht/Anspruch | Vorgesehene Nutzung von OpenStreetMap (kein Missbrauch)         | Nutzungsbedingungen                           | 10 | 10 | 5  |
| VRS: ÖPNV     | Anrecht/Anspruch | Legale Verwendung von ÖPNV                                      | Rechtsvorschrift/<br>Nutzungsvereinbarung     | 10 | 10 |    |
|               | Anrecht          | Vorgesehene Nutzung des Dienstes gemäß der Nutzungsvereinbarung | Rechtsvorschrift                              | 10 | 10 |    |
|               | Anrecht          | Korrekte Angabe der Nutzung des ÖPNV-Dienstes auf jeder         | Nutzungsvereinbarung                          | 10 | 6  | 10 |
| Investoren    | Anspruch         | Erfolg des Systems  | Geldgewinn                                    | 10 |    |    |
|               | Anrecht          | Auszahlung des entsprechenden Anteils                           | Rechtliche Vereinbarung zwischen Investor und | 10 |    |    |
| Wettbewerber  | Interesse        | Kein Erfolg des Systems   | geringere Konkurrenz                          |    |    |    |
| (Fach)presse  | Interesse        | Berichterstattungswürdiges System                               | Wirtschaftliche Interesse                     | 4  | 4  | 5  |

Abbildung 3: Stakeholder Teil 2

## 11. Benutzermodelle

### 11.1 User Profiles

Nach dem bestimmen der Stakeholder folgen die User Profiles. Sie beschreiben eine Zielgruppe mit ihren Merkmalen und Ausprägungen. Bei der Entwicklung eines Systems ist es von hoher Wichtigkeit die eigenen Nutzer mit all ihren Merkmalen so früh wie möglich zu. Erst durch die Erkenntnis über die Intentionen der Nutzer kann die richtige Richtung im Entwicklungsprozess eingeschlagen werden.

#### User Profiles (Merkmale mit Begründung)

Tabelle 9: User Profiles (Merkmale mit Begründung)

| Merkmale                             | Begründung  |
|--------------------------------------|---|
| Demografische Charakteristika        | Alter, Geschlecht, Wohnort, sozial-ökonomischer Status...                                     |
| Formale Qualifikationen              | Berufsausbildungen, akademische Titel, Berufsbegleitenden Qualifikationen, Fortbildungen etc. |
| Fachwissen                           | Kenntnisse im (für den in Rede stehenden Entwicklungsprozess) relevanten Bereich.             |
| Computer Kenntnisse und – Erfahrung  | Erfahrungen im Umgang mit dem Computer.   |
| Verfügbare Technologien              | Computer (Hardware), Software und weitere typische Werkzeuge.                                 |
| Spezielle Produkterfahrung           | Erfahrung mit Produkten von Mitbewerbern.   |
| Fähigkeiten und/oder Einschränkungen | Körperliche/geistliche Begabung oder Beschränkung.  |
| Motivation                           | Beweggründe zur Erreichung eines Ziels.   |
| Aufgaben                             | Die Aufgaben die erledigt werden sollen. Primär/Sekundär...                                   |
| Auswirkungen von Fehlern/Versehen    | Potentielle Auswirkungen von menschlichem Fehlverhalten.                                      |

**User Profile:** Ehepartner (ohne Kinder)*Tabelle 10: User Profile, Ehepartner (ohne Kinder)*

| <b>Merkmale</b>                      | <b>Ausprägung</b>   |
|--------------------------------------|---|
| Demografische Charakteristika        | 20-65 Jahre alt, männlich oder weiblich, verheiratet, wohnt mit Ehepartner zusammen                           |
| Formale Qualifikationen              | Arbeiter  |
| Fachwissen                           | Kennt sich mit der Planung von Aktivitäten und dem Zeitmanagement aus   |
| Computer Kenntnisse und – Erfahrung  | Grundkenntnisse   |
| Verfügbare Technologien              | Smartphone, Tablet, Computer  |
| Spezielle Produkterfahrung           | hauptsächlich keine, nur mit einfachen Terminplanern  |
| Fähigkeiten und/oder Einschränkungen | Sehkraft allgemein (sowohl gute wie auch schlechte)   |
| Motivation                           | Reibungsloser Ablauf am Morgen um pünktlich bei der Arbeit zu sein.   |
| Aufgaben                             | Duschen, Frühstück, rechtzeitig am Ziel ankommen.   |
| Auswirkungen von Fehlern/Versehen    | Das System liefert falsche Zeiten, wodurch sich das Vertrauen und gleichzeitig die Nutzung des Systems sinkt. |

**User Profile:** Ehepartner (mit Kindern)*Tabelle 11: User Profile, Ehepartner (mit Kindern)*

| <b>Merkmale</b>                      | <b>Ausprägung</b>   |
|--------------------------------------|---|
| Demografische Charakteristika        | 20-65 Jahre alt, männlich oder weiblich, verheiratet, wohnt mit Ehepartner und Kind(er) zusammen                    |
| Formale Qualifikationen              | Arbeiter oder Hausmann(oder Hausfrau)   |
| Fachwissen                           | Kennt sich mit der Planung von Aktivitäten und dem Zeitmanagement aus   |
| Computer Kenntnisse und – Erfahrung  | Grundkenntnisse   |
| Verfügbare Technologien              | Smartphone, Tablet, Computer  |
| Spezielle Produkterfahrung           | hauptsächliche keine, nur mit einfachen Terminplanern   |
| Fähigkeiten und/oder Einschränkungen | Sehkraft (sowohl gute wie auch schlechte)   |
| Motivation                           | Reibungsloser Ablauf am Morgen um pünktlich bei der Arbeit zu sein und/oder dass die Kinder gefrühstückt hat/haben. |
| Aufgaben                             | Duschen, Frühstück, rechtzeitig am Ziel ankommen oder Kinder rausschicken.  |
| Auswirkungen von Fehlern/Versehen    | Das System liefert falsche Zeiten, wodurch sich das Vertrauen und gleichzeitig die Nutzung des Systems sinkt.       |

**User Profile:** Schüler (Kind eines Ehepaars)*Tabelle 12: User Profile, Schüler (Kind eines Ehepaars)*

| <b>Merkmale</b>                      | <b>Ausprägung</b>   |
|--------------------------------------|---|
| Demografische Charakteristika        | 6-20 Jahre alt, männlich oder weiblich, Single oder vergeben, wohnt mit seinen Eltern zusammen                |
| Formale Qualifikationen              | Schüler   |
| Fachwissen                           | Kennt sich mit der Planung von Aktivitäten und dem Zeitmanagement nicht aus                                   |
| Computer Kenntnisse und – Erfahrung  | Grundkenntnisse   |
| Verfügbare Technologien              | Smartphone, Tablet, Computer  |
| Spezielle Produkterfahrung           | hauptsächliche keine, nur mit einfachen Terminplanern   |
| Fähigkeiten und/oder Einschränkungen | Keine besonderen  |
| Motivation                           | Reibungsloser Ablauf am Morgen um pünktlich bei der Schule zu sein.   |
| Aufgaben                             | Duschen, Frühstück, rechtzeitig am Ziel ankommen.   |
| Auswirkungen von Fehlern/Versehen    | Das System liefert falsche Zeiten, wodurch sich das Vertrauen und gleichzeitig die Nutzung des Systems sinkt. |

**User Profile:** Mitglied (Wohngemeinschaft)

Tabelle 13: User Profile, Mitglied (Wohngemeinschaft)

| <b>Merkmale</b>                      | <b>Ausprägung</b>   |
|--------------------------------------|---|
| Demografische Charakteristika        | 20-40 Jahre alt, männlich oder weiblich, Single oder vergeben, wohnt mit anderen Leuten zusammen              |
| Formale Qualifikationen              | Arbeiter, Student, Ausbildung   |
| Fachwissen                           | Kennt sich mit der Planung von Aktivitäten und dem Zeitmanagement aus   |
| Computer Kenntnisse und – Erfahrung  | Gute Kenntnisse   |
| Verfügbare Technologien              | Smartphone, Tablet, Computer  |
| Spezielle Produkterfahrung           | hauptsächlich keine, nur mit einfachen Terminplanern  |
| Fähigkeiten und/oder Einschränkungen | Privatsphäre, Toleranz, Kompromissbereitschaft  |
| Motivation                           | Reibungsloser Ablauf am Morgen um pünktlich bei der Arbeit/Uni zu sein.                                       |
| Aufgaben                             | Duschen, Frühstück, rechtzeitig am Ziel ankommen.   |
| Auswirkungen von Fehlern/Versehen    | Das System liefert falsche Zeiten, wodurch sich das Vertrauen und gleichzeitig die Nutzung des Systems sinkt. |



## 11.2 Personae

Personae (auch Personas genannt) sind einzelne Individuen, die der Gruppierung eines User Profiles angehören. Sie stellen eine detaillierte Repräsentation einer echten Person dar. Je detaillierter die Persönlichkeit desto authentischer erscheint die Person. Mit so einer Darstellung ist es für die Entwickler viel einfacher sich dieses Individuum vorzustellen und einzuprägen.

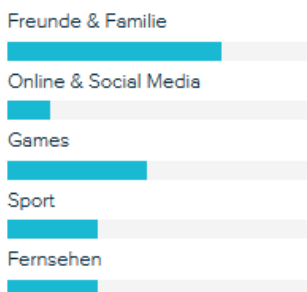
### Paul Bauer



*"Leben heißt Entscheidungen treffen."*

**Alter:** 34  
**Arbeit:** Elektroniker  
**Familie:** Verheiratet  
**Wohnort:** Viersen, NRW  
**Character:** ruhig, selbstsicher  
**Sternzeichen:** Krebs

### Freizeitaktivitäten



### Ziele

- Spaß an der Arbeit.
- Viel reisen und neue Orte in Welt entdecken.
- Erfolgreich sein um meine Familie verwöhnen zu können.

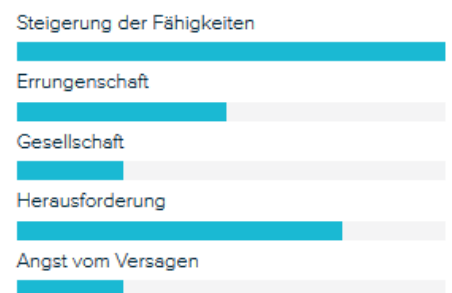
### Bio

Paul ist ein sehr aufgeschlossener junger Mann, der schon in seinen jungen Jahren großes Interesse an der Elektronik gezeigt hat. Schon seit Kindertagen baute er elektronische Schaltungen mit seinem Großvater zusammen. Um seinen Entdeckerdrang zu befriedigen beschloss er schon sehr früh was er später mal als Erwachsener tun möchte. Seit 2004 ist er ein ausgebildeter Elektroniker mit der Fachrichtung für Automatisierung. Mit seinem Traumberuf in der Hand fehlte ihm nur noch seine bessere Hälfte zu seinem vollständigen Glück. Er heiratete 2010 Anna und lebt seitdem glücklich in dem ländlichen Viersen, NRW.

In seiner Freizeit spielt er leidenschaftlich in einem Fußballverein, dass ihm als Ausgleich zur seinem Bürojob dient. Seine Lieblingshobbys sind Spiele zocken auf seiner PS4 und rumbasteln an seinen kleinen Projekten. Sein letztes Projekt beschäftigt sich mit der Thematik des "Smarthome" und der Vernetzung von Bewegungssensoren und Lichtquellen. Was ihm aber am meisten Spaß macht ist etwas mit seinen Freunden oder der Familie zu unternehmen.

Neuerdings fehlt ihm morgens immer mehr Zeit, wodurch sein Beginn in den Tag ziemlich stressig anfängt. Das Bad ist unregelmäßig besetzt und das Frühstück wird aus Zeitgründen vernachlässigt. Was folgt ist schlechte Laune. Um dieses Problem zu lösen setzt er seine Hoffnung in die Zeitmanagement- und Aktivitätenapp "InTime".

### Motivation



### Personalität



### Erwartungen an InTime

- Einfache Bedienung der Software.
- Zuverlässige Zeitangaben und Fahrplanauskünfte.
- Koordinierter Ablauf von Aktivitäten und den Tätigkeiten seiner Frau, da er seine Zeit am Morgen gern ruhig verbringt ohne daran denken zu müssen vor seiner Gattin ins Bad zu kommen.

Abbildung 4: Persona Paul Bauer, Bezug auf User Profile: Ehepartner (ohne Kinder)

## Anna Bauer



*"Leisten wir uns den Luxus, eine eigene Meinung zu haben."*

**Alter:** 31  
**Arbeit:** Krankenpflegerin  
**Familie:** Verheiratet  
**Wohnort:** Viersen, NRW  
**Character:** spontan, gutgläubig  
**Sternzeichen:** Steinbock

### Freizeitaktivitäten

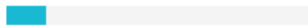
Freunde & Familie



Online & Social Media



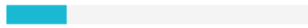
Sport



Fernsehen



Shoppen



Zeichnen



### Ziele

- Menschen so gut es geht zu helfen.
- Viele unterschiedliche Kulturen bereisen und kennenlernen.
- Familie immer treu zur Seite stehen und helfen, wenn jemand in Not geraten ist.
- Finanziell abgesichert zu sein und sich keine Sorgen über die Zukunft machen.

### Bio

In jungen Jahren lernte Anna alles von ihrer Mutter, Sophia Stein über das Verarbeiten von Menschen kennen. Sophie selbst war damals Krankenschwester im örtlichen Krankenhaus von Mönchengladbach, NRW. Die kleine Anna war so begeistert von der Hilfe die man Menschen geben kann um deren Leben zu verbessern oder gar zu retten. Nach Jahren des Selbststudiums und dem Abschluss an dem Hugo-Junkers-Gymnasium war es ihr klar, dass sie den Fußtapfen ihrer Mutter folgt und sich als Krankenpflegerin versuchen will. Mit 17 Jahren machte sie eine Ausbildung zu Gesundheits- und Krankenpflegerin an der Schön Klinik in Düsseldorf. Für diese Ausbildung wohnte Anna zusammen mit einer Bekannten in einer Wohngemeinschaft. Was nicht immer leicht war, da es ab und zu Konflikte über die Belegung des Badezimmers gab. Durch Durchhalte- und Durchsetzungsvermögen überstand es Anna erfolgreich bis zum Abschluss der Ausbildung. Nach 4 Jahren Arbeit bei der DaVita als vollwertige Krankenpflegerin lernte sie Paul Bauer kennen der ein Enkel eines Patienten war. Kurz darauf heiratet Anna Paul und übernimmt seinen Namen. Da eine kleine Wohnung nicht mehr in Frage kam, kauften sich die Bauer ein Haus in Viersen.

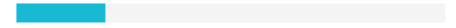
Als glückliches Ehepaar reist Sie gerne mit ihrem Mann durch die Welt und vertreibt sich die Freizeit mit Zeichnen von Stillleben. Untypischer Weise geht Anna nicht gerne shoppen, sondern bestellt alles gerne Online auf Zalando und Amazon.

### Motivation

Steigerung der Fähigkeiten



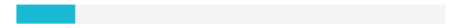
Errungenschaft



Gesellschaft



Herausforderung



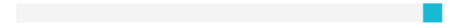
Angst vom Fehlverhalten



### Personalität

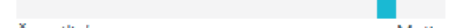
Introvertiert

Extravertiert



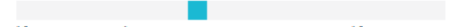
Rational

Emotional



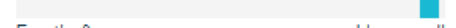
Ängstlich

Mutig



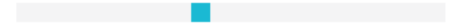
Kompromisslos

Kooperativ



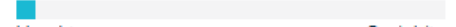
Ernsthaft

Humorvoll



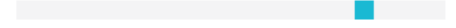
Schmächtig

Kräftig



Unruhig

Geduldig



### Erwartungen an InTime

- Zuverlässige Angaben über anstehende Aktivitäten und wie lange diese Andauern.
- Entlastung bei der Morgenplanung.
- Konflikte um das Benutzen des Badezimmers sollten selbstverständlich verhindert werden.

Abbildung 5: Persona Anna Bauer, Bezug auf User Profile: Ehepartner (ohne Kinder)

## Andreas Stiel



*"Kein Genuss ist vorübergehend,  
denn der Eindruck, den er  
hinterlässt, ist bleibend."*

**Alter:** 46

**Arbeit:** Chefkoch

**Familie:** Verheiratet, Zwei Kinder

**Wohnort:** München, Bayern

**Character:** selbstkritisch,  
Perfektionist

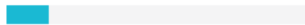
**Sternzeichen:** Löwe

### Freizeitaktivitäten

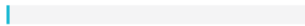
Freunde & Familie



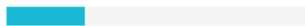
Online & Social Media



Fernsehen



Kochen



Brettspiele



Gartenpflege



### Ziele

- Eine selbstbestimmende Lebensweise ohne auf Verzicht von wertvollen Dingen.
- Erziehen der Kinder zu guten Menschen.
- Ein Zuhause schaffen in dem man seine Zukunft verbringen will.

### Bio

Andreas Stiel ist seit 2001 verheiratet mit Lena Richter, eine Kinderzahnärztin aus München. Er selbst arbeitete 1992 als Koch bei der Sofitel Hotels & Resorts und 2007 als Chefkoch in der Schwarzreiter Tagesbar & Restaurant. Die Leidenschaft zur Kochkunst ist aufgeflammt nachdem der junge Andreas im Alter von 17 auf tragischer Weise seine Mutter verloren hat. Von morgens bis abends musste sein Vater arbeiten und Andreas allein zuhause, musste schnell lernen für sich selbst zu sorgen. Nach und nach gefiel es ihm Rezepte auszuprobieren und selber Kreationen zaubern. Lena lernte er während eine Hochzeit eines Freundes kennen. Dort half er bei dem Buffet und der Hochzeitstorte. Und nach etwas Zeit zusammen verheiratet, kam Alina zu Welt. 5 Jahre danach folgte Kai. Durch den Zuwachs der Familie Stiel wurde die gemeinsame Wohnung in der Nähe von Andreas Arbeitsplatzes ziemlich klein. Etwas Größeres musste her. Nach dem Umzug in das gemütliche Allach-Untermenzing.

Durch die Umstellung der Lebenssituation gibt es viele Veränderungen im Ablauf des Tages. Nicht immer läuft alles nach Plan wie Andreas es plant. Da aber er genau wie seine Frau in Arbeitsschichten arbeitet fallen viele Besorgungen weg oder werden vergessen.

Als Besitzer eines neuen Einfamilienhauses entwickelte sich schnell ein neues Hobby für Andreas. Die Gartenpflege. Die Eigenschaften eines Perfektionisten und eines Überordentlichen Menschen färbten auch auf den Garten ab. Aber nicht nur der Garten wurde zu seinem neuen Hobby. Auch die Familienabende bei den viele Brettspiele gespielt werden wurden zu alltäglichen Gewohnheit.

### Motivation

Steigerung der Fähigkeiten



Errungenschaft



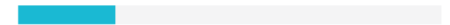
Gesellschaft



Herausforderung



Angst vom Fehlverhalten



### Personalität

Introvertiert

Extravertiert

Rational

Emotional

Ängstlich

Mutig

Kompromisslos

Kooperativ

Ernsthaft

Humorvoll

Schmächtig

Kräftig

Unruhig

Geduldig

### Erwartungen an InTime

- Auf viele Lebenssituationen anwendbar.
- Aktivitäten ergänzen oder erweitern können.
- Benachrichtigungen bekommen falls sich was ändert.
- Richtige Informationsausgabe.
- Auf Veränderungen flexibel und korrekt einen Plan für die ganze Familie erstellen.

Abbildung 6: Persona Andreas Stiel, Bezug auf User Profile: Ehepartner (mit Kinder)

## Lena Stiel



*"Mindestens 98 Prozent der Dinge, um die wir uns Sorgen machen, treffen niemals ein."*

**Alter:** 42

**Arbeit:** Kinderzahnarzt

**Familie:** Verheiratet, Zwei Kinder

**Wohnort:** München, Bayern

**Character:** gebildet, offenherzig

**Sternzeichen:** Krebs

### Freizeitaktivitäten

Freunde & Familie

Online & Social Media

Fernsehen

Fahrrad fahren

Bücher lesen

### Ziele

- Dass die Familie glücklich und gesund ist.
- Perspektiven bei der Arbeit.
- Viel Zeit mit der Familie verbringen.
- Sportlich aktiv bleiben solange es geht.

### Bio

Von Geburt an lebte Lena Bertels mit vier Geschwistern und den Eltern zusammen in München. Die Großfamilie Bertels war eine reine Mädchenbande. Sie besaßen eine kleine Autowerkstatt und wohnten in dem vererbten Haus der Großeltern Mütterlicher Seite in dem Stadtteil Berg am Laim. Mit 7 besuchte Lena die örtliche Grundschule und mit 11 das Michaeli-Gymnasium München. Nach dem Abschluss des Abiturs im Jahre 1994 wollte Lena eine Kinderärztin werden. Sie liebte Kinder über alles und strebte die Arbeit mit ihnen an. An der Ludwig-Maximilians-Universität München studierte sie Zahnmedizin und arbeitet seit 1997 in einer Spezialpraxis für Kinderzahnheilkunde. Nach einer Schicksalhaften Begegnung während einer Hochzeit war Lena mit dem Koch Andreas Stiel zusammen. 2002 Heiratete Lena Andreas nahm seinen Namen an. Durch den großen Wunsch des frisch verheirateten Paares Kinder zu bekommen wurde die Familie schnell größer. 2003 kam Alina zu Welt und 2008 folgte auch Kai. Heute lebt Lena Stiel mit ihrer Familie im ruhigen Allach-Untermenzing.

Als erfolgreiche Zahnärztin und Mutter hat Lena viel zu tun und kann ihren Hobbys wie Fahrrad fahren oder Romane lesen nicht immer nachkommen, weswegen sie sich manchmal wünscht, dass jemand für Sie den Tagesplan bestimmt.

### Motivation

Steigerung der Fähigkeiten

Errungenschaft

Gesellschaft

Herausforderung

Angst vom Versagen

### Personalität

Introvertiert

Extravertiert

Rational

Emotional

Ängstlich

Mutig

Kompromisslos

Kooperativ

Ernsthaft

Humorvoll

Schmächtig

Kräftig

Unruhig

Geduldig

### Erwartungen an InTime

- Organisierter Tagesablauf ohne stressige Probleme.
- Planung von Zielorientierten Zeit- und Routenplanung.
- Automatisierte Benachrichtigung beim Ausfall vom Schulbus.

Abbildung 7: Persona Lena Stiel, Bezug auf User Profile: Ehepartner (mit Kinder)

## Alina Stiel



*"Neid ist die höchste Form der Anerkennung."*

**Alter:** 14

**Arbeit:** Realschüler

**Familie:** Single

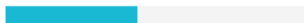
**Wohnort:** München, Bayern

**Character:** eingebildet, modisch

**Sternzeichen:** Stier

### Freizeitaktivitäten

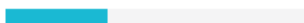
Freunde & Familie



Online & Social Media



Fernsehen



Shoppen



### Ziele

- Viele Freunde die mich so respektieren wie ich bin.
- Den Traumberuf als Modedesignerin verwirklichen.
- Immer modisch auf dem aktuellsten Stand sein.

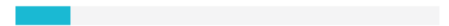
### Bio

Alina Stiel ist am 13. Mai. 2003 in München geboren worden. Sie besucht momentan die 9 Klasse an der Maria-Ward-Mädchenrealschule. Sie ist ein zierliches und sehr Mode bewusstes Mädchen. Ihr Traumberuf ist der des Modedesigners, was Sie mit Hilfe ihrer Tante Emilie Schwarz, die schon Modedesignerin ist, erreichen möchte. Schon als kleines Mädchen mochte sie es neue Kleider auszusuchen und anzuziehen. Falls es mit dem Job als Modedesigner nichts wird will sie als Model durchstarten. Sie hat in dem Bezug noch keine Erfahrung, aber kennt die Verfahren und Techniken aus dem Fernsehen. "Als zielstrebiges und selbstsicheres Mädchen wird sie alles was Sie anstrebt sicherlich erreichen", so die Tante. Hinter diesen Berufswünschen steht nicht jeder in der Familie Stiel hinter ihr. Aber das macht ihr dem dickköpfigen Mädchen nichts aus und beharrt weiterhin auf ihrer Sache. Alina hat einen kleinen der ihr immer als Versuchskaninchen dient bei den Experimenten mit neuen Schminkmethoden.

Ihre Hobbys sind sehr aufs Auftreten und Aussehen bezogen. Shoppen mit Freundinnen und stundenlanges Posen für die Social Media Plattform Instagram gehören zum Alltag dazu.

### Motivation

Steigerung der Fähigkeiten



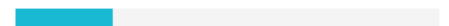
Errungenschaft



Gesellschaft



Herausforderung



Angst vom Fehlverhalten



### Personalität



### Erwartungen an InTime

- Die Aktivitäten müssen einer Priorisierung folgen. Morgens sollte immer Zeit sein zum Schminken.
- Die Zeiten sollten immer genau sein.

Abbildung 8: Persona Alina Stiel, Bezug auf User Profile: Schüler (Kind eines Ehepaars)

## Kai Stiel



*"Ich schreibe nicht ab, das ist Teamwork!"*

**Alter:** 9

**Arbeit:** Realschüler

**Familie:** Single

**Wohnort:** München, Bayern

**Character:** verlässlich, artig

**Sternzeichen:** Steinbock

### Freizeitaktivitäten

Freunde & Familie

Online & Social Media

Fernsehen

Games

Skateboard

Fussball

### Ziele

- Ein Profi-Skateboarder wie Tony Hawk werden.
- Viel Geld besitzen und reich werden.
- Eine hübsche Freundin haben.

### Bio

Kai Stiel, ist ein neunjähriger kleiner und schmächtiger Junge aus München der den Großen Taum hegt Profi Skateboarder zu werden. Jeden Tag trainiert er hart draußen mit seinen Freunden neue Tricks und Moves mit dem Skateboard. Momentan besucht er die 4.Klasse an der Grundschule Berg am Laim. Seine Familie besteht aus den beiden Eltern Andreas und Lena Stiel und seiner großen nervigen Schwester Alina. Er hat seine große Schwester gern aber nur dann, wenn sie nicht versucht an ihm neue Schminke zu testen.

In seiner Freizeit verbringt er bei gutem Wetter immer draußen und trainiert seine Künste mit dem Board während bei miesen Wetterverhältnissen er lieber zuhause an dem Fernseher klebt und Videospiele spielt. Aber auch die Familienabende, die ab und zu gemacht werden findet er sehr unterhaltsam.

### Motivation

Steigerung der Fähigkeiten

Errungenschaft

Gesellschaft

Herausforderung

Angst vom Versagen

### Personalität

Introvertiert

Extravertiert

Rational

Emotional

Ängstlich

Mutig

Kompromisslos

Kooperativ

Ernsthaft

Humorvoll

Schmächtig

Kräftig

Unruhig

Geduldig

### Erwartungen an InTime

- Soll verhindern, dass der Schulbus nicht verpasst wird.

Abbildung 9: Persona Kai Stiel, Bezug auf User Profile: Schüler (Kind eines Ehepaars)



## Stefan Kahl



*"Dinge haben immer nur den Wert, den du ihnen verleihst."*

**Alter:** 24  
**Arbeit:** BWL Student  
**Familie:** Single  
**Wohnort:** Münster, NRW  
**Character:** gesellig, neugierig  
**Sternzeichen:** Wassermann

### Freizeitaktivitäten

Freunde & Familie

Online & Social Media

Games

Karate

Sportsbar

Bücher lesen

### Ziele

- Grundlage für die Zukunft schaffen und sich absichern.
- Erfolgreich sein und selbständig werden.
- Nicht mehr zu rauchen.

### Bio

Mit 6 Jahren wurde Stefan Kahl eingeschult in die Grundschule Maria Frieden in Coesfeld, NRW. Schon in den frühen Jahren bemerkte seine alleinerziehende Mutter Brigitte, dass der Junge ein Händchen für Zahlen hatte. Gerne löste er Sudoku- oder Matherätsel. Mit Matheaufgaben tat er sich nie schwer und erledigte sie sofort nach dem Nachhause kommen. Nur sprachliche Fächer lagen ihm nicht besonders gut. Die Grundschule beendete er 2003 und besuchte darauffolgend die Theodor-Heuss-Realschule in Coesfeld. Dort nahm er in der 6. Klasse erfolgreich an einer regionalen Matheolympiade teil. Nach seinem exzellenten Abschluss an der Realschule 2009 führte ihn sein Werdegang Richtung Abitur. Dafür besuchte er das Städtisches Ganztagsgymnasium Nepomucenum. In dieser Zeit entschloss er sich sein Wissen mittels eines Studium zu erweitern. Mit dem Jahr 2012 war auch an dem Gymnasium der Abschluss erreicht. Stefan machte sich viele Gedanken über seine Zukunft. Wo sollte er leben? Bei seiner Mutter und dann jeden Morgens pendeln oder lieber doch eine Wohngemeinschaft suchen? Und wo studieren? Fragen über Fragen. Er durchdachte jeden Schritt und informierte sich über Unis in der Nähe. Nach langem Suchen und vielen Aussortierungen entschloss er sich für die FH Münster die Betriebswirtschaft als Studiengang anbietet. Der entscheidende Faktor war hier eine Wohngemeinschaft, die in der Nähe der FH liegt. Es war ein neuer Anfang und ein riesiger Schritt in das Selbstständige leben. Der Abschied von seiner Mutter viel erstmal schwer, aber die Tatsache, dass der neue Wohnort eine Stunde mit der Bahn entfernt ist hat ihn und seine Mutter doch beruhigt. Die neue Lebenseinstellung war auch eine Erfahrung für sich. Der sonst eher kompromisslose Stefan musste schnell einsehen, dass das Leben auf einer engen Umgebung mit Fremden für einige Konflikte sorgt und Kooperation das A und O in so einer Gemeinschaft ist. Umso eine Zeit friedlich zu überstehen braucht man gutes Zeit- und Selbstmanagement.

### Motivation

Steigerung der Fähigkeiten

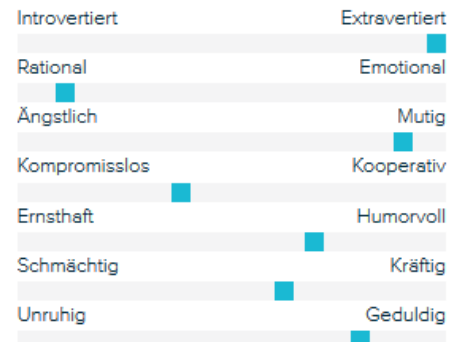
Errungenschaft

Gesellschaft

Herausforderung

Angst vom Versagen

### Personalität



### Erwartungen an InTime

- Unterstützt bei der Planung des Tagesablaufs.
- Aktuelle Veränderungen die meine Route betreffen anzeigen.
- Zuverlässige Auskünfte.
- Flexible Anpassung an das Verändern der Aktivitäten und ihrer Dauer.
- Intuitives steuern durch das User Interface.

Abbildung 10: Persona Stefan Kahl, Bezug auf User Profile: Mitglied (Wohngemeinschaft)

## Can Gür



*"Schöne Worte sind nicht immer wahr, wahre Worte sind nicht immer schön."*

**Alter:** 22

**Arbeit:** Medieninformatik Student

**Familie:** Single

**Wohnort:** Gummersbach, NRW

**Character:** couragiert, pragmatisch

**Sternzeichen:** Krebs

## Freizeitaktivitäten

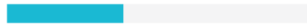
Freunde & Familie



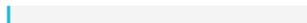
Online & Social Media



Games



Sport



Anime gucken



Programmieren



## Ziele

- Erfolgreiches Abschließen des Studiums.
- Familie finanziell unterstützen.
- Zukünftig ein Beruf ausüben der so Spaß macht, dass er sich nicht nach Arbeit anfühlt.

## Bio

Can Gür ist seit seinem 4 Lebensjahr in Deutschland. Seine Eltern sind aus der Türkei während der Jahrtausendwende ausgewandert, da aus Familiären Gründen es eine Möglichkeit bestand als Spätaussiedler die deutsche Staatsbürgerschaft zu beantragen. Diese Chance nutzten Atakan und Hilal Gür, die Eltern von Can. Die Reise ins unbekannte startete aus der Großstadt Gebze, über Istanbul nach Deutschland mit dem Ziel Friedland für die notwendige Erfassung und Zulassungen der Personen. Nach einen Monat Aufenthalt in einem Lager ging die Reise weiter in das ländliche Düren, dass sich als Endstation herausstellte. Schnell integrierten sich die Gür in die europäische Gesellschaft. Atakan und Hilal Gür fanden schnell Arbeit und der kleine Can besuchte einen städtischen Kindergarten. Der Besuch des Kindergartens war nicht ganz so einfach. Die Gür mussten jeden morgen früh raus und mit dem Fahrrad 7 Kilometer fahren um rechtzeitig den kleinen Can abzuliefern und pünktlich an dem Arbeitsplatz zu erscheinen. Eine schwierige Zeit ohne Auto. Mit sieben Jahren verließ der junge Can den Kindergarten und wurde an der Martin-Luther-Grundschule eingeschult. Die Zeit in der Grundschule war nicht immer einfach. Sprachbezogene Fächer fielen ihm besonders schwer was auch in den Noten stark bemerkbar war. Nichts desto trotz meisterte Can es in den Jahren durch starke Disziplin und Entschlossenheit die Grundschule. 2006 begann die Zeit der Sekundarstufe 1 auf der Realschule Düren Bretzelnweg. Während diesem Aufenthalts setzte sich Can das erste Mal mit Informatik auseinander. Über ein Wahlpflichtfach entstand der Kontakt mit der Thematik. Can war hin und weg von der Vielfalt an Tätigkeiten und dem Potenzial als Programmierer wodurch sich der Wunsch dies auch Beruflich zu betreiben immer weiter stärkte. Mit einem Ziel vor Augen machte er sein Abschluss an der Realschule 2012 und das Fachabitur als Informationstechnischer Assistenten auf einem Berufskolleg erledigte er 2015. Wissensbegierig beschloss er bei seinen Eltern auszuziehen und in das 100 km gelegene Gummersbach in einer Wohngemeinschaft umzuziehen. Der Grund dafür ist ein Studium. Die Technische Hochschule Köln führte in Gummersbach einen Campus und bot attraktive Informatikstudiengänge an wodurch sich die Entscheidung die Eltern zu verlassen leichter machte. Mit Fremden Leuten kam Can immer bestens aus. Mit seinem Humor und seiner ehrlichen Art hat er immer einen Draht zu anderen Menschen.

## Motivation

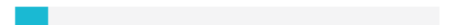
Steigerung der Fähigkeiten



Errungenschaft



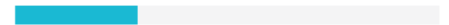
Gesellschaft



Herausforderung

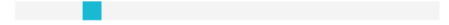


Angst vom Versagen



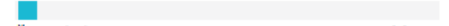
## Personalität

Introvertiert



Extravertiert

Rational



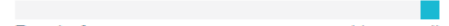
Emotional

Ängstlich



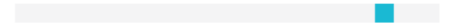
Mutig

Kompromisslos



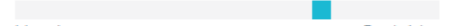
Kooperativ

Ernsthaft



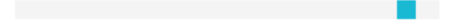
Humorvoll

Schmächtig



Kräftig

Unruhig



Geduldig

## Erwartungen an InTime

- Eindeutiges Design.
- Exakte Angaben über die aktuelle Lage der Abfahrtszeiten.
- Möglichkeiten seine eigenen Aktivitäten zu erstellen.
- Die wichtigsten Sachen müssen als erstes abgearbeitet werden.

Abbildung 11: Persona Can Gür, Bezug auf User Profile: Mitglied (Wohngemeinschaft)



---

## 12. Benutzungsmodelle

### 12.1 Use Cases

Ein use case, auch Anwendungsfall genannt, stellt die Interaktion zwischen einer Rolle (Stakeholder, zum Beispiel Benutzer) und einem System dar. Diese Interaktion beinhaltet die Aufgabe eines Benutzers, die mit Hilfe eines Systems gelöst werden sollen. Dabei wird das Ziel, dass für die Erreichung des use cases angepeilt eindeutig beschrieben. Des Weiteren verfügt das use case über Informationen, die für das Identifizieren von Problemen oder Schwachpunkten zwischen Benutzer und System dienen können.

Ein Anwendungsfall besteht nach Alistair Cockburn aus mehreren Bestandteilen.

- ➔ Einer eindeutigen Nummer zur Identifikation eines Anwendungsfalls. Wobei es auch eine kurze und Aussagekräftige Bezeichnung sein kann.
- ➔ Charakteristischen Informationen die den Hauptakteur, das Ziel, das System und die damit verbundenen Situationen genauesten beschreiben.
- ➔ Ein Haupt-Szenario, dass in geordneter Reihenfolge die ausführenden Schritte des ganzen Szenarios beschreiben.
- ➔ Ggf. auch Erweiterungen oder Unter-Variationen von dem Haupt-Szenario und weiter optionale Angaben.

Der Benutzer registriert sich im System an.

Tabelle 14: Use case #1

| <b>USE CASE #1</b>        |  | <b>Erstellen eines neuen Accounts</b>  |
|---------------------------|--|--|
| Goal in Context           | Einen neuen Account (ggf. mit allen wichtigen Informationen über die Person, sonst nur Benutzername und Passwort als Pflichteingabe) erfolgreich erstellt. |  |
| Scope & Level             | InTime Android Client , Subfunction  |  |
| Preconditions             | Der Benutzer hat die App installiert und den Nutzerbedingungen zugestimmt.   |  |
| Success End Condition     | Account erstellt.  |  |
| Failed End Condition      | <ul style="list-style-type: none"> <li>• Verlust des Internets.</li> <li>• Vergebener Benutzername.</li> <li>• Passwort zu kurz oder zu lang.</li> </ul>   |  |
| Primary, Secondary Actors | Benutzer der „InTime“ App  |  |
| Trigger                   | Button „Account erstellen“ vom Benutzer betätigt.  |  |
| <b>DESCRIPTION</b>        | <b>Step</b>  | <b>Action</b>  |
|                           | 1  | - Benutzer betätigt den Button „Account erstellen“.  |
|                           | 2  | - Benutzer gibt seine persönlichen Daten ein.  |
|                           | 3  | - Benutzer trägt seinen gewünschten Benutzernamen und Passwort ein.  |
|                           | 4  | - Benutzer bestätigt die Eingaben.   |
| <b>EXTENSIONS</b>         | <b>Step</b>  | <b>Branching Action</b>  |
|                           | 2a   | - Solange der Benutzer keine Eingaben in die Felder tätigt werden diese rot markiert um zu signalisieren, dass dort noch was fehlt, sodass der Nutzer diese nachtragen kann.<br>Hinzukommend wir ein kleiner Benachrichtigungstext ausgegeben. |
|                           | 3a   | - Der Benutzer gibt kein Passwort ein und wird benachrichtigt (siehe 2a).  |
| <b>SUB-VARIATIONS</b>     | <b>Step</b>  | <b>Branching Action</b>  |
|                           |  | -  |

Der Benutzer meldet sich im System an.

Tabelle 15: Use Case #2

| <b>USE CASE #2</b>        |   | <b>Im System anmelden</b>  |
|---------------------------|---|--|
| Goal in Context           | Erfolgreiche Anmeldung des Benutzers.   |  |
| Scope & Level             | InTime Android Client , Subfunction   |  |
| Preconditions             | Benutzer muss den Benutzernamen und Passwort kennen.  |  |
| Success End Condition     | Nutzer ist erfolgreich im System angemeldet.  |  |
| Failed End Condition      | <ul style="list-style-type: none"> <li>• Verlust der Internetverbindung.</li> <li>• Benutzername falsch geschrieben oder dem System nicht bekannt.</li> <li>• Passwort falsch geschrieben oder dem System nicht bekannt.</li> </ul> |  |
| Primary, Secondary Actors | Benutzer der „InTime“ App   |  |
| Trigger                   | Button „Login“ vom Benutzer betätigt.   |  |
| <b>DESCRIPTION</b>        | <b>Step</b>   | <b>Action</b>  |
|                           | 1   | - Benutzer betätigt den Button „Login“ im Startscreen.   |
|                           | 2   | - Benutzer gibt seinen Benutzernamen im oberen Eingabefeld ein.  |
|                           | 3   | - Benutzer gibt sein Passwort im unteren Eingabefeld ein.  |
|                           | 4   | - Benutzer betätigt erneut den „Login“-Button.   |
| <b>EXTENSIONS</b>         | <b>Step</b>   | <b>Branching Action</b>  |
|                           | 2a  | - Solange der Benutzer den Benutzernamen nicht eingibt wird dieses Feld rot markiert.                    |
|                           | 3a  | - Solange der Benutzer das Passwort nicht eingibt wird dieses Feld rot markiert.                         |
| <b>SUB-VARIATIONS</b>     | <b>Step</b>   | <b>Branching Action</b>  |
|                           | 2b, 3b  | - Der Benutzer wählt die mögliche Funktionalität sich mit einem Google oder Facebook Account anzumelden. |

Der Benutzer trägt Aktivitäten ins System ein.

Tabelle 16: Use Case #3

| <b>USE CASE #3</b>        |   | <b>Aktivitäten ins System eintragen</b>   |
|---------------------------|---|---|
| Goal in Context           | Erfolgreich Aktivitäten vom Benutzer eingetragen.   |   |
| Scope & Level             | InTime Android Client , Primary Task  |   |
| Preconditions             | Im System angemeldet.   |   |
| Success End Condition     | Aktivitäten wurden eingetragen und gespeichert.   |   |
| Failed End Condition      | <ul style="list-style-type: none"> <li>Das Abspeichern der Aktivitäten dauert zu lange und wird abgebrochen.</li> </ul> |   |
| Primary, Secondary Actors | Benutzer der „InTime“ App   |   |
| Trigger                   | Button „Add activity“ vom Benutzer betätigt.  |   |
| <b>DESCRIPTION</b>        | <b>Step</b>   | <b>Action</b>   |
|                           | 1   | - Der Benutzer betätigt den Button „Add activity“.                                      |
|                           | 2   | - Der Benutzer trägt den Namen der Aktivität ein.                                       |
|                           | 3   | - Der Benutzer wählt die gewünschte Priorität aus für die zu erstellende Aktivität aus. |
|                           | 4   | - Der Benutzer wählt die Zeit aus die er für die Aktivität vorgesehen hat.              |
|                           | 5   | - Der Benutzer bestätigt die Eingaben, indem er den Button „Fertig“ betätigt.           |
| <b>EXTENSIONS</b>         | <b>Step</b>   | <b>Branching Action</b>   |
|                           |   | -   |
| <b>SUB-VARIATIONS</b>     | <b>Step</b>   | <b>Branching Action</b>   |
|                           |   | -   |

Der Benutzer erstellt einen Zeitplan.

Tabelle 17: Use Case #4

| <b>USE CASE #4</b>        |  | <b>Zeitplan wird erstellt</b>   |
|---------------------------|--|---|
| Goal in Context           | Erfolgreiches Erstellen eines Zeitplans.   |   |
| Scope & Level             | InTime Android Client , Primary Task   |   |
| Preconditions             | Aktivitäten müssen vorhanden sein.   |   |
| Success End Condition     | Ein Zeitplan wurde erstellt.   |   |
| Failed End Condition      | <ul style="list-style-type: none"> <li>• Viele Aktivitäten mit hoher Priorität verhindern das Erstellen eines Zeitplans.</li> <li>• Ankunftszeit liegt in der Vergangenheit.</li> <li>• Ankunftsort ist unbekannt.</li> <li>• Die minimale Aufweckzeit liegt zu nah an der Ankunftszeit, sodass es keine ausreichende Zeit für die Ausführung der Aktivitäten gibt.</li> <li>• Zu viele oder zu lange Aktivitätenbearbeitungsdauern-.</li> <li>• Aktuelle Position nicht bekannt oder wird nicht erkannt.</li> </ul> |   |
| Primary, Secondary Actors | Benutzer der „InTime“ App  |   |
| Trigger                   | Button „Zeitplan erstellen“ vom Benutzer betätigt.   |   |
| DESCRIPTION               | Step   | Action  |
|                           | 1  | - Der Benutzer betätigt den Button „Zeitplan erstellen“.  |
|                           | 2  | - Der Benutzer betätigt den Button „Aktivitäten einfügen“.  |
|                           | 3  | - Der Benutzer wählt die gewünschten Aktivitäten aus einer Liste aus und bestätigt mit dem Button „Fertig“.                     |
|                           | 4  | - Der Benutzer trägt den aktuellen Aufenthaltsort ein.  |
|                           | 5  | - Der Benutzer trägt den Ankunftsort ein.   |
|                           | 6  | - Der Benutzer trägt die Ankunftszeit ein.  |
|                           | 7  | - Der Benutzer trägt die minimale Aufweckzeit ein.  |
| EXTENSIONS                | Step   | Branching Action  |
|                           |  | -   |
| SUB-VARIATIONS            | Step   | Branching Action  |
|                           | 4a   | Der Benutzer verwendet die mögliche Funktion „Aktuelle Position“ um mittels GPS seinen aktuellen Aufenthaltsort herauszufinden. |

## 13. Anforderungen

Durch die Zielhierarchie erhielten wir einen groben Überblick über die Anforderungen des Systems, was aber in keinsten Weise ausreichend für die weiterführende Entwicklung ist. Deswegen erstellten wir als Leitfaden für die Gestaltung eine ausführliche Aufzählung von funktionalen, non-funktionalen und Organisatorischen Anforderungen.

### 13.1 Funktionale Anforderungen

#### → Das System soll/muss...

- ... Aktivitäten speichern können.
- ... eine Wecker Funktion besitzen.
- ... die Abfahrtszeit des Verkehrsmittels berechnen können.
- ... anzeigen wann die nächste Aktivität beginnt.
- ... falls notwendig den Benutzer benachrichtigen können.
- ... Accounts im selben Haushalt verbinden können.
- ... die Aktivitäten nach Dringlichkeit der jeweiligen Personen eigenständig verteilen können (Beispielsweise das Bad).
- ... GPS-Koordinaten des Nutzers erfassen und verarbeiten können.
- ... die Zeit berechnen können, die ein Nutzer von seinem Standort zur Bushaltestelle benötigt.
- ... auf die Busfahrpläne zugreifen können.

### 13.2 Nicht funktionale Anforderungen

#### 13.2.1 Produktqualität *(nach ISO 25010)*

#### → Externe Qualität

- Funktionalität
  - Das System muss die Aufgaben für den Benutzer fehlerfrei bewältigen.
  - Das System muss die Aufgaben vollständig bearbeiten.
- Performanz
  - Das System muss über eine kurze Antwortzeit auf Nutzereingaben verfügen.
  - Das System sollte nicht überflüssige Ressourcen gebrauchen oder besetzen.
  - Das System sollte sparsam mit dem Speicherplatz umgehen.
- Kompatibilität
  - Das System sollte auf unterschiedlichen Android Geräten laufen können.

- Das System muss für die volle Funktionalität mit den notwendigen Systemen kommunizieren können.
- Das System muss aus dem Datenstrom Informationseinheiten identifizieren können.
- Benutzbarkeit
  - Die Benutzerschnittstelle des Systems sollte dem Nutzer in irgendeiner Weise vertraut vorkommen.
  - Die Verwendung des Systems sollte für den Benutzer leicht erlernbar sein.
  - Das System muss für den Benutzer die wichtigsten Funktionalitäten leicht zugänglich zu Verfügung stellen.
- Zuverlässigkeit
  - Das System muss auf Fehler des Benutzers reagieren.
- Sicherheit
  - Das System muss vertraulich mit den Daten vom Benutzer umgehen.
  - Das System sollte auf Angriffe an verwendete Diensten (z.B. API) verhindern können.
  - Das System sollte nicht für Angriffe gegen andere Dienste missbraucht werden können.
  - Das System sollte Nutzerauthentifizierungen durchführen können.

#### → Interne Qualität

- Wartbarkeit
  - Das System sollte sauber und performant programmiert sein.
  - Das System sollte für Fachpersonal wartbar sein.
- Portierbarkeit
  - Das System sollte erweiterbar sein.

#### Organisatorische Anforderungen

- Das Produkt sollte in jedem Entwicklungsschritt mit einer Dokumentation begleitet werden.
- Es muss einen Projektplan geben der alle Aktivitäten auflistet. Mit dem sich auch bestimmen lässt wann und wie lange eine Person an einer Aktivität gearbeitet hat.

## 14. Kommunikationsmodell

Ein Kommunikationsmodell ist eine Darstellung von dem gegenwärtigen und der möglichen Verständigung von Instanzen. Diese Instanzen haben sich bei der Stakeholder Analyse herausgestellt.

### 14.1 Deskriptives Kommunikationsmodell

Bei dem Deskriptiven Kommunikationsmodell handelt es sich um ein Abbild der gegenwärtigen Situation und Kommunikation. Dieses Modell kann mit einem Foto verglichen werden, dass etwas schon Vorhandenes in der Welt darstellt.

Wie man in der Abbildung 3. erkennen kann ist der Austausch von Informationen ausschließlich Synchron. Nur ein geringer Teil der Kommunikation erfolgt zwischen Nutzer und der API's. Meist ist der Nutzer mit sich selbst beschäftigt. Dies sorgt dafür, dass der Nutzer auf vieles selbst achten muss und eigenständig planen und prüfen muss, um sein Ziel zu erreichen.

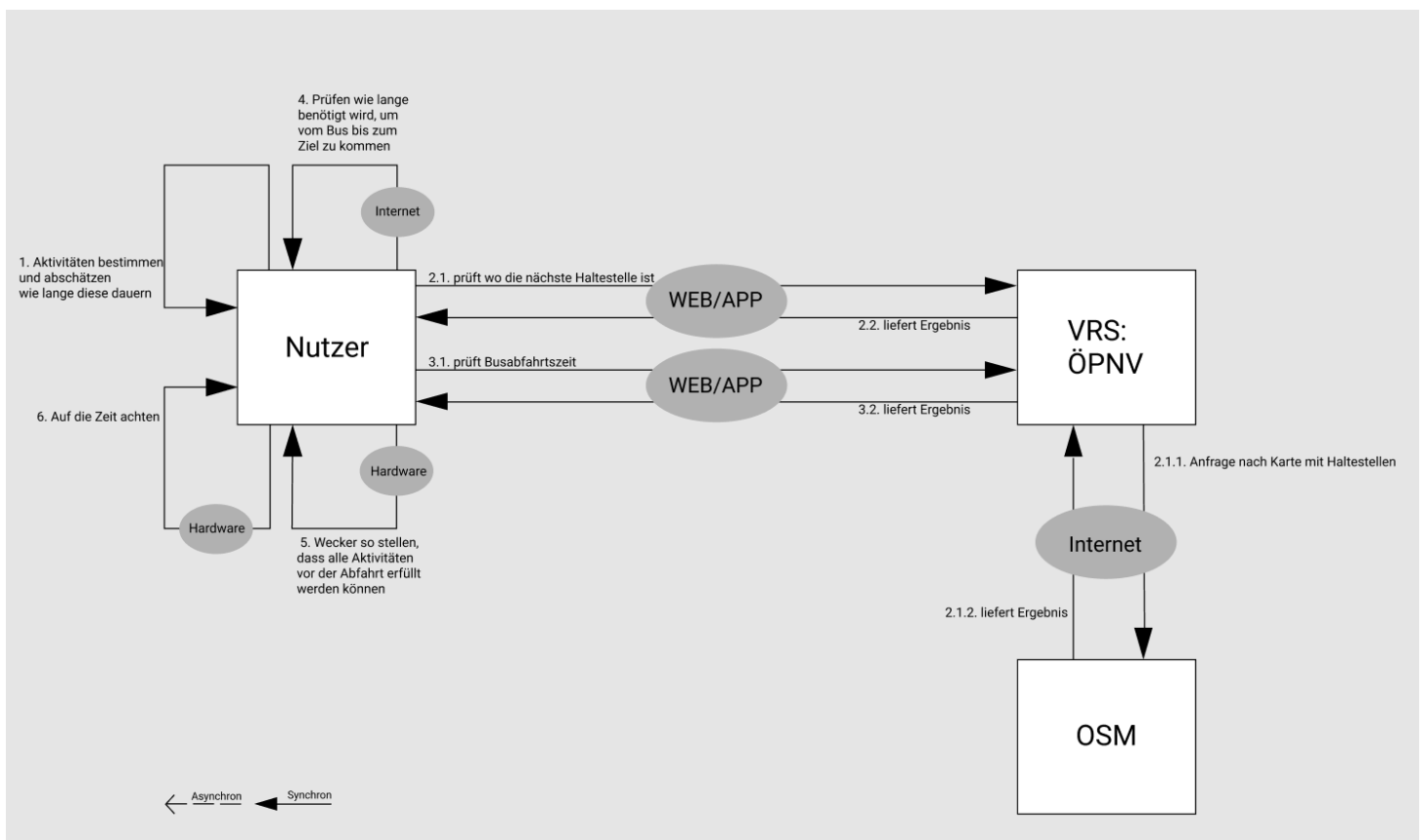


Abbildung 12: Deskriptives Kommunikationsmodell



## 14.2 Präskriptive Kommunikationsmodell

Beim Präskriptiven Kommunikationsmodell wird eine Form dargestellt wie die Kommunikation eigentlich sein sollte. Es ist ein Modell das als ein Bauplan fungiert und erstellt wird um es später zu realisieren und entwickeln.

Aus der Abbildung 4. kann man entnehmen, dass der Nutzer durch das Produkt „InTime“ sehr entlastet wird. Hier regelt das Produkt die Zeit und die Aktivitäten. Auch die Kommunikationen der APIs muss nicht mehr vom Nutzer über das Web oder einer einfachen Auskunft-Applikation erfolgen. Das Produkt übernimmt dies und hält den Benutzer mittels asynchronen Benachrichtigungen auf dem laufenden.

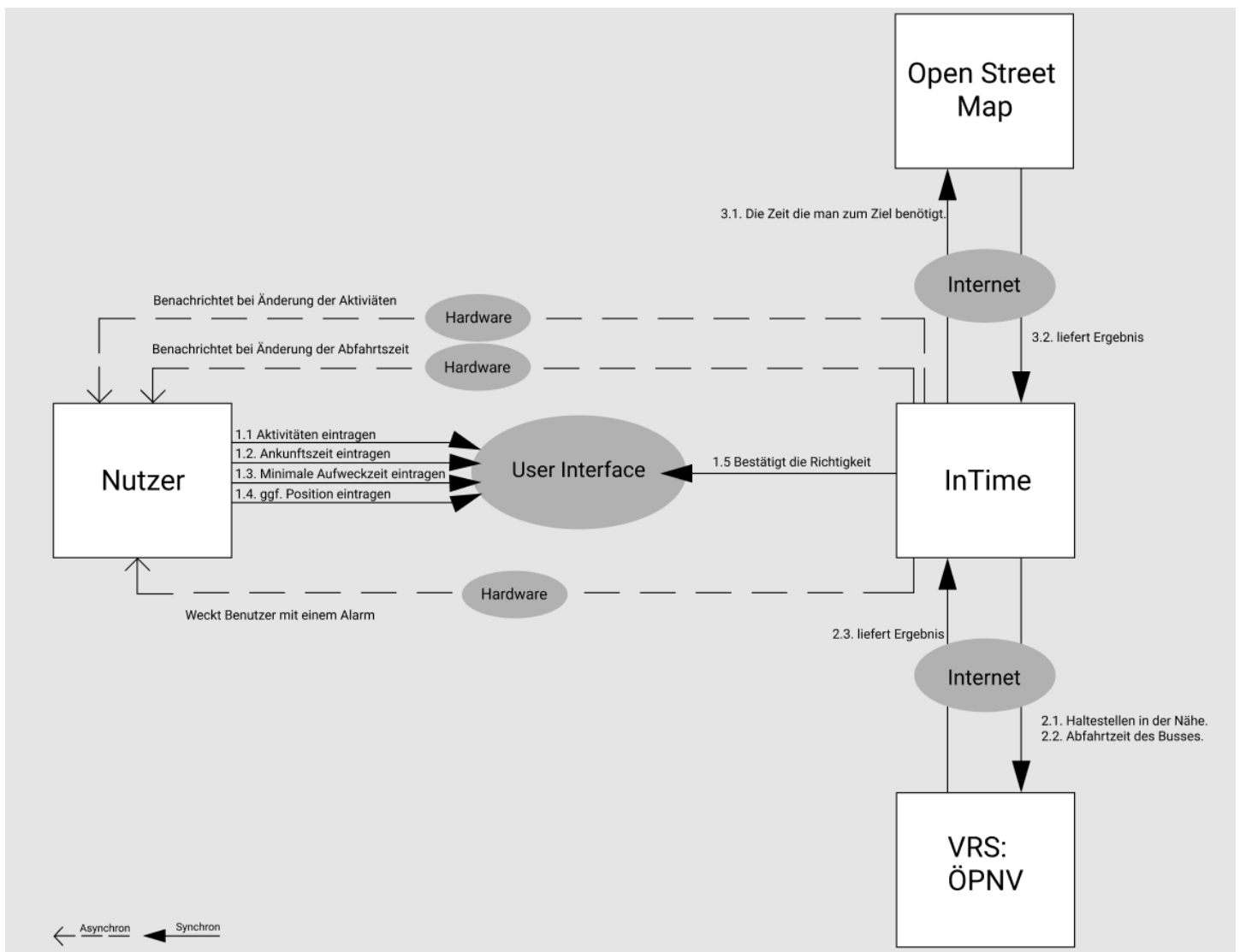


Abbildung 13: Präskriptives Kommunikationsmodell

---

## 15. Architekturdiagramm

Zur Erstellung des Architekturdiagramms galt es die Kommunikation der unterschiedlichen Softwarekomponenten zu modellieren, zu denen zusätzlich die Benennung der jeweiligen Kommunikationsprinzipien und Protokolle notwendig ist. Des Weiteren ist es erforderlich die Repräsentation der Netzwerkdaten klar aufzuzeigen, während außerdem die Verteiltheit des Systems deutlich zu erkennen ist.

Zuerst ist zu erwähnen, dass die Trennung des Systems in einen Dienstgeber und einen Dienstinutzer den Zweck verfolgt, sowohl dem letztendlichen Endgerät als auch dem API-Anbieter „VRS“ eine markante Menge Rechenarbeit abzunehmen. Dies hat zur Folge, dass zum einen im Optimalfall der Benutzer keine langen Ladezeiten erdulden muss und zum anderen wir uns damit an die vom API-Anbieter gestellten Nutzungsvereinbarungen halten. Denn diese besagen Sinngemäß unter anderem, dass das von uns entwickelte System zu keinen signifikanten Performanceeinbrüchen seitens der API-Server führen darf, beziehungsweise sollte. Um diesem Punkte so gut wie möglich gerecht zu werden, entschlossen wir uns, die verschiedenen Nutzer unseres Systems sich nur so selten wie möglich mit der API der VRS verbinden zu lassen.

So schickt der Dienstinutzer nur zu Beginn der Zeitplanerstellung eine einmalige Anfrage an die API, wodurch dieser in der Lage ist, die für den Benutzer benötigten Zeitplan korrekt zu erstellen. Da es bei öffentlichen Verkehrsmitteln allerdings zu Verspätungen und Ausfällen kommen kann, wäre es dem Dienstinutzer notwendig in regelmäßigen Abständen eine Anfrage an die API zu stellen, also „polling“ zu betreiben. Dies hat allerdings den großen Nachteil, dass es bei einer großen Nutzermenge zu einer enormen Anfragemenge pro Minute kommen kann. Um dieses Problem zu lösen, entschlossen wir uns, dem Dienstgeber die Aufgabe zuzuteilen sich um die Verspätungen der Verkehrsmittel zu kümmern. So fragt der Dienstgeber nach jetzigem Stand gemäß einem gewählten Zeitintervall die Daten zur Bestimmung von Verspätungen ab und stellt diese den Dienstinutzern bereit, welche sich dadurch für die Aktualisierung von Abfahrtszeiten ausschließlich an den Dienstgeber richten. Da damit das eingegangene Problem allerdings nur verschoben und nicht behoben ist, entschlossen wir uns zudem, das Publish-Subscribe Kommunikationsprinzip zu verwenden. Unter Verwendung dessen, stellen die Dienstinutzer nicht mehr die Anfragen direkt an den Dienstgeber, sondern bekommen die für sie notwendigen Daten bei Veränderung automatisch mitgeteilt, sodass damit das Problem der möglichen Überbelastung weitestgehend aus dem Weg geräumt wurde.

Da es dem Dienstgeber jedoch notwendig ist, die Daten der VRS-API abzurufen, stellt dieser beispielsweise alle 30 Sekunden lediglich eine Anfrage an den Anbieter. Das Ergebnis dieser architektonischen Umstellung ist eine signifikant niedrigere Anfragenmenge an den API-Anbieter, denn verglichen mit dem zuvor genannten Szenario steigt die Anzahl der Anfragen pro Minute mit jedem Benutzer unseres Systems, während sich die

---

Anfragenmenge nun stetig auf einem vergleichsweise niedrigen Niveau befindet.

Zur Bestimmung der Laufdauer zu einer Bushaltestelle und dem letztendlichen Ziel ist es notwendig einen weiteren API-Anbieter, in unserem Fall OpenStreetMap, in das System einzubinden. Mit diesem wird allerdings das Gerät jedes Nutzers direkt kommunizieren, da dort nur eine niedrige Anfragenanzahl pro Verwendungsfall zu erwarten ist.

Da unser System allerdings zusätzlich in der Lage sein soll, für die Zeitplanerstellung notwendige Daten selbstständig annähern und ermitteln zu können, umfasst die Kommunikation unter den Systemkomponenten außerdem den Kommunikationskanal zwischen Dienstanutzer und Dienstgeber.

In diesem werden bei der initialen Nutzung eines Endgerätes einmalig Daten vom Dienstgeber erfragt, während im späteren Verlauf die Kommunikation überwiegend vom Dienstanutzer ausgehend stattfindet. Aus dem Grund, dass der Dienstgeber über „predictive analytics“ notwendige Daten ermittelt, ist es erstrebenswert möglichst viele Einflussnehmende Daten in die Erstellung von Vorhersagen einfließen zu lassen. Um dies zu gewährleisten schickt der Dienstanutzer in einem zu bestimmenden Intervall die erforderlichen Daten an den Dienstgeber, sodass dieser sie für die Berechnung aufgreifen und verwenden kann.

Mit Ausnahme des bereits genannten Publish-Subscribe Prinzips findet die Übermittlung von Daten stets in synchroner Art statt. Dies hat den Grund, dass die verwendeten API-Anbieter lediglich eine solche synchrone Kommunikation anbieten, was allerdings auch den Vorteil hat, dass es jedem Nutzer ermöglicht, die für sich notwendigen Daten spezifisch abzufragen. Des Weiteren ist die Netzwerkdarstellung mit einer Ausnahme stets als XML definiert. Diese Verwendung liegt auch hier darin Begründet, dass die Kommunikation mit den API-Anbietern stets über XML stattfindet. Sowohl Anfragen als auch Antworten werden in diesem Format dargestellt. Da beide API-Anbieter ein XML-Schema bereit stellen ist es dadurch möglich alle einkommenden und ausgehenden Netzwerkdaten überprüfen zu können um sicher zu stellen, dass keine Fehlerhaften oder falschen Daten von uns abgeschickt oder zur Verarbeitung herangezogen werden.

In der zuvor genannten Ausnahme erfolgt die Kommunikation zwischen Dienstanutzer und Dienstgeber mittels der JSON-Darstellung. Aus dem Grund, dass die Überprüfung der Daten unter verschiedenen Systemen entfällt, wurde sich an dieser Stelle aufgrund der übersichtlichen Notation und der Unterstützung von zahlreichen Programmiersprachen für JSON entschieden. Dadurch, dass der Dienstanutzer in unserem System in Java und der Dienstgeber in JavaScript realisiert wird, wird, aufgrund des letzteren Aspektes, der JSON-Darstellung in diesem Kommunikationsfall ein besonders hoher Stellenwert bemessen.

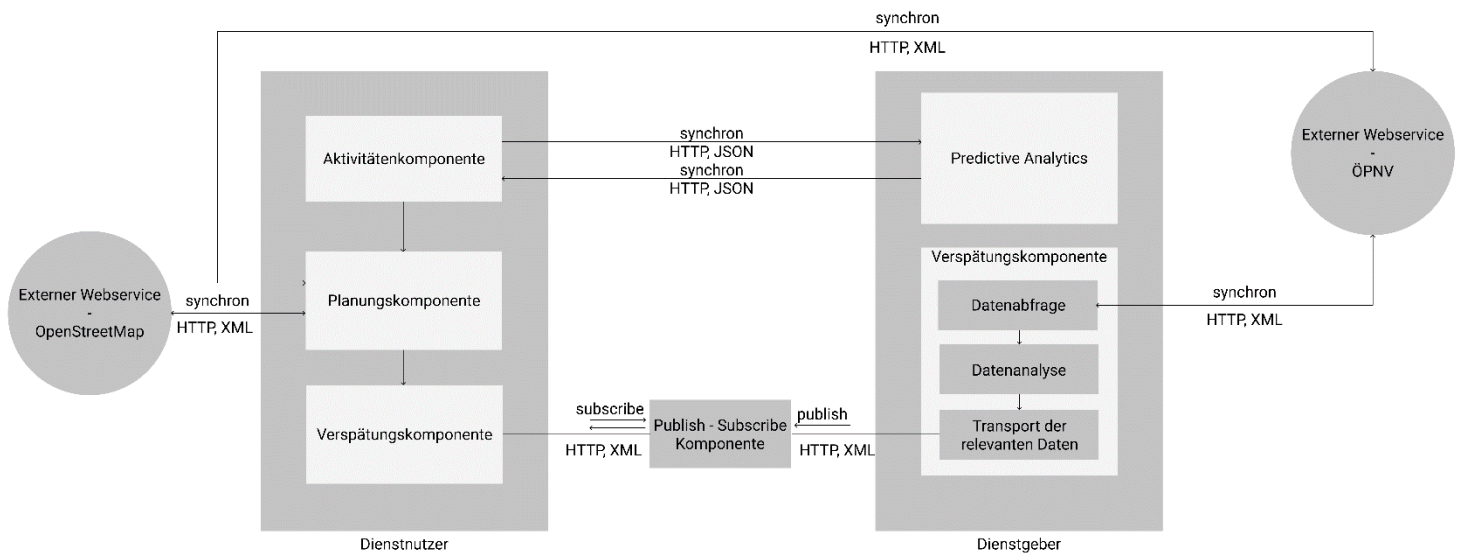


Abbildung 14: Architektordiagramm

---

## 16. Risiken

Mit der Berücksichtigung auf das Kommunikationsmodell und dem Architekturmodell lassen sich Risiken eindeutig erkennen. Die Aufzählung der Risiken lässt Schwachstellen im System diagnostizieren. Diese verwenden wir dann für das Proof of Concept als Überprüfung und Verbesserung des Systems.

### 1. Eingabe

- 1.1. Zum eingegebenen String konnte keine Aktivität gefunden werden

### 2. Zeitberechnung

- 2.1. Abfrage der notwendigen Zeit für Aktivitäten
  - 2.1.1. Zur Datenabfrage der Aktivitäten wird vom Dienstgeber keine Antwort empfangen (Verbindung zum Internet abgebrochen, 500er Fehler, etc.)
- 2.2. Abfrage der mit der Busfahrt zusammenhängenden Daten
  - 2.2.1. ÖPNV – VRS-API
    - 2.2.1.1. Verbindung mit API nicht möglich/Verbindungsabbruch
    - 2.2.1.2. Erlangte Datenstruktur stimmt nicht mit dem erwarteten Ergebnis überein
    - 2.2.1.3. Erlangte Daten stimmen mit Erwartung überein, sind allerdings lückenhaft
    - 2.2.1.4. Passende Bushaltestelle in der Nähe konnte nicht gefunden werden
  - 2.2.2. OpenStreetMap-API
    - 2.2.2.1. Verbindung mit API nicht möglich/Verbindungsabbruch
    - 2.2.2.2. Weg zu einer bestimmten Bushaltestelle konnte nicht ermittelt werden
    - 2.2.2.3. Weg von einer bestimmten Bushaltestelle zu Ziel konnte nicht ermittelt werden
    - 2.2.2.4. Zeitberechnung von einem Standpunkt zum nächsten nicht erfolgreich (Standpunkt – Bushaltestelle, Bushaltestelle – Zielort)

### 3. Zeitplanzusammenstellung

- 3.1. Es lässt sich aus den zurzeit vorliegenden Daten kein passender Zeitplan erstellen
- 3.2. Im Zuge der zeitlichen Anpassung der Aktivitäten wird der jeweilige minimale Wert einer Aktivität unterschritten
- 3.3. Zeitliche Anpassungen der Aktivitäten nicht möglich (Fallback aus 3.1), aufgrund von mehreren Aktivitäten mit der höchsten Priorität

### 4. Ereignisverarbeitung

- 4.1. Veränderung externer Einflussfaktoren
  - 4.1.1. Busverspätung tritt ein
  - 4.1.2. Busverspätung tritt ein und führt zu einem 3.1
  - 4.1.3. Busausfall tritt ein

- 
- 4.1.4. Es kann keine Alternative zu einem ausgefallenen Bus gefunden werden
  - 4.1.5. Die regelmäßig vom Dienstgeber ausgehende Abfrage von veränderten Abfahrtszeiten führt zu keiner Antwort.
  - 4.2. Veränderung interner Daten durch vom Nutzer ausgehende Eingabe
    - 4.2.1. Geforderte zusätzliche Zeit führt zu einem 3.1
  - 5. Bearbeitungsdauer von Aktivitäten abspeichern**
    - 5.1. Dienstgeber nicht erreichbar (Verbindung zum Internet abgebrochen, 500er Fehler, etc.)
    - 5.2. Keine Aktivität, in den die entsprechende Zeit einfließen könnte wurde gefunden
  - 6. Allgemeines**
    - 6.1. GPS-Daten des Nutzers nicht nutzbar

## **Risiken unter Berücksichtigung des Haushaltsmodus**

- 1. Eingabe**
  - 1.1. ...
- 2. Zeitberechnung**
  - 2.1. Abfrage der notwendigen Zeit für Aktivitäten
    - 2.1.1. ...
  - 2.2. Abfrage der mit der Busfahrt zusammenhängenden Daten
    - 2.2.1. ÖPNV – VRS-API
      - 2.2.1.1. ...
    - 2.2.2. OpenStreetMap-API
      - 2.2.2.1. ...
- 3. Zeitplanzusammenstellung**
  - 3.1. ...
  - 3.2. ...
  - 3.3. Nutzung einer begrenzten Ressource (Raum, Posten, etc.) nicht möglich, da durchgängig besetzt
- 4. Ereignisverarbeitung**
  - 4.1. Veränderung externer Einflussfaktoren
    - 4.1.1. ...
  - 4.2. Veränderung interner Daten durch vom Nutzer ausgehende Eingabe
    - 4.2.1. ...
    - 4.2.2. Zusätzlich geforderte Zeit bei Benutzung einer begrenzten Ressource führt zu einem 3.1 der dadurch betroffenen Benutzer
- 5. Bearbeitungsdauer von Aktivitäten abspeichern**
  - 5.1. ...
- 6. Allgemeines**
  - 6.1. ...
  - 6.2. Zur Nutzung des Haushaltsmodus angegebene Nutzernamen (NutzerIDs) konnten nicht gefunden werden

## **7. Sonstiges Haushaltsspezifisches**

- 7.1. Erstmalige Einordnung mehrerer Nutzer in einen Haushalt
  - 7.1.1. Dienstgeber nicht erreichbar (Verbindungsabbruch zum Internet, 500er Fehler, etc.)
  - 7.1.2. Ein Nutzer des zu bildenden Haushaltes ist bereits einem anderen Haushalt zugewiesen
- 7.2. Nachträgliche Änderung eines Haushalts
  - 7.2.1. Zu betretender Haushalt wurde nicht gefunden
  - 7.2.2. Bei Verlassen eines Haushaltes entsteht ein Haushalt mit einer Person, was dazu führt, dass die Nutzung eines Haushaltsmodus für den betroffenen Nutzer obsolet ist
  - 7.2.3. Nutzer ist bereits einem Haushalt eingeordnet, möchte aber einen anderen beitreten
  - 7.2.4. Nutzer verlässt einen Haushalt, das System ist aber noch im Haushaltsmodus (obsoleter Funktion)

## **Risiken im Projektablauf**

### **a. Androidprogrammierung**

- a. Das neue Feld der Androidprogrammierung könnte eine zu große Wissensaufbereitung erfordern, sodass wichtige Punkte nicht programmiert werden könnten.
- b. Durch fehlende Vorkenntnisse könnten einige Aufwandsschätzungen weit vom realistischen Maße entfernt sein

### **b. Wissenslücken**

- a. Die Wissensaufbereitung von WBA und MCI Inhalten nimmt nicht mehr tragbare Ausmaße an, sodass die Qualität der erarbeitenden Artefakte zu leiden hat.

### **c. Allgemeines**

- a. Sollte sich im Nachhinein herausstellen, dass vorher getroffene Planungsinhalte mangelhaft, bzw. unzutreffend sein, könnte ein erhöhter Arbeitsaufwand entstehen, als zu Beginn gedacht was wiederum zu einem c.b. führt.
- b. Geplante Features können aufgrund von Zeitmangel nicht umgesetzt werden (bspw. Haushaltsmodus)
- c. Arbeitsmittel (Laptop, Software, o.ä.) sind im Arbeitszeitraum nicht mehr nutzbar
- d. Aufgrund von externen, unvorhersehbaren Ereignissen könnte es dazu kommen, dass der initiale Zeitplan nicht eingehalten werden kann

## 17. REST Modellierung

Tabelle 18: REST Modellierung

| Ressource           | Methode | Semantik   | Request Content-Type | Response Content-Type |
|---------------------|---------|--|----------------------|-----------------------|
| /household          | POST    | Sich einem Haushalt zuweisen   | Application/json     | Application/json      |
| /household{:id}     | PUT     | Die Informationen des eigenen Haushaltes ändern  | Application/json     | Application/json      |
| /household          | GET     | Die Informationen des eigenen Haushaltes anzeigen  | -                    | Application/json      |
| /activity           | POST    | Eine neue Aktivität in die interne Datenbank einführen   | Application/json     | Application/json      |
| /activity{:id}      | GET     | Die Informationen einer Aktivität erfragen   | -                    | Application/json      |
| /activity{:id}      | PUT     | Die Daten einer Aktivität ändern   | Application/json     | Application/json      |
| /transport?location | GET     | Die notwendigen Daten erfragen, um mit einem öffentlichen Verkehrsmittel zu dem Ziel zu gelangen | -                    | Application/xml       |
| /transport{:id}     | GET     | Die Informationen einer expliziten Fahrt erfragen  | -                    | Application/xml       |
| /walkTime?location  | GET     | Die (Lauf)dauer zu einem Ort abfragen  | -                    | Application/xml       |



---

## 18. Proof of Concept

### **Risiko 3.1 Zeitplanerstellung nicht möglich**

Beschreibung: Die vom Nutzer angestrebten Aktivitäten nehmen als kollektiv mehr Zeit in Anspruch, als dem Nutzer bis zur Deadline zur Verfügung steht, weswegen ein passender Zeitplan mit den vorliegenden Daten nicht erstellt werden kann.

Exit-Kriterium: Die Gesamtdauer der Aktivitäten unterschreitet die verfügbare Zeit bis zur Deadline, sodass ein Zeitplan erstellt werden kann in dem alle gewünschten Aktivitäten untergebracht werden können.

Fail-Kriterium: Die Gesamtdauer der Aktivitäten überschreitet die verfügbare Zeit bis zu Deadline, sodass mit den momentan vorliegenden Daten kein passender Zeitplan erstellt werden kann.

Fallback: Der Nutzer hat nun die Möglichkeit die momentan vorliegende Bearbeitungsdauer der geplanten Aktivitäten manuell anzupassen oder einzelne Aktivitäten zu entfernen. Alternativ ist der Nutzer auch in der Lage diese zeitliche Anpassung automatisch geschehen zu lassen. Das System vermindert in diesem Fall entsprechend des Prioritätswertes die Bearbeitungsdauer der einzelnen Aktivitäten, bis der Zeitplan eingehalten werden kann. Je niedriger dabei die Priorität einer Aktivität ist, desto mehr Zeit wird von dieser entfernt.

### **Risiko 3.2 Zeitliche Verminderung der Aktivitätsbearbeitungsdauer nicht möglich**

Beschreibung: Im Falle einer automatischen zeitlichen Anpassung der Bearbeitungsdauer der Aktivitäten kommt es zu dem Problem, dass einer Tätigkeit ein zu niedriger Wert zugewiesen wird, wie beispielsweise ein Zeitraum von einer Minute für das Frühstück. Richtwert ist dabei ein prozentualer Wert jeder Aktivität, sodass beispielsweise die Bearbeitungsdauer nicht um mehr als 30% des Initialwertes erniedrigt werden darf.

Exit-Kriterium: Alle Aktivitäten unterschreiten nicht die jeweilige minimale Bearbeitungsdauer.

Fail-Kriterium: Eine oder mehrere Aktivitäten würden im Zuge der zeitlichen Anpassung die jeweilige minimale Bearbeitungsdauer unterschreiten, sodass kein passender und realistisch umsetzbarer Zeitplan erstellt werden kann.

Fallback: Der Nutzer hat die Möglichkeit die Bearbeitungsdauer von Aktivitäten manuell anzupassen, worauf keine Überprüfung auf Einhalten eines Minimalwertes stattfindet. Des Weiteren ist der Nutzer in der Lage, Aktivitäten aus dem Zeitplan zu entfernen. Die angegebenen Daten werden für den weiteren Systemablauf verwendet.

### **Risiko 3.3 Zeitliche Anpassung nicht möglich**

Beschreibung: Im Falle einer automatischen zeitlichen Anpassung der Bearbeitungsdauer der Aktivitäten kommt es zu dem Problem, dass alle vorhandenen Tätigkeiten die höchste Priorität aufweisen, was zur Folge hat, dass die Zeit keiner Aktivität vermindert werden kann.

Exit-Kriterium: Die zeitliche Anpassung der entsprechenden Aktivitäten ist erfolgreich und die Bearbeitungsdauer einer Tätigkeit unterschreitet nicht den minimalen relativen Wert und eine Anpassung durch den Nutzer ist nicht notwendig.

Fail-Kriterium: Die vorhandenen Aktivitäten der höchsten Prioritätsstufe überschreiten additiv die verfügbare Zeit bis zur absoluten Deadline, weswegen eine automatische zeitliche Anpassung dieser nicht möglich ist.

Fallback: Der Nutzer hat die Möglichkeit die Dauer der Aktivitäten manuell anzupassen oder Aktivitäten zu entfernen. Bei Bestätigung der Daten werden die Eingaben für den weiteren Systemablauf verwendet.

## 19. UI Prototyp

Um die Benutzerschnittstellen der einzelnen Systemkomponenten nun möglichst repräsentativ abbilden zu können, wurden von uns grafische Darstellungen der Schlüsselemente erarbeitet.

Begonnen wurde dabei mit der Entwicklung von groben Wireframes mit Stift und Papier, sodass sich beide Parteien des Teams auf die grundlegenden Größen, Elemente und Verteilungen dieser einigen konnten.

### 19.1 Maße

In Folge der Erstellung der Wireframes wurden diese bisher groben Entwürfe verfeinert, indem gleichbleibende Größen und Abstände festgelegt wurden.

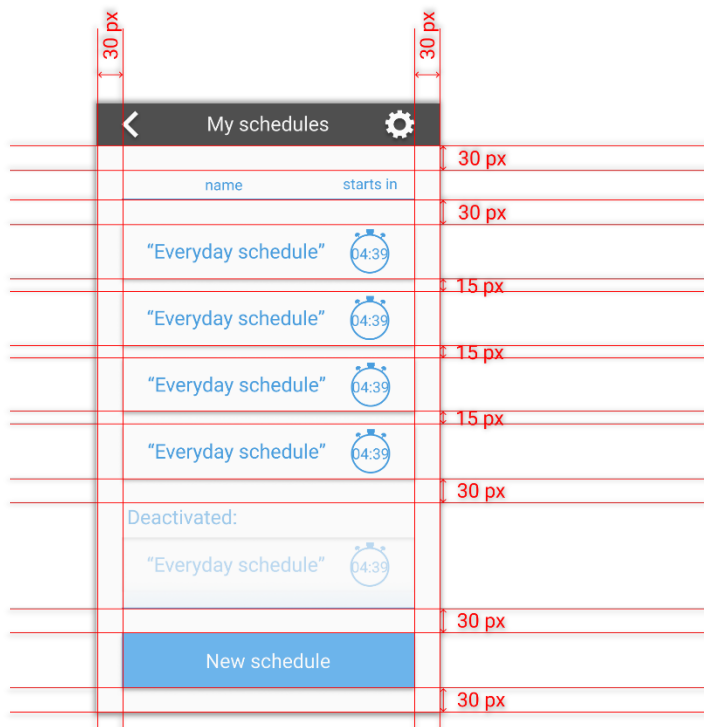


Abbildung 15: UI Prototyp, My schedules Teil 1

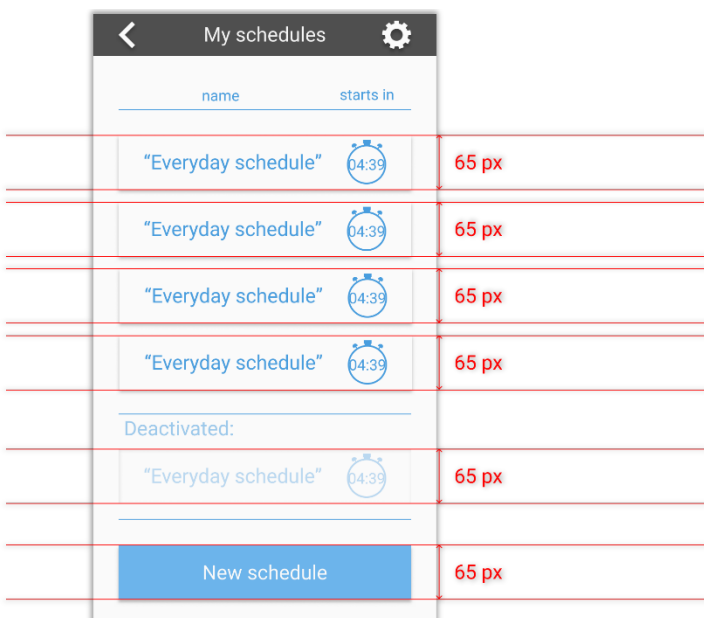


Abbildung 16: UI Prototyp, My schedules Teil 2

Wie in Abbildung 15 zu erkennen, entschieden wir uns sowohl bei Abständen zum Rand des Bildschirms sowie bei Entfernungen zu anderen Elementgruppen innerhalb des Screens einen Abstand von 30px zu wählen. Dieses Maß wird auch bei allen weiteren Screens eingehalten, sodass eine gewisse Konsistenz entsteht. Bei Abständen zwischen Elementen, die derselben Gruppe angehören, wird dagegen ein Abstand von 15px gewählt. Wichtig dabei ist, dass dieser Zwischenraum erkennbar kleiner ist als jener, der für die Trennung von unterschiedlichen Elementgruppen verwendet wird. Dadurch werden diese vom Nutzer, gemäß dem Gesetz der Nähe (vgl. [17]), als Zusammengehörig anerkannt, wodurch wiederum die Struktur der Benutzerschnittstelle und die Navigation durch diese verbessert, bzw. erleichtert wird. Auch bei den primären Interaktionsobjekten wurde durchgängig dieselbe Größe (65px) verwendet (siehe Abbildung 16) Hierbei ist

zusätzlich Wert darauf zu legen, dass der Nutzer zur Eingabe überwiegend den eigenen Daumen nutzt und nicht die schmalere Finger. Dies hat zur Folge, dass die Interaktionsobjekte eine entsprechend größere Breite aufweisen sollten, weswegen wir dies auch in unserem Entwurf beherzigten.

## **19.2 Positionierung der Interaktionsobjekte**

Die wichtigsten Interaktionsobjekte wurden von uns an möglichst leicht erreichbaren Bereichen platziert, sodass der Weg des Fingers, bzw. des Daumens zu eben jenen Schlüsselobjekten möglichst gering ausfällt. Dies hat den Vorteil, dass unser System dadurch entsprechend einfacher, schneller und angenehmer zu nutzen ist. Verweisen ist dabei darauf auf Fitts' Gesetz, welches besagt, dass die benötigte Zeit um einen Zielbereich zu erreichen abhängig von der Distanz zu der Fläche und der Größe dessen ist. (vgl. [18])

Wie bereits im vorhergegangenen Kapitel genannt, weisen die Schlüsselobjekte unseres Entwurfes eine erhöhte Größe auf, sodass das Gesetz auch dahingehend beachtet wird, ein möglichst optimales Endergebnis zu erreichen.

Die Position dieser Interaktionsobjekte ist außerdem auf jedem Screen dieselbe, sodass der Nutzer zusätzlich unterbewusst verinnerlichen kann, dass sich das Zielobjekt zur Fortführung des Anwendungsprozesses stets an dem selben Ort befindet.

Des Weiteren wurden sämtliche Objekte, die eine erhöhte Priorität aufweisen, stets mit einem Schatten hinterlegt, sodass sich diese nicht nur farblich von dem restlichen Inhalt des Screens unterscheiden, sondern auch indem dem Nutzer suggeriert wird, dass sich diese Buttons weiter im Vordergrund befinden würden.

## **19.3 Farbgebung**

Aus dem anschließenden Selbststudium der Farblehre ging heraus, dass Cyan Bewusstsein und Klarheit und Blau Vertrauen und Pflichttreue vermittelt. (vgl. [19]) Diese Umstände führten uns zu der Entscheidung, mit einem hellen Blau einen Mittelweg zu wählen, sodass bei bloßer Betrachtung unserer UI der Nutzer möglichst alle der genannten Aspekte mit unserem System verbindet. Da „InTime“ darauf abzielt, auch wichtige und kritische Momente im alltäglichen Leben zu lösen, sind die Vermittlung eben jener Eigenschaften unabdingbar, sodass der Nutzer dadurch bestärkt wird, das System besten Gewissens nutzen zu können.

Um neben der Primärfarbe auch die Sekundärfarben zu wählen, entschieden wir uns für die Nutzung eines Webtools, der unter Angabe einer Hauptfarbe auch mögliche weitere Farben angibt. [20]



Abbildung 17: Primär- und Sekundärfarben

Das Ergebnis dieses Arbeitsschrittes ist in Abbildung 17 zu sehen. Nun jedoch war es noch notwendig eine Hintergrundfarbe zu wählen, wo wir auch hier erneut auf die Farblehre zurückgriffen und uns für Weiß entschieden. Diese besitzt, neben der Primärfarbe, auch die Eigenschaft, dass sie Klarheit und außerdem Reinheit vermittelt, was wie bereits erläutert, dem Aufbau von Vertrauen zu unserem System entgegenkommt.

## 19.4 Farbgebung der Interaktionsobjekte

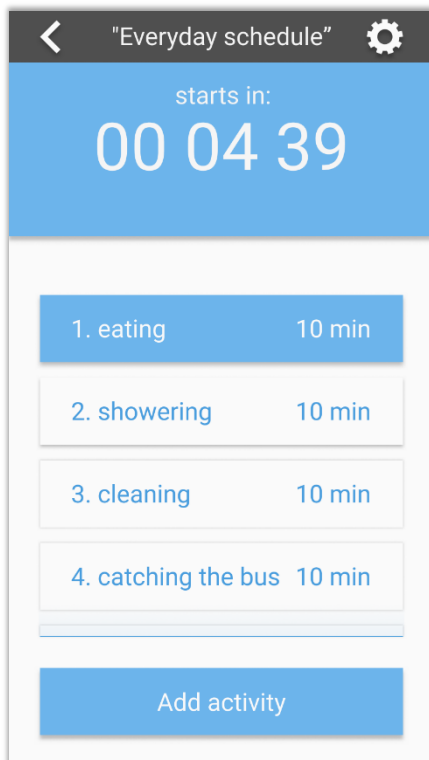


Abbildung 18: UI Prototyp, Everyday schedule

Um die soeben erarbeiteten Farben möglichst optimal in unseren bisherigen Grobentwurf einzubringen, folgte daraufhin der Schritt die Auswahl jener Interaktionsobjekte die eine erhöhte Priorität aufweisen und somit eine gesonderte Farbgebung aufweisen müssen. Diese als von uns wichtig klassifizierten Objekte sind zum einen solche, die zur Verknüpfung des Programmablaufes notwendig sind und zum anderen jene, die der Nutzer möglichst zu beachten hat. Zu letzterem zählen Beispielsweise die in Abbildung 18 gezeigte „eating“ Aktivität, da diese die erste Tätigkeit ist, die der Nutzer durchzuführen hat und somit sicherheitshalber darüber informiert wird. Jene Interaktionsobjekte, die zum Verlauf des Projektablaufes notwendig sind, sind unter anderem der in Abbildung 18 gezeigte „Add activity“ Button. Dieser erfüllt wie die restlichen dieser Einordnungsgruppe eine Schlüsselfunktion und ist somit nach unserer Definition dafür prädestiniert mit der Primärfarbe versehen zu werden. Der Grund,

weswegen die wichtigsten Objekte mit dieser Farbe versehen werden, ist jener, dass diese den höchsten Kontrast aufweisen und somit dem Nutzer als hervorstehend wahrgenommen werden.

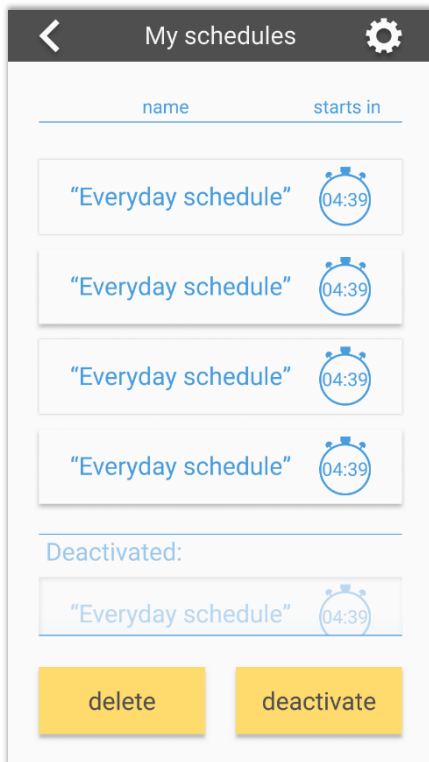


Abbildung 19: UI Prototyp, My schedules  
Teil 3

Die Sekundären Interaktionsobjekte wiederum werden von uns auch mit der entsprechenden Nebenfarbe belegt. Diese erfüllen in unserem System den Zweck der Warnung, da die darauf liegenden Elemente zusätzliche eine schwarze Farbe aufweisen. Diese Kombination von gelb und schwarz wird häufig aus Erfahrungen mit anderen Softwaresystemen mit Warnungen und kritischen Bereichen verbunden, was dem eben genannten hauptsächlichen Verwendungszweck entspricht. (siehe Abbildung 19)

---

## 20. Verzeichnisse

### 20.1 Literaturverzeichnis

- [1] Peter Kairies: *So analysieren Sie ihre Konkurrenz*, (2013), Auflage 9, S.20
- [2] „Google Now - Wikipedia“, unter: [https://de.wikipedia.org/wiki/Google\\_Now](https://de.wikipedia.org/wiki/Google_Now) (abgerufen am 29.10.17).
- [3] „Google Now – Official Page“, unter: <https://www.google.com/intl/de/landing/now/> (abgerufen am 29.10.17).
- [4] Marco Engelen (02. März 2016), „Hound ist Siri und Google Now auf Steroiden“, unter: <https://curved.de/news/hound-ist-siri-und-google-now-auf-steroiden-354672> (abgerufen am 29.10.17).
- [5] „Hound – SoundHound Inc.“, unter: <https://soundhound.com/hound> (abgerufen am 29.10.17).
- [6] „Siri oder Alexa“, unter: <https://www.vorteile-nachteile.info/leben/siri-oder-alexa-1/> (abgerufen am 29.10.17).
- [7] „Das neue Amazon Echo“, unter: <https://www.amazon.de/Amazon-Echo-2nd-Generation-Anthrax-Stoff-/dp/B06ZXQV6P8> (abgerufen am 29.10.17).
- [8] Charles Nyce, „Predictive Analytics White Paper“, unter: <https://www.the-digital-insurer.com/wp-content/uploads/2013/12/78-Predictive-Modeling-White-Paper.pdf>, S.1 (zuletzt abgerufen am 01.11.17).
- [9] Heather McIlvaine (21.03.2013), „Was ist eigentlich Predictive Analytics“, unter: <https://news.sap.com/germany/was-ist-eigentlich-predictive-analytics/> (zuletzt abgerufen am 01.11.17).
- [10] „What is Predictive Analytics“, unter: <https://www.predictiveanalyticstoday.com/what-is-predictive-analytics/> (abgerufen am 01.11.17)
- [11] „Entwicklerleidfaden“, unter: [https://cloud.google.com/prediction/docs/developer-guide#what\\_is\\_the\\_prediction\\_api](https://cloud.google.com/prediction/docs/developer-guide#what_is_the_prediction_api) (zuletzt abgerufen am 01.11.17)
- [12] „Cloud Machine Learning Engine“, unter: <https://cloud.google.com/ml-engine/> (zuletzt abgerufen am 01.11.17)
- [13] „ÖPNV-Daten für Webentwickler“, unter: <https://www.vrsinfo.de/fahrplan/oepnv-daten-fuer-webentwickler.html> (zuletzt abgerufen am 13.11.17)

- 
- [14] „Using OpenStreetMap“, unter: [http://wiki.openstreetmap.org/wiki/Using\\_OpenStreetMap#Software\\_development](http://wiki.openstreetmap.org/wiki/Using_OpenStreetMap#Software_development) (zuletzt abgerufen am 13.11.2017)
- [15] „API v0.6“, unter: [http://wiki.openstreetmap.org/wiki/API\\_v0.6](http://wiki.openstreetmap.org/wiki/API_v0.6) (zuletzt abgerufen am 03.12.2017)
- [16] „InTime“ Projektplan, unter: [https://github.com/sgeier96/EISWS1718DemizkiGeier/blob/master/MS1/DemizkiGeier\\_ProjektplanEISWS1718.pdf](https://github.com/sgeier96/EISWS1718DemizkiGeier/blob/master/MS1/DemizkiGeier_ProjektplanEISWS1718.pdf) (zuletzt abgerufen am 12.11.17)
- [17] Max Wertheimer, „Untersuchung zur Lehre von der Gestalt“, unter: [http://gestalttheory.net/download/Wertheimer1923\\_Lehre\\_von\\_der\\_Gestalt.pdf](http://gestalttheory.net/download/Wertheimer1923_Lehre_von_der_Gestalt.pdf), S. 9 (zuletzt abgerufen am 20.12.2017)
- [18] Kevin Hale, „Visualizing Fitts’s Law“, unter: <http://particletree.com/features/visualizing-fittss-law/> (zuletzt abgerufen am 20.12.2017)
- [19] Prof. Dr. Christian Noss, „Gestaltungselemente“, unter: [https://www.medieninformatik.th-koeln.de/wiki/images/2/2e/030\\_gestaltungselemente.pdf](https://www.medieninformatik.th-koeln.de/wiki/images/2/2e/030_gestaltungselemente.pdf), S. 179 & 180 (zuletzt abgerufen am 21.12.2017)
- [20] „Paletton – The Color Scheme Designer“, unter: <http://paletton.com> (zuletzt abgerufen am 20.12.2017)



**20.2 Abbildungsverzeichnis**

|  |           |
|--|-----------|
| <i>Abbildung 1. Das für unser Vorhaben angepasste Wasserfallmodell</i> | <i>10</i> |
| <i>Abbildung 3: Stakeholder Teil 1</i>                                 | <i>18</i> |
| <i>Abbildung 4: Stakeholder Teil 2</i>                                 | <i>19</i> |
| <i>Abbildung 5: Persona Paul Bauer</i>                                 | <i>25</i> |
| <i>Abbildung 6: Persona Anna Bauer</i>                                 | <i>26</i> |
| <i>Abbildung 7: Persona Andreas Stiel</i>                              | <i>27</i> |
| <i>Abbildung 8: Persona Lena Stiel</i>                                 | <i>28</i> |
| <i>Abbildung 9: Persona Alina Stiel</i>                                | <i>29</i> |
| <i>Abbildung 10: Persona Kai Stiel</i>                                 | <i>30</i> |
| <i>Abbildung 11: Persona Stefan Kahl</i>                               | <i>31</i> |
| <i>Abbildung 12: Persona Can Gür</i>                                   | <i>32</i> |
| <i>Abbildung 13: Deskriptives Kommunikationsmodell</i>                 | <i>40</i> |
| <i>Abbildung 14: Präskriptives Kommunikationsmodell</i>                | <i>41</i> |
| <i>Abbildung 15: Architekturdiagramm</i>                               | <i>44</i> |
| <i>Abbildung 16: UI Prototyp, My schedules Teil 1</i>                  | <i>49</i> |
| <i>Abbildung 17: UI Prototyp, My schedules Teil 2</i>                  | <i>49</i> |
| <i>Abbildung 18: Primär- und Sekundärfarben</i>                        | <i>49</i> |
| <i>Abbildung 19: UI Prototyp, Everyday schedule</i>                    | <i>49</i> |
| <i>Abbildung 20: UI Prototyp, My schedules Teil 3</i>                  | <i>49</i> |

### 20.3 Abkürzungsverzeichnis

| Abkürzung | Bedeutung   |
|-----------|---|
| API       | Application Programming Interface oder auch „Anwendungsprogrammierschnittstelle“.                     |
| UI        | User Interface oder auch „Benutzerschnittstelle“  |
| ISO       | International Organization for Standardization (auf Deutsch: Internationale Organisation für Normung) |
| DIN       | Deutsches Institut für Normung  |
| FDD       | Feature Driven Developments (Sammlung von Arbeitstechniken, Methoden)                                 |
| REST      | Representational State Transfer   |
| OSM       | Open Street Map   |
| ÖPNV      | Öffentlicher Personennahverkehr   |
| VRS       | Verkehrsverbund Rhein-Sieg  |
| XML       | Extensible Markup Language  |
| JSON      | JavaScript Object Notation  |
| http      | Hypertext Transfer Protocol   |
| GPS       | Global Positioning (auf Deutsch: Globales Positionsbestimmungssystem)                                 |
| ASS       | Auskunft Service System   |
| DDos      | Distributed Denial of Service (Verteiltes Verweigerung des Dienstes)                                  |
| XML       | Extensible Markup Language (Eine Auszeichnungssprache)  |
| URL       | Uniform Resource Locator (einheitlicher Ressourcenzeiger)   |
| POI       | Point of Interest (interessanter Ort, für Routenplanung)  |
| REST      | Representational state transfer   |
| APP       | Application (Anwendung)   |

**20.4 Tabellenverzeichnis**

|  |           |
|--|-----------|
| <i>Tabelle 1: InTimes Vor- und Nachteile</i>                   | <i>5</i>  |
| <i>Tabelle 2: Google Now's Vor- und Nachteile</i>              | <i>6</i>  |
| <i>Tabelle 3: Hound's Vor- und Nachteile</i>                   | <i>7</i>  |
| <i>Tabelle 4: Alexa's Vor- und Nachteile</i>                   | <i>7</i>  |
| <i>Tabelle 5: Zielhierarchie Teil 1</i>                        | <i>11</i> |
| <i>Tabelle 6: Zielhierarchie Teil 2</i>                        | <i>12</i> |
| <i>Tabelle 7: Zielhierarchie Teil 3</i>                        | <i>12</i> |
| <i>Tabelle 8: Zielhierarchie Teil 4</i>                        | <i>12</i> |
| <i>Tabelle 9: User Profiles (Merkmale mit Begründung)</i>      | <i>20</i> |
| <i>Tabelle 10: User Profile, Ehepartner (ohne Kinder)</i>      | <i>21</i> |
| <i>Tabelle 11: User Profile, Ehepartner (mit Kinder)</i>       | <i>22</i> |
| <i>Tabelle 12: User Profile, Schüler (Kind eines Ehepaars)</i> | <i>23</i> |
| <i>Tabelle 13: User Profile, Mitglied (Wohngemeinschaft)</i>   | <i>24</i> |
| <i>Tabelle 14: Use case #1</i>                                 | <i>34</i> |
| <i>Tabelle 15: Use Case #2</i>                                 | <i>35</i> |
| <i>Tabelle 16: Use Case #3</i>                                 | <i>36</i> |
| <i>Tabelle 17: Use Case #4</i>                                 | <i>37</i> |
| <i>Tabelle 18: REST Modellierung</i>                           | <i>48</i> |