

Technische Hochschule Köln
Fakultät 10
-
Institut für Ingenieurwissenschaften und Informatik

Konzept

„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

1. Exposé	3
2. Domänenrecherche	4
3. Konkurrenzanalyse	5
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
4. Vorgehensmodell	9
5. Projektplan	11
6. Zielhierarchie	11
6.1 Strategische, taktische und operative Ziele	13
7. Recherche zu „predictive analytics“	13
7.1 Was ist „predictive analytics“?	13
7.2 „Predictive Analytics Process“	13
7.3 API (Application Programming Interface)	14
8. User Needs	15
9. Stakeholderanalyse	16
10. Anforderungen	18
10.1 Funktionale Anforderungen	18
10.2 Nicht-funktionale Anforderungen	18
10.2.1 Produktqualität & Organisatorische Anforderungen	18
11. Kommunikationsmodell	20
11.1 Deskriptives Kommunikationsmodell	20
11.2 Präskriptives Kommunikationsmodell	21
12. Architekturdiagramm	22
13. Risiken	25
14. Proof of Concepts	28
15. Verzeichnisse	30
15.1 Literaturverzeichnis	30
15.2 Abkürzungverzeichnis	31

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

„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“

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.

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

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]).

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

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]).

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

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]).

Vorteile	Nachteile
<ul style="list-style-type: none"> • Sprachfunktion • Smart Home Geräte steuern • Einkaufsliste erstellen und direkt bestellen. • Einfache Informationsbeschaffung. • Starke und viele Mikrofone zur eindeutigen Spracheingabe • Musikkwiedergabe etc. • Taxi bestellen. 	<ul style="list-style-type: none"> • Preis • Raumgebunden • Nur eine einfache To-Do Liste die sich erstellen lässt. • Stetige Tonaufnahme über das Mikrofon, da 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

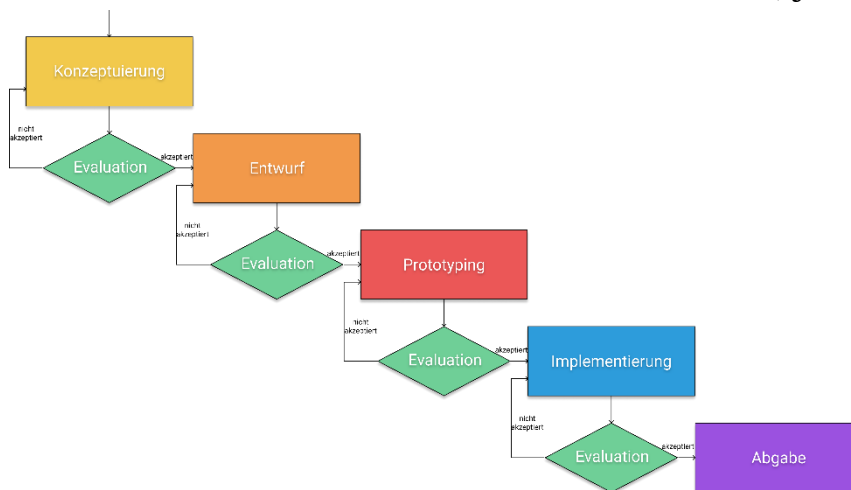


Abbildung 1. Das für unser Vorhaben angepasste Wasserfallmodell

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 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

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 [13].

6. Zielhierarchie

6.1 Strategische, Taktische und Operative Ziele

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.

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.

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

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. Recherche Predictive Analytics

7.1 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]).

7.2 „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]).

7.3 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.

8. User Needs (Erfordernisse)

Um die Erfordernisse zu benennen, muss man erst die Nutzungssituation richtig verstehen und erkennen was ein Benutzer in einer bestimmten Situation benötigt.

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 Abfahrtszeit> <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 Abfahrtszeiten> <wissen>, um <den Bus><nicht verpassen> zu können.

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

9. Stakeholderanalyse

Was sind Stakeholder?

Stakeholder sind meist mehrere Gruppen von Personen die irgendein Belangen am Produkt haben. 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 Nachkalibrierung 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/ 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

10. Anforderungen

10.1 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.

10.2 Nicht funktionale Anforderungen

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.

11. Kommunikationsmodell

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

11.1 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.

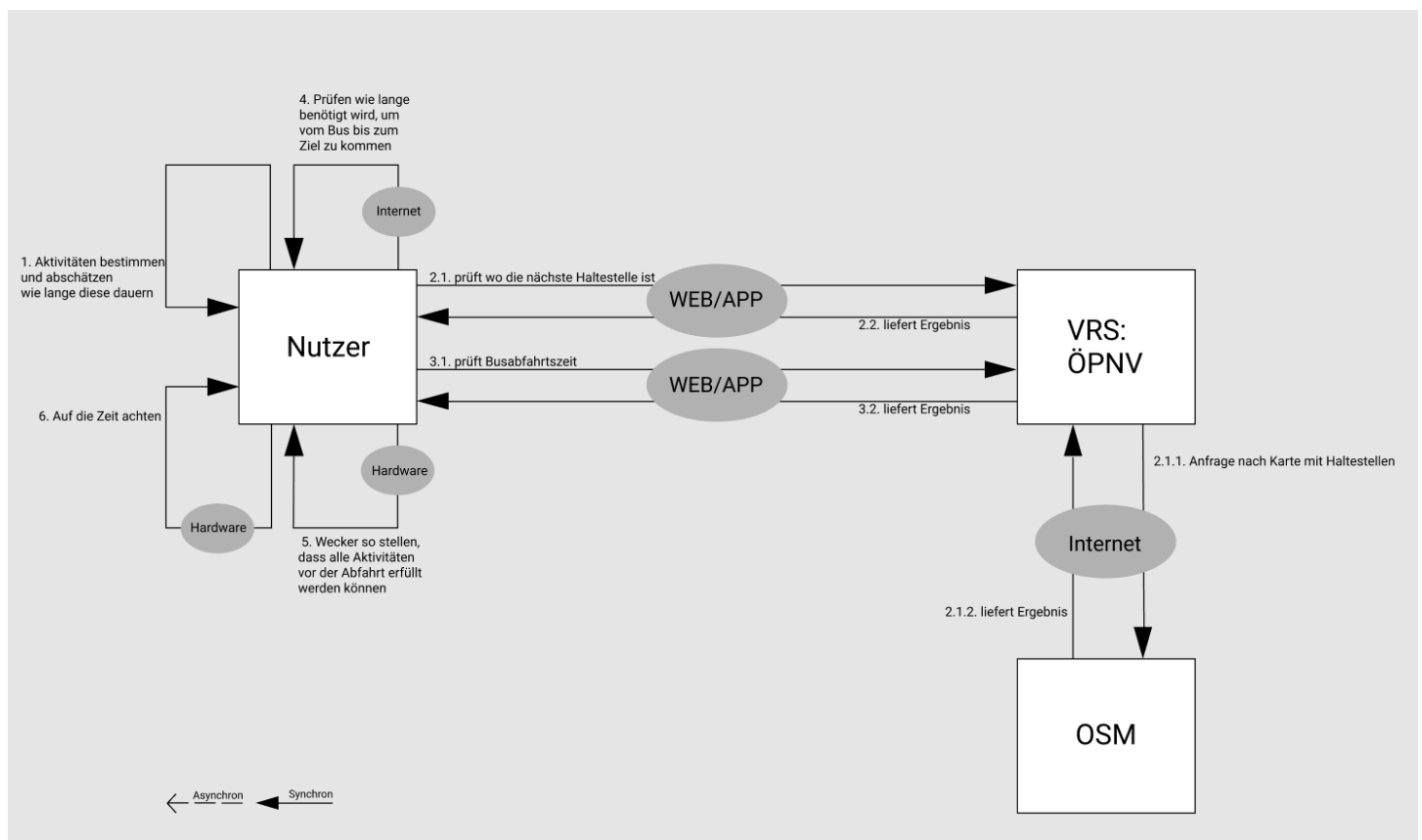


Abbildung 4: Deskriptives Kommunikationsmodell

11.2 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.

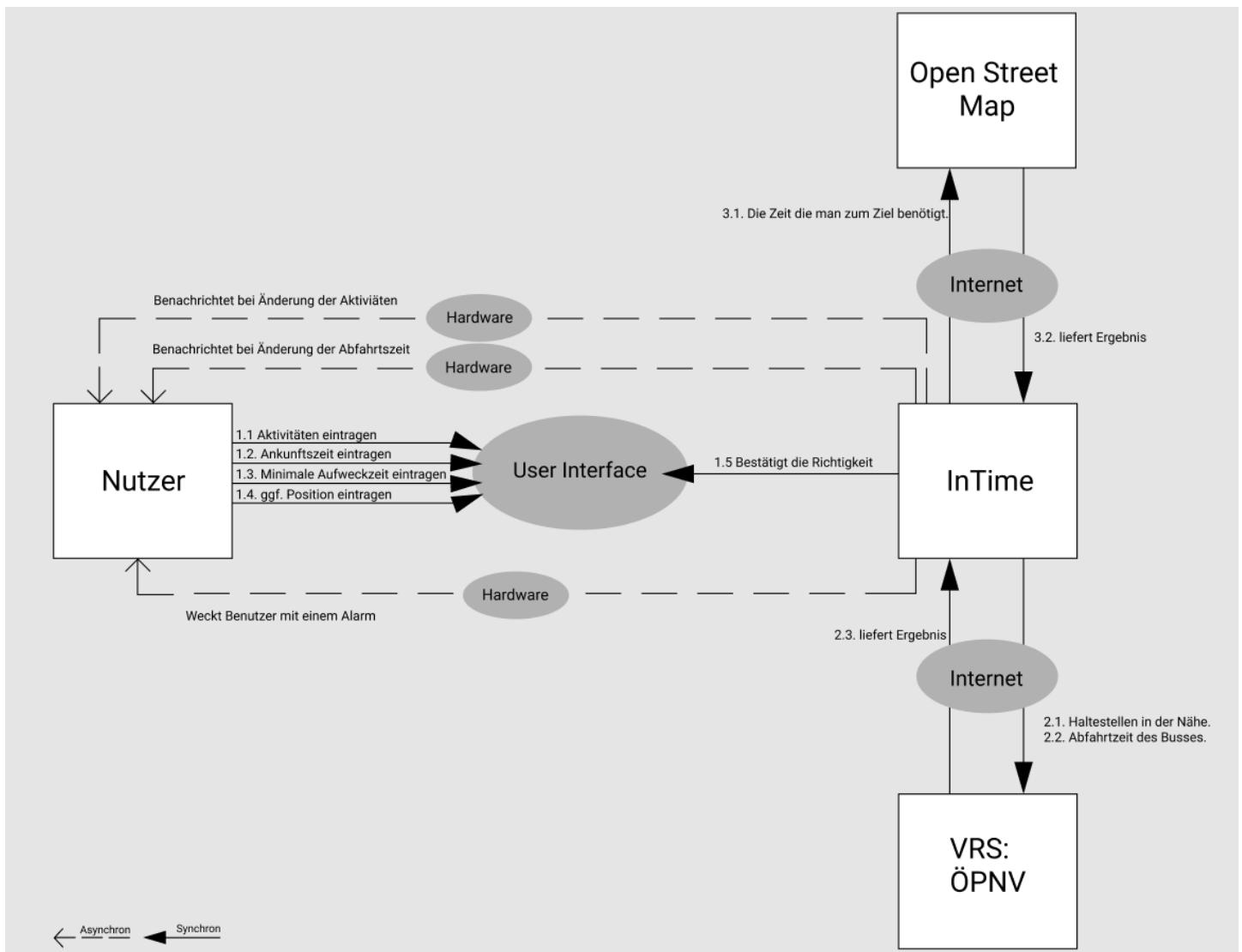


Abbildung 5: Präskriptives Kommunikationsmodell

12. 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.

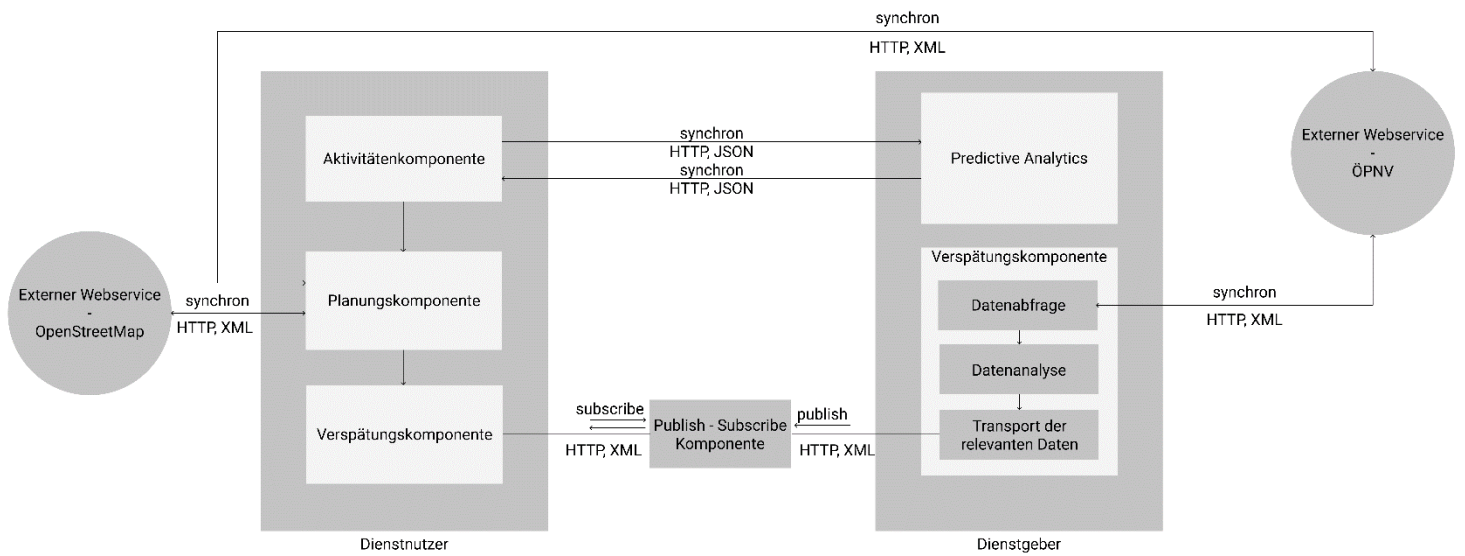


Abbildung 6: Architekturdiagramm

13. Risiken

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 entstünde 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 (obsolete 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

14. 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.

15. Verzeichnisse

15.1 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 (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-Anthrazit-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 (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] „InTime“ Projektplan, unter: https://github.com/sgeier96/EISWS1718DemizkiGeier/blob/master/MS1/DemizkiGeier_ProjektplanEISWS1718.pdf (zuletzt abgerufen am 12.11.17)

15.2 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)