

FACHHOCHSCHULE KÖLN FAKULTÄT FÜR INFORMATIK UND  
INGENIEURWISSENSCHAFTEN

---

## ENTWICKLUNGSPROJEKT INTERAKTIVE SYSTEME

### Meilenstein 2

Campus Gummersbach  
im Studiengang  
Medieninformatik

Betreut von:

Prof. Dr. Kristian Fischer  
Prof. Dr. Gerhard Hartmann  
Robert Gabriel, B. Sc.

ausgearbeitet von:

DERYA ERGUEL  
SINEM KAYA

26. Juni 2015

## Inhaltsverzeichnis

<b>1</b>	<b>Zielhierarchie</b>	<b>3</b>
1.1	Strategische Ziele . . . . .	3
1.2	Taktische Ziele . . . . .	3
1.3	Operative Ziele . . . . .	3
<b>2</b>	<b>related-works</b>	<b>4</b>
2.1	ReiterApp . . . . .	4
2.2	Cavallo-Reitcoach . . . . .	4
2.3	sonstige . . . . .	4
<b>3</b>	<b>Alleinstellungsmerkmale</b>	<b>6</b>
3.1	Routenplaner . . . . .	6
3.2	Wetterbenachrichtigungen . . . . .	6
<b>4</b>	<b>Methodischer Rahmen - Mensch Computer Interaktion</b>	<b>7</b>
4.1	User Centered Design vs. Usage Centered Design . . . . .	7
4.2	MCI-Vorgehensmodelle . . . . .	7
4.3	Methoden der MCI . . . . .	8
4.4	Identifizierung der Stakeholder . . . . .	10
4.5	Benutzer und Intressengruppen . . . . .	11
4.6	Stakeholderanalyse . . . . .	11
4.6.1	Reiter . . . . .	11
4.6.2	Reit-Interessierte . . . . .	11
4.6.3	Sponsoren . . . . .	11
<b>5</b>	<b>Kommunikationsmodell</b>	<b>13</b>
5.1	Deskriptives Kommunikationsdiagramm . . . . .	13
5.2	Präskriptives Kommunikationsdiagramm . . . . .	15
<b>6</b>	<b>Risiken</b>	<b>16</b>
6.1	Implementierung der Schnittstellen . . . . .	16
6.2	Zeitliches Nicht-Erreichen des Ziels . . . . .	16
6.3	geringe Nachfrage . . . . .	16
6.4	Keine genau Lokalisierung des Anwenders . . . . .	16
6.5	Keine oder falsche Wetterinformationen . . . . .	16
<b>7</b>	<b>Spezifikation der Proof-Of-Concepts</b>	<b>17</b>
7.1	Wetterdaten mit der GET-Methode abrufen . . . . .	17
7.2	GPS Lokalisierung ermöglichen . . . . .	17
7.3	Versenden und Empfangen von Nachrichten über Google Cloud Messaging . . . . .	17
<b>8</b>	<b>Architekturdiagramm</b>	<b>19</b>
<b>9</b>	<b>Architekturbegründung</b>	<b>20</b>

<b>10 Projektplan</b>	<b>22</b>
<b>11 Literaturverzeichnis</b>	<b>24</b>

# 1 Zielhierarchie

Im Folgenden werden die Ziele des Projekts dargelegt. Eine konkrete Formulierung der Ziele ist für die weitere Planung zwingend notwendig und trägt zum Erfolg des Projekts bei.

## 1.1 Strategische Ziele

Durch die folgenden Ziele sollte den Benutzern der Anwendung eine Möglichkeit gegeben werden das Reiten außerhalb des eigenen Hofs möglichst angenehm zu planen und zu gestalten.

Als primäres Ziel soll auf langfristiger Sicht die individuelle Routenplanung einfach und übersichtlich ermöglicht werden, sodass die Reiter vorausschauend planen können. Die Benutzer sollen Informationen über die Route erhalten und über das Wetter informiert werden. Weiterhin soll die Möglichkeit geboten werden, ihre Erfahrungen untereinander auszutauschen und somit auch die Möglichkeit haben, sich an den Routen von anderen Benutzern zu orientieren.

## 1.2 Taktische Ziele

Aus den strategischen Zielen werden die taktischen Ziele abgeleitet. Zunächst sollte eine Benutzermodellierung durchgeführt werden, damit alle Erwartungen und Anforderungen der Stakeholder berücksichtigt werden. Unter anderem müssen Schnittstellen zur Ortung und einer Wetter API integriert werden, damit der Benutzer, Informationen über seinen aktuellen Standort bekommt. Das User Interfaces soll möglichst unkompliziert gestaltet werden, um den Benutzern der Anwendung eine komfortable Nutzung auch während eines Rittes zu ermöglichen.

Um die Reiter auf dem aktuellsten Stand halten zu können, sollen sie über wichtige Neuigkeiten, wie ein nahender Unwetter informiert werden.

## 1.3 Operative Ziele

Die operativen Ziele befassen sich mit der Frage, welche Schritte durchgeführt werden müssen, um das Endziel zu erreichen.

Eine Wahl zu den Techniken der Vorgehensmodelle zur Mensch-Computer-Interaktion sollte getroffen werden.

Die technischen Anforderungen an das System müssen gefunden werden. Dafür ist eine Abwägung zwischen unterschiedlichen Möglichkeiten notwendig. Es soll ein Prototyp erstellt werden, der die benötigten Funktionen technisch durchspielen kann und somit die Proof-Of-Concepts erfüllt.

Der Prototyp soll anschließend von realen Testpersonen getestet werden, welches in der Evaluierungsphase geschehen soll.

## 2 related-works

Eine Marktrecherche ist zwingend notwendig, um das eigene Projekt von den Konkurrenzprodukten unterscheiden zu können. Daraus leitet sich oftmals das Alleinstellungsmerkmal ab, welches im weiteren Verlauf von Meilenstein 2 dargestellt wird. Vor- und Nachteile der folgenden Konkurrenzprodukte werden erörtert und abgewogen:

### 2.1 ReiterApp

Durch die ReiterApp (1) soll es Benutzern der Anwendung gelingen die Momente im Alltag festzuhalten und zu dokumentieren. Dies geschieht in der Form von Aufzeichnungen von Routen oder aber auch durch Fotos, die jeder Benutzer mit anderen Benutzern teilen kann. Es handelt sich bei dieser App um eine Community, die versucht Freunde, die als gemeinsames Interesse das Reiten teilen, zusammen zu bringen. Als zusätzliches Feature werden aktuelle News aus der Pferdewelt angezeigt.

Als positives Feedback ist zu erwähnen, dass verschiedene Kartenansichten genutzt werden können. Im Großen und Ganzen ist die Applikation ein gut geeignetes Community-Portal, womit es ermöglicht wird die Momente festzuhalten und mit Freunden zu teilen und neue Routen zu entdecken. Jedoch sind einige wichtige Funktionalitäten, die für den Reiter von Bedeutung sind nicht mit inbegriffen. Beispielsweise werden die Benutzer nicht über wichtige Wetterinformationen informiert. Umgebungsinformationen sind ebenfalls nicht vorhanden.

### 2.2 Cavallo-Reitcoach

Bei der Cavallo-Reitcoach (2) handelt es sich um eine Applikation, die verschiedene Übungseinheiten und Strategien anbietet. Die Übungseinheiten bestehen aus Texten und Bildern. Informationen zum Schwierigkeitsgrad, zur Ausrüstung für das Pferd, zu dem Reiter und zu Trainingsorten sind auch vorhanden.

Im Vergleich zur ReiterApp bietet diese Applikation nur die Möglichkeit, das Pferd zu fördern. Dadurch könnte eine potentielle Verbesserung der Leistungen des Pferdes erreicht werden. Allerdings werden keinerlei Routen zum Planen und zum Entdecken angeboten, welches das primäre Ziel unseres Projekts darstellt.

### 2.3 sonstige

Alternativ sind verschiedene GPS-Tracker optional verfügbar. Jedoch sind diese allgemein und dementsprechend nicht explizit für die Domäne angepasst. Beispielsweise ist die Applikation GeoTracker(11) eine gute Alternative um eine Route aufzuzeichnen. Es werden zusätzliche Informationen zur Zeit, Höhe, Geschwindigkeit etc. angegeben. Wichtige Aspekte für eine Reitroute werden aber auch hier nicht mit eingebunden, da die Anwendung viel zu allgemein ist und nicht domänenspezifisch.

Eine weitere Möglichkeit, wie man an aktuelle Informationen gelangt sind die Informationen, die von den Reiterverbänden angeboten werden. Einige Verbände bieten sogar Zeitschriften an. Des weiteren sind Reiterhöfe auch eine gute Option, um sich über die Domäne zu informieren. Dort könnte sogar für Reit-Interessierte die Möglichkeit geboten werden einen ersten Eindruck des Reitens zu bekommen.

### **3 Alleinstellungsmerkmale**

Das Alleinstellungsmerkmal zeichnet aus, in dem den Benutzern eine Anwendung geschaffen werden soll, die es ermöglicht, sichere und weitestgehend barrierefreie Routen zu planen und aufzuzeichnen. Dabei soll der Benutzer während der ganzen Zeit begleitet werden und Informationen automatisch erhalten, die ihm bei seinen Ritten mit dem Pferd erleichtert und von möglichen Gefahren abwendet. Das Ganze soll möglichst einfach, aber ansprechend im Design realisiert werden, um den Benutzern einen komfortablen Umgang mit dem System zu ermöglichen.

#### **3.1 Routenplaner**

Der Routenplaner soll dem Benutzer die Möglichkeit für das Erstellen von neuen Routen geben. Die Route soll zusätzliche Informationen zur aktuellen Ortschaft und über eventuelle Gefahren beinhalten. Diese Gefahren können sowohl giftige Pflanzen, als auch gefährliche Straßenschäden sein. Der Benutzer soll durch die vorhandenen Informationen die Routen planen können. Der Reiter kann für ihn fremde Landschaften entdecken und bekannte Gefahren vorausschauend umgehen.

#### **3.2 Wetterbenachrichtigungen**

Die Wetterbenachrichtigung soll die Benutzer vor Antritt des Unwetters, zeitgerecht warnen, sodass die Planung einer Rückkehr zum Reiterhof vorgenommen werden kann. Ist der Reiterhof nicht in unmittelbarer Nähe, kann der Reiter weitere Reiterhöfe in der Umgebung aufsuchen, die ihm auf der Route angezeigt werden.

## 4 Methodischer Rahmen - Mensch Computer Interaktion

### 4.1 User Centered Design vs. Usage Centered Design

Der User Centered Design beschäftigt sich mit der benutzerorientierten Gestaltung von Produkten bzw. Systemen. Das Ziel hierbei ist das System gebrauchstauglich und zweckdienlich zu gestalten. Es soll die vielen unterschiedlichen Merkmale, Aufgaben und Bedürfnisse der Benutzer während des Entwicklungsprozesses in einem System berücksichtigen. Der wesentliche Unterschied zum Usage Centered Design besteht darin, dass der Nutzungskontext variieren kann. Die Nutzungsanforderungen sind beim Usage Centered Design der Kernpunkt, worauf die Entwicklung des Systems stattfinden muss.

Das System soll mit Hilfe des User Centered Design eine nutzerzentrierte Gestaltungslösung bieten, um den Anforderungen und den Erfordernissen der Benutzer gerecht zu werden. Die Benutzer sollen während des Entwicklungsprozesses in jeder Aktivität miteinbezogen werden, um die Ziele bestmöglich umsetzen zu können.

### 4.2 MCI-Vorgehensmodelle

Die Norm ISO 9241 Teil 210(12) ist ein internationaler Standard. Sie enthält Richtlinien der Mensch-Computer-Interaktion. Die Anbindung von realen Benutzern soll eine konkrete Benutzermodellierung ermöglichen. Die Allgemeinheit der Norm im Vergleich zu den anderen Vorgehensmodellen soll als Vorteil gesehen werden. Die Arbeitsweise kann dadurch individuell gestaltet werden. Ein wesentlicher Vorteil sind die Iterationen, die in jedem Schritt der Entwicklungsphase durchgeführt werden können, bis die Gestaltungslösung die Nutzungsanforderungen erfüllt.

Neben der ISO 9241 Teil 210 sind auch andere Vorgehensmodelle wie das Scenariobased usability engineering nach Rosson und Carroll oder das Usability engineering lifecycle nach Mayhew vorhanden. Das Usability engineering lifecycle ermöglicht eine parallele Planungs- und Entwicklungszeit, jedoch wird die Gebrauchstauglichkeit erst zu einem späteren Zeitpunkt sichergestellt. Dies ist ein großer Nachteil, welches die Umsetzung des Projekts negativ beeinflussen kann. Es ist von großer Bedeutung, dass der Benutzer in jedem Entwicklungsprozess miteingebunden wird, weshalb dieses Vorgehensmodell ausgeschlossen werden kann.

Das Scenariobased usability engineering nach Rosson und Carroll ist ein Vorgehensmodell, welches den Benutzer aus kognitiver Sicht mit einbringt. Ein wesentlicher Nachteil hierbei ist der zeitliche Aufwand, der notwendigerweise bei der Wahl dieses Vorgehensmodells mitgebracht werden muss. Zeit ist ein enormer Risikofaktor für das Projekt und die Risiken sollten soweit wie möglich



ausgeschlossen werden.

Nach einer Abwägung der Vor- und Nachteile der Vorgehensmodelle, fällt die Wahl auf die Norm ISO 9241 Teil 210. Durch die ISO 9241 Teil 210 muss der Benutzer konkret analysiert werden. Der Benutzer muss in jedem Entwicklungsschritt mit eingebunden werden. Vielmehr sollte erwähnt werden, dass die Anbindung von realen Benutzern das System um ein vielfaches optimiert.

Um geeignete Design-Lösungen zu finden, sollten in der Entwicklungsphase Papierbasierte-Prototypen erstellt werden. Computerbasierte Prototypen bieten zwar eine von selbst ablaufende Animationsmöglichkeit an, jedoch ist die Umsetzung einer Flash-Animation beispielsweise aus zeitlichen Gründen nicht realisierbar. Ausserdem sollte erwähnt werden, dass mehrere Lösungen zum Prototypen vorhanden sein sollten, um die bestmögliche Design-Lösung zu erhalten. Dieses Argument spricht für das Papierbasierte-Prototyp, da diese aus reinen Skizzen besteht und einen nicht zu großen Arbeitsaufwand, im Vergleich zur Animation, mit sich trägt.

### 4.3 Methoden der MCI

Im folgenden Abschnitt sollen geeignete Methoden für die Umsetzung der nutzerzentrierten Gestaltungslösung gewählt werden. Um präzisere Entscheidungen treffen zu können, soll der in Abbildung 1 dargestellte benutzerzentrierte Gestaltungsprozess nach der Norm ISO 9241 Teil 210 nochmals in Betracht gezogen werden.

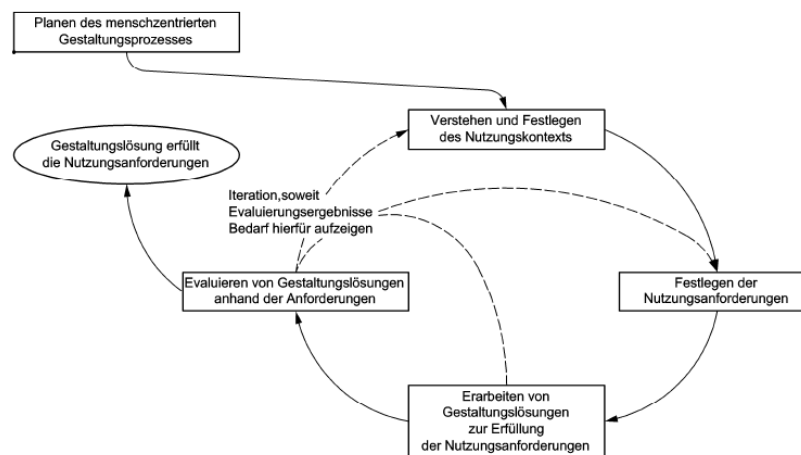


Abbildung .1: Wechselseitige Abhängigkeit menschenzentrierter Gestaltungsaktivitäten nach der Norm ISO 9241 Teil 210)

Der erste Schritt besteht darin, dass der menschenzentrierte Gestaltungsprozess geplant werden soll. Dies beginnt mit dem Verstehen und Festlegen des Nut-

zungskontextes. Die Identifizierung und Erstellung von Nutzungskontexten dient zur Ermittlung von Anforderungen der Benutzer. Um die Anforderungen der Benutzer spezifisch erfassen zu können, muss eine detaillierte Benutzermodellierung erstellt werden. Zunächst müssen die Stakeholder identifiziert und anschließend User Profiles erstellt werden. Des weiteren sollen Personae erstellt werden, um die Merkmale der Testpersonen eindeutig identifizieren zu können. Anschliessend sollen Use Cases erstellt werden, um die Intention des Benutzers zu erfassen. Nach der Benutzermodellierung kann die Identifizierung des Nutzungskontextes erfasst werden.

Um geeignete Gestaltungs-Lösungen zu finden, sollten in der Entwicklungsphase Papierbasierte-Prototypen erstellt werden. Computerbasierte Prototypen bieten zwar eine von selbst ablaufende Animationsmöglichkeit an, jedoch ist die Umsetzung einer Flash-Animation beispielsweise aus zeitlichen Gründen nicht realisierbar. Ausserdem sollte erwähnt werden, dass mehrere Lösungen zum Prototypen vorhanden sein sollten, um die bestmögliche Design-Lösung zu erhalten. Dieses Argument spricht für das Papierbasierte-Prototyp, da diese aus reinen Skizzen besteht und einen nicht zu großen Arbeitsaufwand, im Vergleich zur Animation, mit sich trägt.

Zu guter Letzt soll die Evaluation durchgeführt werden. Ein analytisches Verfahren zur Evaluation mit Usability-Experten würde mit hohen Kosten zusammenhängen, weshalb ein empirisches Verfahren angewendet werden soll. Des weiteren werden quantitative und qualitative Verfahren unterschieden. Das quantitative Verfahren beruht auf der Evaluation von Ergebnissen, die Zahlen basiert vermessen werden sollen. Im Gegensatz dazu geht es beim qualitativen Verfahren um die sprachliche Einschätzung von Testpersonen. Für das Projekt kommen nur die qualitativen Verfahren in Frage. Die nötigen Verbesserungen sollen aufbauend auf den Äußerungen und den Anregungen der Benutzer geplant und umgesetzt werden.

Es bieten sich folgende Evaluationsmethoden an: Heuristische Evaluation, Cognitive Walkthrough, AttrakDiff und Think Aloud.

Bei AttrakDiff handelt es sich um einen quantitativen Fragebogen, weshalb diese als Möglichkeit weg fällt oder eventuell als zweite Evaluationsmethode verwendet werden kann.

Die Heuristiken bei der heuristischen Evaluation sind stark verallgemeinert, so dass spezielle Anforderungen an die Usability beispielsweise nicht abgedeckt werden können.

Beim Cognitive Walkthrough findet die Evaluation mit Spezialisten der Mensch-Computer-Interaktion statt. Jedoch sollte die Evaluation von Benutzern durchgeführt werden, die Interesse an dem System teilen. Da das System an sich wenig Funktionalitäten und Komplexität aufweist, ist der Cognitive Walkthrough von Polson, Lewis nicht als Evaluationsmethodik geeignet.

Deshalb sollte als Evaluationsmethode Think Aloud bevorzugt werden. Dadurch käme auch die kognitive Sicht der Benutzer zum Vorschein. Die Methode wird von einem Moderator und dem Probanden durchgeführt. Es werden zwei Phasen

unterschieden: die Aufwärmphase und die Evaluationsphase. Dabei soll der Proband in der Evaluationsphase gewisse Funktionalitäten, die er vom Moderator als Anweisung erhält, simulieren können und dabei gleichzeitig laut denken. Es besteht zwar die Gefahr, dass die Probanden nicht genügend laut denken. Um dieses Problem zu vermeiden sollen sie in der Aufwärmphase reichlich ermutigt werden.

#### 4.4 Identifizierung der Stakeholder

Um eine möglichst präzise Benutzermodellierung zu erlangen, soll die Identifizierung von potentiellen Stakeholdern erfasst werden, welches für die menschenzentrierte Gestaltung laut ISO 9241 Teil 210 (3) notwendig ist.

Stakeholder werden als

*Einzelperson oder Organisation, die ein Anrecht, einen Anteil, einen Anspruch oder ein Interesse auf ein beziehungsweise an einem System oder an dessen Merkmalen hat, die ihren Erfordernissen und Erwartungen entsprechen* (4)

definiert. Dementsprechend werden zunächst grob die potentiellen Stakeholder aufgelistet, die in Meilenstein 3 nochmal , die für den Meilenstein 2 notwendigerweise erfasst werden müssen, um das Kommunikationsmodell qualitativ gestalten und erfassen zu können. Die Benutzermodellierung wird anschließend in Meilenstein 3 fortgesetzt.

## 4.5 Benutzer und Interessengruppen

Bezeichnung	Beschreibung
Reiter	Alle Reiter, die seit mehreren Jahren regelmäßig reiten
Reit-interessierte	Personen die, das Reiten erlernen möchten oder momentan neu erlernen
Sponsoren	Unternehmer, die für ihr Unternehmen werben
Reittrainer	Trainer, die Reitstunden geben
Reitvereine	Kleine bis große Vereine(regionale oder landesweit)
Veranstalter	Veranstalter, die Reitturniere veranstalten
Höfe	Reiterhöfe und Bauernhöfe, die den Reitern eine Möglichkeit zum Rasten anbieten

Tabelle .1: Benutzer- und Interessengruppen

## 4.6 Stakeholderanalyse

Anhand in der Tabelle 1 aufgelisteten Benutzer und Interessengruppen, können im weiteren Verlauf die Stakeholder definiert werden. Um eine ordentliche Repräsentationen der Stakeholder zu erhalten ist eine ausführliche Auflistung der Benutzermerkmale essentiell notwendig. Zunächst werden die Benutzer in Gruppen eingeteilt, die wie folgt aussehen:

### 4.6.1 Reiter

Der Stakeholder Reiter schließt alle Reiter, die seit mehreren Jahren reiten und Trainer ein. Reiter sollen die primären Benutzer des Systems darstellen. Sie sollen mit dem System direkt interagieren. Die Anwendung soll hauptsächlich zu ihrem Nutzen sein. Sie sollen alle Funktionalitäten des Systems nutzen können, weshalb die Gestaltung besonders zu ihrem Interesse geplant und vorgenommen werden sollte.

### 4.6.2 Reit-Interessierte

Hierbei handelt es sich um Reit-Interessierte, die Interesse an der selben Aktivität teilen oder sich vor kurzem zum Reiten entschlossen haben. Reit-Interessierte stellen unsere sekundären Benutzer dar. Sie können die Anwendung zu sonstigen Zwecken nutzen, wie z.B. interessenbedingt als Informationsquelle und als Kommunikationsmittel.

### 4.6.3 Sponsoren

Der Stakeholder Sponsoren schließt Reitvereine, Veranstalter, Sponsoren und Höfe ein. Durch die Anwendung können sie innerhalb der Domäne Aufmerksamkeit erreichen. Die Anwendung soll nur auf den Reiter spezialisiert sein. Das

bedeutet, dass Sponsoren nichts mit den Funktionalitäten der Anwendung anfangen können. Somit gelten sie als tertiäre Benutzer, die kein Interesse an dem System teilen, sondern als Zweckmäßigkeit nutzen können.

## 5 Kommunikationsmodell

### 5.1 Deskriptives Kommunikationsdiagramm

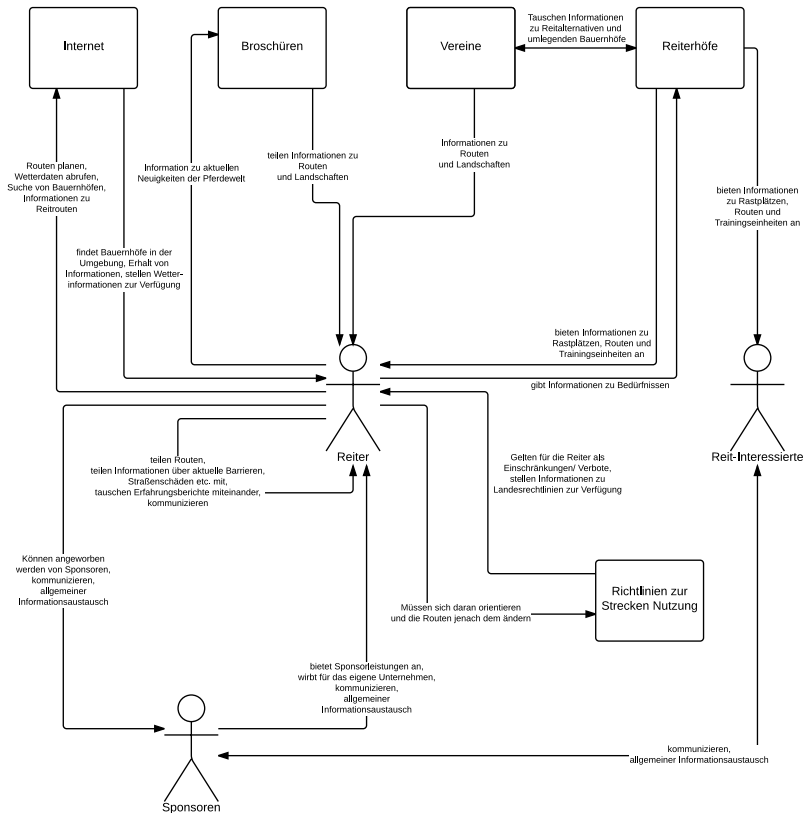


Abbildung .2: Deskriptives Kommunikationsdiagramm

In dem deskriptiven Kommunikationsdiagramm sind u.A. die Benutzer und die Informationsquellen abgebildet. Das Diagramm stellt die zuvor aufgelisteten Stakeholder dar. Die Informationsquellen sind die Vereine, die Broschüren, die Reiterhöfe und das Internet. Sicherheitsvorkehrungen und die rechtlichen Richtlinien werden hier als "Richtlinien zur Strecken-Nutzung" dargestellt. Die größte Bandbreite an Informationen wird über das Internet bereitgestellt. Der Abruf der aktuellen Wetterdaten wird ermöglicht. Ausserdem werden Informationen zu Landschaften und Routenplanungen bereitgestellt. Innerhalb eines Vereins kann der Reiter Informationen über Strecken erhalten und über Erfahrungen

von anderen profitieren oder sogar die eigenen Routen teilen. Möchte der Reiter sich vorab über Landschaften erkundigen und dabei auch Informationen über die Umwelt erhalten können spezielle Broschüren und Handbücher sich hierfür eigenen. Über die Kommunikation von Vereine und Reiterhöfe, können sich beide Instanzen auf dem Laufenden halten. Darüber hinaus werden somit die Bedürfnisse der Reiter auf die eigenen Angebote angepasst und Informationen aktuell halten. Die Reiterhöfe bieten den Reitern Rastplätze zum Erholen und Möglichkeiten zum Trainieren an. Des weiteren haben die Reiter über die Reiterhöfe die Möglichkeit direkt über anliegende Streckenänderungen, Gefahren-Stellen, Informationen zu erhalten. Die Reiter kommunizieren untereinander und teilen ihre persönlichen Erfahrungen über Routen. Die Informationen, die die Reiter untereinander austauschen liegen unter dem gleichen Interesse. Sie werden somit über mögliche Probleme aufgeklärt und können somit präzisere Informationen z.B. zur Streckenbeschaffenheit weitergeben. Die "Richtlinien zur Strecken-Nutzung" dienen dazu, dass die Reiter sich vorab über verbotene Straßen informieren, bekannte Gefahrenquellen auf der eigenen Route erkennen und anhand dieser Informationen ihre Routen orientieren. Die Reiter können sich von Sponsoren anwerben lassen und somit ihre Bedürfnisse nach außen tragen und sich z.B. ein besseres Equipment leisten. Die Sponsoren können Reiter als Werbeplattform für ihr Unternehmen anwerben und Sponsorleistungen anbieten.

**Fazit** Aus dem deskriptiven Kommunikationsdiagramm ist stark zu erkennen, dass die Nutzer mehrere Informationsquellen nutzen müssen um die benötigten individuellen und präzisen Informationen zu erhalten. Mehrere unterschiedliche Informationsquellen sind nötig um eine Route unter mehreren Aspekten zu planen.

## 5.2 Präskriptives Kommunikationsdiagramm

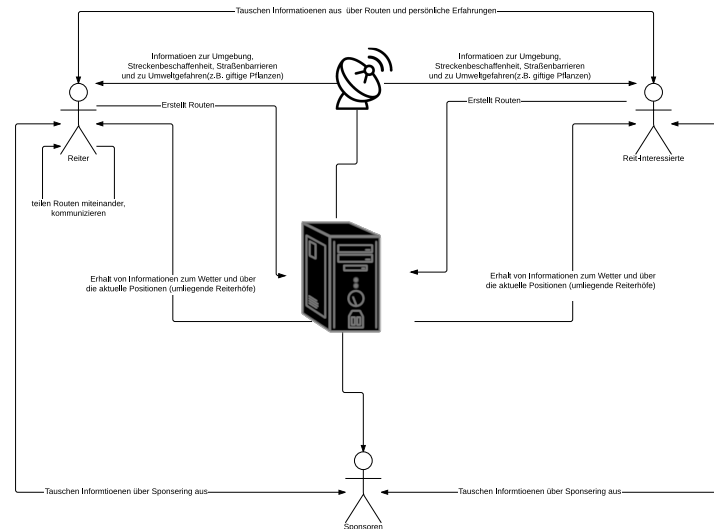


Abbildung .3: Präskriptives Kommunikationsdiagramm

Das Abgebildete präskriptive Kommunikationsdiagramm zeigt die einzelnen Kommunikationspartner und Informationsquellen. Die Stakeholder ändern sich nicht im Vergleich zu dem deskriptiven Kommunikationsdiagramm. Aus dem deskriptiven Kommunikationsdiagramm resultierte, dass die Nutzer mehrere Wege gehen müssen, um die geeigneten Informationen für ihre Route ausfindig zu machen. Die Vielfalt von Informationsquellen werden minimiert. Dem Benutzer wird eine individuelle Routenplanung gewährleistet, die unter Berücksichtigung von mehreren Aspekten zur Verfügung steht. Das präskriptive Kommunikationsdiagramm muss eine Veranschaulichung des gezielten Soll-Zustandes und der Verbesserung des Ist-Zustandes erreicht werden.

Die Reiter, die eine Route erstellen, erhalten außerhalb der Routenplanung, Informationen zur gewünschten Strecke, zur umliegenden Landschaft, zum aktuellen Wetter und zu Gefahrenquellen für Mensch und Tier. Außerdem erhält der Reiter Informationen über erlaubte und verbotene Strecken. Der Sponsor kann Informationen von Reitern erhalten. Die Reiter tauschen sich untereinander über Routen aus und können darüber berichten wie die eigene Erfahrung mit der Route war.



## **6 Risiken**

### **6.1 Implementierung der Schnittstellen**

Das Projekt weist viele unterschiedliche technische Funktionalitäten auf, die umgesetzt werden müssen. Es besteht die Gefahr, dass die Umsetzung scheitert, da mehrere Schnittstellen implementiert werden müssen um die Funktionalitäten zu gewährleisten. Scheitert eine Abfrage kann der Dienst nicht erfolgreich ablaufen. Um dieses Risiko zu minimieren, sollten zunächst nur die als wichtig erachteten technischen Funktionalitäten des Systems realisiert werden. Ausserdem kann beispielsweise die OpenStreet API durch die Google Maps API ersetzt werden.

### **6.2 Zeitliches Nicht-Erreichen des Ziels**

Aufgrund der Tatsache, dass das Projekt im Gegensatz zur verfügbaren Zeit, sehr umfangreich ist, könnte das Risiko eintreten, dass das Ziel nicht erreicht werden kann. Um dieses Risiko zu vermeiden, sollten die wichtigsten Kernpunkte zur Realisierung der Anwendung möglichst genau besprochen werden. Ist das Risiko eingetreten, sollte im schlimmsten Fall die Implementierung des Google Cloud Messagings weg gelassen werden, da es für den Erfolg der Anwendung nicht von Bedeutung ist. Es stellt lediglich eine Zusatzfunktion dar. Dadurch sollte genügend Zeit gespart werden.

### **6.3 geringe Nachfrage**

Ist die Anwendung auf dem Markt, besteht die Gefahr, dass eine zu geringe Nachfrage besteht. Damit dieses Risiko nicht eintritt, muss reichlich genug geworben werden. Das Geschäftsmodell soll dafür sorgen, dass Angebot und Nachfrage durchgehend steigen.

### **6.4 Keine genau Lokalisierung des Anwenders**

Es kann passieren das das GPS Ortungsdienst keine genaue Ortung vornimmt und dem Benutzer falsche Informationen leitet. Um dieses Problem frühzeitig zu erkennen, muss die Anwendung an mehreren Orten wie zum Beispiel im mehreren Wald-Bereichen getestet werden, die in den Proof-Of-Concepts vorgenommen werden.

### **6.5 Keine oder falsche Wetterinformationen**

Es besteht die Gefahr, dass der Benutzer keine oder falsche Daten zum Wetter erhält. Dies kann bei einem Ritt außerorts bei plötzlich hervor ziehendem Unwetter dem Reiter Probleme bereiten. Um dieses Problem zu umgehen, sollen in verschiedenen Städten die Informationen zum Wetter abgerufen werden, um einen bestmöglichen Vergleich zu erhalten. Diese Funktion wird ebenfalls in den Proof-Of-Concepts getestet.

## 7 Spezifikation der Proof-Of-Concepts

Die aus den Risiken erarbeiteten Problematiken dienen uns als Hilfe für die Erstellung der Proof-Of-Concepts. Das Risiko Fehlschlagen der Implementierung kann eine drastische Folge für die Umsetzung des Projekts sein. Daraus ableitend sollen folgende Funktionalitäten zwingend getestet und realisiert werden. Die Verwirklichung der Funktionalitäten 7.1, 7.2 und 7.3 sind von äusserster Bedeutung für den Erfolg der Anwendung. Deshalb sollten diese Funktionen ordentlich durchdacht werden, um die Umsetzung zu gewährleisten.

### 7.1 Wetterdaten mit der GET-Methode abrufen

Es soll getestet werden ob der Erhalt der Wetterdaten mit der GET-Methode realisiert werden kann. Hierzu soll eine einfache Anwendung programmiert werden, welches die Wetterdaten im XML-Format ausgibt.

Um den Erfolg der Anwendung beweisen zu können, soll die Anwendung durch Eingabe von 20 verschiedenen Städten getestet werden. Das Testen gilt als erfolgreich, wenn der Erhalt von aktuellen Wetterdaten zu der erwünschten Stadt, ermöglicht werden kann. Das Fail-Kriterium wäre dementsprechend das Fehlschlagen des Abrufs mit der GET-Methode.

Das Fallback für dieses Proof-Of-Concept wäre die Nutzung der AccuWeather API, die auch eine REST-Schnittstelle zur Verfügung stellt.

### 7.2 GPS Lokalisierung ermöglichen

Die GPS Lokalisierung ist zwingend notwendig, um die Routenplanungen in der Implementierungsphase zu ermöglichen. Die Erstellung der Routen soll anhand der Google Maps API realisiert werden.

Der Erfolg der Funktion hängt von der Präzision der Lokalisierung ab, die an 10 verschiedenen Standorten getestet werden soll. Hier soll die Zeit ebenfalls gemessen werden, da die Ortung nicht all zu lang dauern sollte. Die zeitliche Spanne von 30 Sekunden solle auch nicht gesprengt werden.

Das Exit-Kriterium ist bei dieser Funktion die präzise Ortung an verschiedenen Orten. Als Fail-Kriterium würde es genügen, wenn eine einzige Lokalisierung, die in den 30 Testversuchen vorgenommen werden sollen, fehlschlägt.

Besteht die Gefahr, dass die Google Maps API nicht verwendet werden kann, sollte die Alternative der Gebrauch der OpenStreetMap API sein, die eine andere Vorgehensweise und einen anderen Komplexitätsgrad aufweist.

### 7.3 Versenden und Empfangen von Nachrichten über Google Cloud Messaging

Der Nachrichtenaustausch soll über die Schnittstelle Google Cloud Messaging realisiert werden. Es soll getestet werden, ob das Versenden einer Nachricht von Benutzer A zu Benutzer B und anders herum das Empfangen umgesetzt werden kann.

Zum Testen sollen zwei unterschiedliche Smartphones mit dem Betriebssystem Android genutzt werden, die alle miteinander Nachrichten austauschen sollen. Die Benachrichtigung sollte binnen 10 Sekunden ankommen.

Sowohl das Versenden als auch das Empfangen der Nachrichten sollte ermöglicht werden. Schlägt die Umsetzung bei einem einzigen Smartphone fehl, gilt sie als gescheitert. Da diese Funktion aber als zusätzliche Funktion zur Optimierung der Anwendung gedacht ist und die Umsetzung nicht für den Erfolg der Anwendung notwendig ist, soll bei Scheitern dieses Proof-Of-Concepts auf die Funktion verzichtet werden. Dadurch kann Zeit für die programmierlastige Präzisierung der zuvor genannten wichtigen Funktionen gewonnen werden.

## 8 Architekturdiagramm

Das Architekturdiagramm soll die Verteiltheit und die einzelnen Kommunikationsabläufe des Systems darstellen. Die Abbildung wurde anhand der erarbeiteten Funktionalitäten skizziert.

Die Anwendungsnutzer unterscheiden sich nicht, da jeder Benutzer Zugriff auf alle Funktionalitäten haben, auch wenn beispielsweise Sponsoren nicht alle Funktionalitäten des Systems benötigen.

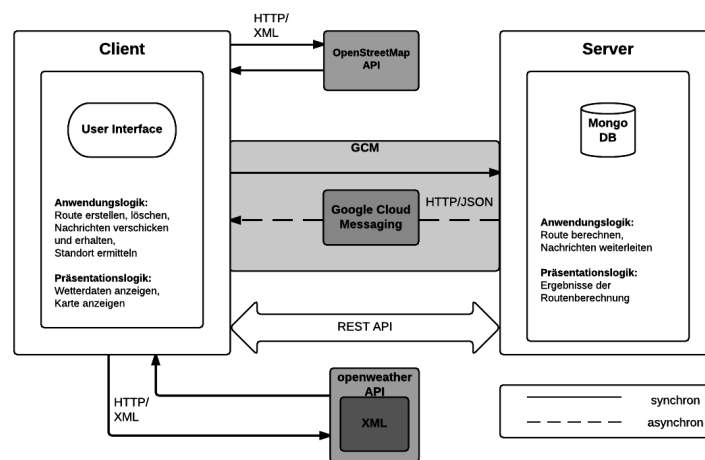


Abbildung .4: Systemarchitektur

### Anwendungslogik und Präsentationslogik

Bei 'Route erstellen' werden die von dem Benutzer eingegebenen Werte zum Server versandt. Daraufhin wird die Route beim Server berechnet. Die errechneten Informationen werden zurück zum Client geschickt. Daraufhin erscheint die Karte auf dem Device des Clients, welches die Präsentationslogik kennzeichnet. Das Versenden von Nachrichten wird über GCM realisiert. Sobald der Server Nachrichten an den Client versendet gelangen diese zu zunächst zu GCM. Die Nachrichten die bei GCM ankommen sind, werden an den Client zugestellt. Auch wenn der Client gerade im Stand-By-Modus ist. Auf dem Client erscheint die Anzeige, dass eine Nachricht empfangen worden ist.

## 9 Architekturbegründung

Aufgrund der Tatsache, dass die Mobilität der Reiter unterstützt werden soll, bietet sich uns die einzige Option einer mobilen Anwendung. Vielfältige Betriebssysteme bieten sich auf dem Markt zur Programmierung von mobilen Anwendungen an, unter Anderem Google Android, IOS uvm.

Blackberry hat ihr Betriebssystem perfekt auf Business-Anwender abgestimmt. Zudem ist die Nutzung der E-Mail Dienste leistungsstark. Eingeschränkte Multimedia-Eigenschaften benachteiligen das Betriebssystem. Die Vorteile von Blackberry OS sind nicht für unser Projekt von Vorteil.

Das Betriebssystem IOS von Apple bietet auch eine sehr hohe Performance. Jedoch ist die Nutzung verschiedener Schnittstellen von Google nicht nutzbar. Eine Einarbeitung in die Programmiersprache Swift kann enorm viel Zeit in Anspruch nehmen.

Da bei Android(7) mit der Programmiersprache Java programmiert werden kann und die Google Produkte perfekt eingebunden werden können fällt die Entscheidung auf Android. Im Zusammenhang mit Android ist eine Programmierung mittels Java und XML die beste Möglichkeit, das Projekt umzusetzen, da das Projekt Java voraussetzt und XML in dem Modul Web-basierte Anwendungen praktisch schon umgesetzt wurde.

Die zentrale Datenerhaltung soll auf dem Server stattfinden. Dadurch können Datenverluste vermieden werden. Durch das Architekturmodell REST - Representational State Transfer - für verteilte Systeme, soll die Anwendung ihre Daten publizieren und abgreifen können. Die Clients sollen auf die vom Server bereitgestellten Ressourcen durch das Transportprotokoll HTTP zugreifen können. Im Gegensatz zur Webservice SOAP - Simple Object Access Protocol - bietet REST nicht so komplexe XML-Nachrichten, welches ein wesentlicher Nachteil von SOAP ist. (8)

Als Kommunikationsmittel und gleichzeitig als Push-Dienst soll Google Cloud Messaging (5) verwendet werden. Die Gründe hierfür beruhen sich auf der Tatsache, Google Cloud Messaging ein kostenloser Dienst ist und es genügend Informationen zur Implementierung vorhanden sind, die bei der Umsetzung von Bedeutung sein können, da es sich um einen erstmaligen Kontakt mit Android handelt. Durch Google Cloud Messaging wird die synchrone und asynchrone Kommunikation zwischen Serverinstanzen und der mobilen Endgeräte ermöglicht.

Zum Realisieren eines Webservers soll Node.js genutzt werden. Die Begründung hierfür ist zum einen, dass Node.js in WBA2 erlernt wurde. Ganz wichtig ist die Kompatibilität von Node.js mit den Schnittstellen, die im Projekt genutzt werden. Die Kommunikation der Benutzer im System untereinander soll mittels XMPP eine ermöglichen. XMPP-CLients sind für alle gängigen Architekturen und Systeme frei verfügbar. Als Framework bietet sich am Besten die asmack an, da diese für die Programmierung mittels Android angepasst wurde.(12)

Die Bereitstellung der Wetterdaten soll durch die Schnittstelle openweather API (6) realisiert werden. Die Openweather API ermöglicht einen kostenlosen Zugriff auf die Wetterdaten, solange es sich nicht um ein kommerzielles Produkt handelt. Es sollte lediglich die Herkunft der Wetterdaten im Endprodukt sichtbar sein. Ein Nachteil ist, dass die kostenlose Schnittstelle auf 10000 Zugriffe im Monat eingeschränkt ist. Ein weiterer Grund für die Wahl der Wetter API ist, dass für die Anfragen an die Schnittstelle ein einfaches REST-Format verwendet wird. Die Wetterinformationen sollen durch die HTTP-Operation GET abgerufen und im XML-Format herausgegeben werden.

Zur Routenplanung stehen zwei Alternativen zur Verfügung, zum Einen Google Maps und zum Anderen die OpenStreetMap. Die OpenStreetMap ist flexibler im Gegensatz zu Google Maps und bietet eine große Bandbreite an Informationen in den Karten. Außerdem sind die Karten von der OpenStreetMap eng gehalten, sodass die Karten einige Adressen, Parks oder beispielsweise sogar wichtige Gebäude wie Krankenhäuser nicht enthalten. Daraus ist ab zu leiten, dass eventuell wichtige Routen zum Reiten fehlen können. Diese relevanten Fakten sollten für das Planen der Routen, durch die Verwendung von Google Maps API sprechen. Die API ist weit verbreitet und stellt Karten für mobile Anwendungen zur Verfügung. Zu dem ist die API völlig kostenlos und ermöglicht als Routenplaner 2500 Anfragen pro Tag. Zudem sollte erwähnt werden, dass bereits vorhanden Routen zum Reiten in Google Maps vorhanden sind.(9)

## 10 Projektplan

Datum / KW	Aktivität	1. Unteraktivität	2. Unteraktivität	Workload geplant /	Workload gesamt	Workload Derya Ergue	tatsächlich Sinem Kaya
14	Exposé	Ideenfindung	Brainstorming	3h		3h	3h
	Dokumentaufbau	Layout / Struktur	Latex	3h		0h	4h
					6h	3h	7h
13.04.2015	<b>Meilenstein 1</b>	Nutzungsproblem		1h		1h	1h
		Zielsetzung		1h		1h	1h
		Verteilte Anwendungslogik		1h		1h	1h
		Wirtschaftliche und gesellschaftliche Relevanz		1h		1h	1h
					4h	4h	4h
27.04.2015	<b>Meilenstein 2</b>	Zielhierarchie	Strategische Ziele	1h		2h	2h
			Taktische Ziele	1h		2h	2h
			Operative Ziele	1h		1h	1h
		related-works	ReiterApp	1h		0h	1h
			Cavallo-Reitcoach	1h		1h	0h
			sonstige	1h		0h	1h
		Alleinstellungsmerkmale		1h		0,5h	0,5h
		Methodischer Rahmen (MCI)		10h		6h	6h
		Kommunikationsmodell		4h		4h	0h
		Risiken		3h		2h	2h
		Spezifikation der POCs		1h		0h	1,5h
		Architekturdiagramm/ Architekturbeurteilung	Architekturdiagramm	1h		2h	0h
			Architekturbeurteilung	2h		0h	2h
		Projektplan		2h		0h	1h
						20,5h	20h
			Insgesamt	Soll: 40h		Ist: 27,5h	Ist: 31h

Abbildung .5: Projektplan zu Meilenstein 1 und 2

## Abbildungsverzeichnis

.1	Wechselseitige Abhängigkeit menschzentrierter Gestaltungsaktivitäten nach der Norm ISO 9241 Teil 210) . . . . .	8
.2	Deskriptives Kommunikationsdiagramm . . . . .	13
.3	Präskriptives Kommunikationsdiagramm . . . . .	15
.4	Systemarchitektur . . . . .	19
.5	Projektplan zu Meilenstein 1 und 2 . . . . .	22

## Tabellenverzeichnis

.1	Benutzer- und Interessengruppen . . . . .	11
.2	Abkürzungsverzeichnis . . . . .	23

## Abkürzungsverzeichnis

Abkürzung	Bedeutung
etc.	et cetera (und die Übrigen)
bzw.	beziehungsweise
z.B.	zum Beispiel
u.A.	unter Anderem
XMPP	Extensible Messaging and Presence Protocol

Tabelle .2: Abkürzungsverzeichnis



## 11 Literaturverzeichnis

- [1] <http://www.reiterapp.de> - Sichtungsdatum: 21.04.2015
- [2] <http://www.cavallo.de/news/jetzt-fuer-android-besser-reiten-mit-der-cavallo-reitcoach-app-powered-by-leovet.1115306.233219.htm#1> - Sichtungsdatum: 21.04.2015
- [3] Ergonomie der Mensch-System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010); Deutsche Fassung EN ISO 9241-210:2010 - Sichtungsdatum: 22.04.2015
- [4] German UPA e.V., Arbeitskreis Qualitätsstandards, German UPA Qualitätsstandard für Usability Engineering, April 2012, Seite 77 [http://www.germanupa.de/data/mediapool/n070\\_qualitaetsstandard\\_der\\_german\\_upa.pdf](http://www.germanupa.de/data/mediapool/n070_qualitaetsstandard_der_german_upa.pdf) - Sichtungsdatum: 22.04.2015
- [5] <https://developer.android.com/google/gcm/index.html> - Sichtungsdatum: 24.04.2015
- [6] [http://www.wetter.com/apps\\_und\\_mehr/website/api/](http://www.wetter.com/apps_und_mehr/website/api/) - Sichtungsdatum: 24.04.2015
- [7] <http://de.statista.com/statistik/daten/studie/182363/umfrage/prognostizierte-marktanteile-bei-smartphone-betriebssystemen/> - Sichtungsdatum: 24.04.2015
- [8] <https://www.mittwald.de/blog/webentwicklung-webdesign/webentwicklung/restful-webservices-1-was-ist-das-uberhaupt> - Sichtungsdatum: 24.04.2015
- [9] <https://developers.google.com/maps/?hl=de> - Sichtungsdatum: 24.04.2015
- [10] <https://play.google.com/store/apps/details?id=com.ilyabogdanovich.geotracker&hl=de> - Sichtungsdatum: 21.04.2015
- [11] <https://play.google.com/store/apps/details?id=com.ilyabogdanovich.geotracker&hl=de> - Sichtungsdatum: 21.04.2015
- [12] <https://wiki.archlinux.de/title/XMPP> - Sichtungsdatum: 25.04.2015