

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
Betreuer1
Betreuer2

ausgearbeitet von:

DERYA ERGUEL
SINEM KAYA

28. April 2015

Inhaltsverzeichnis

1	Zielhierarchie	2
1.1	Strategische Ziele	2
1.2	Taktische Ziele	2
1.3	Operative Ziele	2
2	related-works	3
2.1	ReiterApp	3
2.2	Cavallo-Reitcoach	3
2.3	sonstige	3
3	Alleinstellungsmerkmale	5
3.1	Routenplaner	5
3.2	Wetterbenachrichtigungen	5
4	Methodischer Rahmen - Mensch Computer Interaktion	6
4.1	User Centered Design vs. Usage Centered Design	6
4.2	MCI-Vorgehensmodelle	6
4.3	Methoden der MCI	7
5	Kommunikationsmodell	9
5.1	Kommunikationsdiagramm	9
5.2	"Route erstellen"	10
5.3	"Neighbourhood"	11
6	Risiken	12
6.1	Implementierung der Schnittstellen	12
6.2	Zeitliches Nicht-Erreichen des Ziels	12
6.3	geringe Nachfrage	12
6.4	Keine genau Lokalisierung des Anwenders	12
6.5	Keine oder falsche Wetterinformationen	12
7	Spezifikation der Proof-Of-Concepts	13
7.1	Wetterdaten per Push-Notifications erhalten	13
7.2	GPS Lokalisierung ermöglichen	13
7.3	Versenden und Empfangen von Nachrichten über Google Cloud Messaging	13
8	Architekturdiagramm	14
8.1	Architekturbegründung	14
9	Projektplan	16
10	Literaturverzeichnis	18

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 Hofes 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 sein aktuellen Standort bekommt. Die Gestaltung des 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 auswärts 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 Text und Bild. Informationen zum Schwierigkeitsgrad, Ausrüstung für das Pferd und dem Reiter oder Trainingsorte sind auch vorhanden.

Im Vergleich zur ReiterApp bietet diese Applikation nur die Möglichkeit das Pferd zu fördern. Dadurch könnte eine potenzielle Verbesserung des Pferdes erreicht werden. Allerdings werden keinerlei Routen zum Planen und Entdecken angeboten, welches unser primäres Ziel des 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(10) 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 seitens der Reiterverbände 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 sich dadurch aus, dass dem Benutzer eine Anwendung geschaffen werden soll, die es ermöglicht, sichere und weitestgehend barrierefreie Routen zu planen und aufzuzeichnen. Dabei soll der Nutzer während der ganzen Zeit begleitet werden und automatische Informationen erhalten, die ihm bei seiner Unternehmung mit seinem Pferd erleichtert und von möglichen Gefahren abwendet. Das ganze soll möglichst einfach jedoch ansprechend im Design realisiert werden, um die Benutzer eine komfortable Nutzung des Systems 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 zu den Routen vorausschauend 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 in unmittelbarer Nähe, kann der Reiter weitere Reiterhöfe in der Umgebung aufsuchen, indem er die Route öffnet.

4 Methodischer Rahmen - Mensch Computer Interaktion

4.1 User Centered Design vs. Usage Centered Design

Um eine möglichst hohe Gebrauchstauglichkeit des Systems zu erlangen, soll in den nächsten Kapiteln eine Abwägung der Vorgehensmodelle stattfinden, um eine zielorientierte Auswahl treffen zu können.

Der User Centered Design beschäftigt sich mit der benutzerorientierten Gestaltung von Produkten beziehungsweise Systemen. Dabei soll eine möglichst hohe User-Experience und Usability erzielt werden. 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 durch das mit Hilfe des User Centered Design eine nutzerzentrierte Gestaltungslösung bieten können, um den Anforderungen des Benutzer gerecht zu sein. Die Benutzer sollen während des Entwicklungsprozesses in jeder Aktivität miteinbezogen werden, um die Ziele bestmöglich umsetzen zu können.

4.2 MCI-Vorgehensmodelle

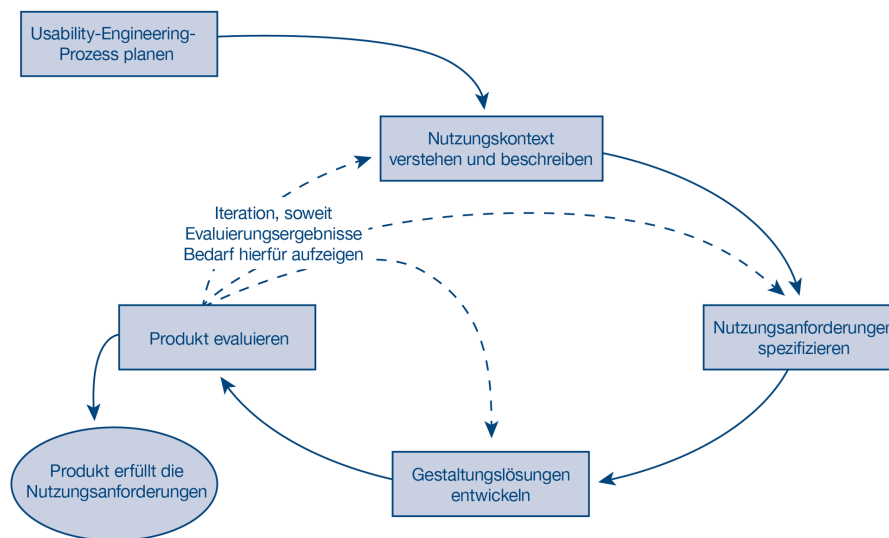
Die Norm ISO 9241 Teil 210 ist ein internationaler Standard. Sie enthält Richtlinien der Mensch-Computer-Interaktion. Die Anbindung von realen Benutzern soll eine konkretere Benutzermodellierung ermöglichen. Die Allgemeinheit der Norm im Vergleich zu den anderen Vorgehensmodellen soll als Vorteil gesehen werden. Die Arbeitsweise kann dadurch individuell gestaltet werden. Ein wesentlicher Vorteil ist die Iteration, die so oft stattfinden kann, solange die Gestaltungslösung die Nutzungsanforderungen nicht 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.

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.

Nach einer Abwägung der Vor- und Nachteile der Vorgehensmodelle, fällt die Wahl auf die Norm ISO 9241 Teil 210. Durch die vielen Vorteile der Norm, kann der Benutzer konkret analysiert werden. Der Benutzer wird in jedem Ent-

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.



Der erste Schritt besteht darin, dass der menschenzentrierte Gestaltungsprozess geplant werden soll. Dies beginnt mit dem Verstehen und Festlegen des Nutzungskontextes. 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. Zur Erstellung der User Profiles werden reale Testpersonen befragt. Anschließend sollen Personae erstellt werden, um die Merkmale der Testpersonen eindeutig darstellen zu können. 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 wird zwischen quantitatives und qualitatives Verfahren unterschieden. Das quantitative Verfahren beruht sich auf der Evaluation von Ergebnissen, die zahlenbasiert vermessen werden sollen. Im Gegensatz dazu geht es beim qualitativen Verfahren um die sprachliche Einschätzung von Testpersonen.

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.

Die Heuristiken bei der heuristischen Evaluation sind stark verallgemeinert, sodass 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. 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.

Da das System an sich wenig Funktionalitäten und Komplexität aufweist, ist der Cognitive Walkthrough von Polson, Lewis nicht als Evaluationsmethodik geeignet.

5 Kommunikationsmodell

5.1 Kommunikationsdiagramm

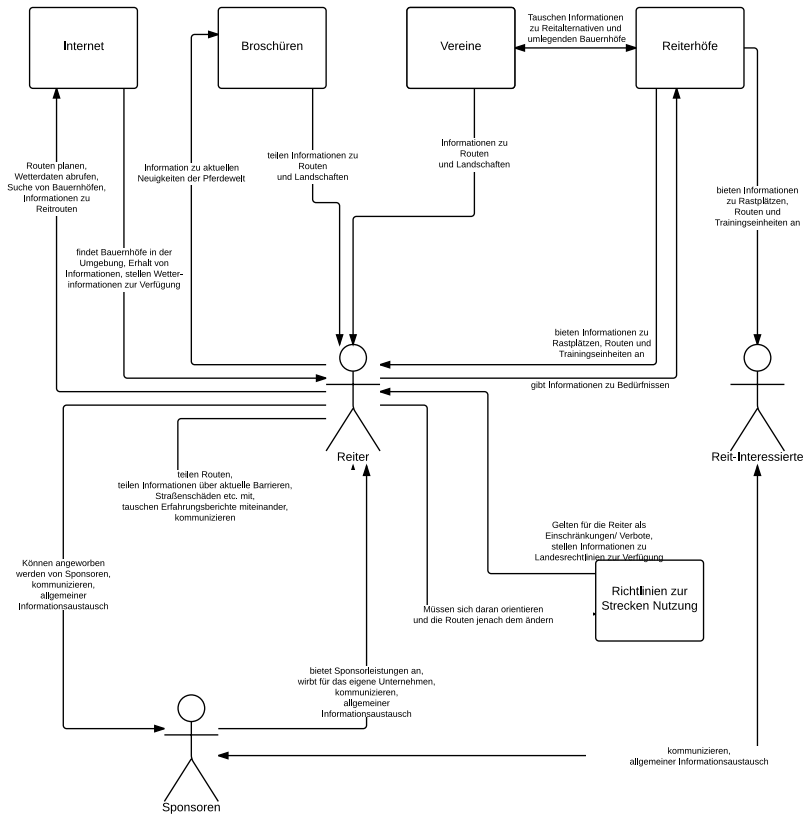


Abbildung .2: Deskriptives Kommunikationsdiagramm

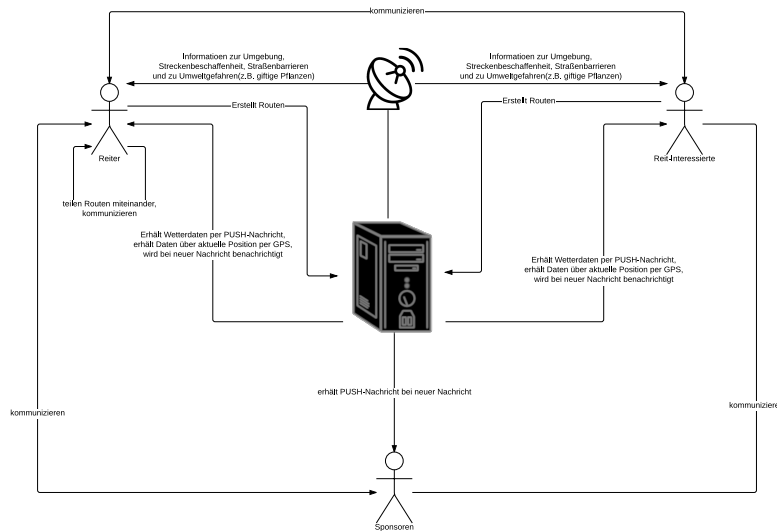


Abbildung .3: Präskriptives Kommunikationsdiagramm

Unter Betrachtung der Funktionen werden die Kommunikationsabläufe zwischen dem Nutzer und Funktionalitäten die in Anspruchgenommen werde.

5.2 "Route erstellen"

Wenn der Nutzer die Funktion "Route erstellen" nutzt, aktiviert sich automatisch das GPS und führt die direkte Ortung des aktuellen Standortpunkts aus. Durch die Lokalisierung startet automatisch die Wetter API, die dem Nutzer das Wetter anzeigt. Daraufhin wird der Nutzer aufgefordert sein Zielort einzugeben. Das GPS erstellt die Route und gibt bei Möglichkeit weitere Alternative-Routen mit aus. Der Nutzer wählt eine Route aus. Ist die gewünschte Route vom Nutzer ausgewählt wird über den Server Informationen zur Route aufgerufen und dem Nutzer angezeigt. Dem Nutzer werden allgemeine Informationen und Informationen zu Gefahren auf der Strecke wie zum Beispiel Strecken Hindernisse, befahrene Straßen, giftige Pflanzen, Rastplätze für Pferd und Mensch und diverse weitere Informationen, die den Ausflug auf der Route anzeigt, übermittelt, um den Ausflug so sicher wie möglich gestalten zu können. Der Nutzer startet selbst die Route und kann jederzeit die Route beenden. Während er die Route bestreitet kann der Nutzer über Push-Nachrichten automatisch über Gefahren und Unwetter informiert werden.

Möchte der Nutzer während er unterwegs ist und die Route angezeigt wird spontan rasten, kann er durch Anklicken eines angezeigten Rastplatzes diesen

wählen und das GPS Signal ändert das Zielort und erstellt eine Route zum gewünschten Ziel. Über den Aufruf neue Route und Änderung des Ziels erhält der Server eine neue Anfrage und gibt als Antwort Informationen zum Zielort.

5.3 "Neighbourhood"

Auf jedem Profil erscheint unter der Funktion Neighbourhood Routen von anderen Nutzer in der Umgebung. Diese werden auf dem eigenen Profil als automatische Push-Nachricht angezeigt. Jeder Nutzer kann seine Neighbourhood erweitern und seine Route teilen; ist aber nicht zwingend dazu verpflichtet seine Routen zuteilen. Bei Interesse an einer Route muss der Nutzer die Route anklicken und bekommt weitere Informationen zur Route. Des weiteren kann der Nutzer bei Unklarheiten, den Nutzer über Kommentarzeilen Texte schreiben. Diese wird anschließend über das GCM als Push-Nachrichten an den anderen Nutzer der die Route veröffentlicht hat weitergeleitet. Dieser hat die Möglichkeit dann auf die Fragen des Nutzers zu antworten.

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 eine andere Schnittstelle zur Umsetzung gewählt werden. Tritt dieses Risiko dennoch ein, können konkrete Alternativen, die in den Proof-Of-Concepts erarbeitet werden, den Erfolg des Projekts steuern.

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 das Projekt so weit wie möglich komprimiert 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.

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 hervorziehendem 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.

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 per Push-Notifications erhalten

Es soll getestet werden ob der Erhalt von Push-Notifications realisiert werden kann.

Das Testen gilt als erfolgreich, wenn der Erhalt von wichtigen Wetterdaten bei Nicht-Ausführung der Applikation ermöglicht wird. Das Fail-Kriterium wäre der Erhalt von Wetterdaten, jedoch nicht in Form von Push-Benachrichtigungen.

Ist die Push-Benachrichtigung nicht umsetzbar, soll darauf verzichtet werden.

Die Wetterdaten sollen durch die Anwendung direkt abrufbar sein.

7.2 GPS Lokalisierung ermöglichen

Die GPS Lokalisierung ist zwingend notwendig, um die Routenplanungen zu ermöglichen.

Das Exit-Kriterium ist bei dieser Funktion die Ortung und zugleich das Aufzeichnen einer Route. Das Fail-Kriterium wäre das Fehlschlagen von einem oder beiden Funktionen, die die Anwendung realisieren soll.

Eine Alternative ist für diese Problematik noch nicht vorhanden, da diese Funktion garantiert implementiert werden muss.

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.

Sowohl das Versenden als auch das Empfangen der Nachrichten sollte ermöglicht werden. Schlägt die Umsetzung fehl, soll die Funktion weg gelassen werden, da die Umsetzung nicht für den Erfolg der Anwendung notwendig ist. Dadurch kann Zeit für die programmierlastige Präzisierung der zuvor genannten 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.

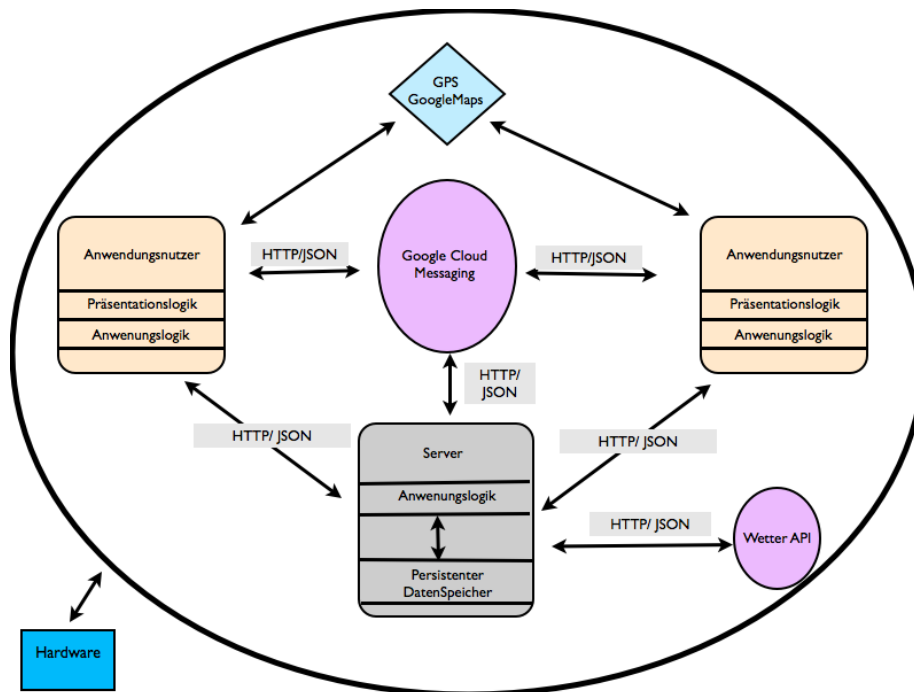


Abbildung .4: Architekturdiagramm zu dem System

8.1 Architekturbegründung

Das System soll eine Applikation auf Smartphones ermöglichen. Da das mobile Betriebssystem Android laut Statistiken (7) Marktführer im eigenen Segment ist, soll eine Android Applikation programmiert werden. 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.

Als Kommunikationsmittel und gleichzeitig als Push-Dienst soll Google Cloud Messaging (5) verwendet werden. Die Gründe hierfür beruhen sich auf der Tat-

sache, 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.

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 soll lediglich die Herkunft der Wetterdaten im Endprodukt sichtbar sein. Ein Nachteil ist, dass die kostenlose Schnittstelle auf 10000 Zugriffe im Monat eingeschränkt ist. Jedoch sollten die Zugriffe völlig ausreichen. Ein weiterer Grund für die Wahl der weather API ist, dass für die Anfragen an die Schnittstelle ein einfaches REST-Format verwendet wird. Als Datenaustauschformat kann XML benutzt werden.

Das Planen von Routen soll durch die Google Maps API geschehen. Die API ist weit verbreitet und stellt Karten für mobile Anwendungen zur Verfügung. Zudem ist die API völlig kostenlos und ermöglicht als Routenplaner 2500 Anfragen pro Tag. Karten können individuell angepasst werden. Das heisst, dass Daten wie zum Beispiel vorhandene Routen oder Bilder beliebig hervorgehoben werden können.(9)

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)

9 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	Meilenstein 1	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	Meilenstein 2	Zielhierarchie	Strategische Ziele	1h		2h	2h
			Taktische Ziele	1h		2h	2h
			Operative Ziele	1h		1h	1h
		related-works	ReiterApp	1h		0h	1h
			Cavallo-Retcoach	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	Benutzerzentrierter Gestaltungsprozess (© itemis AG, nach ISO 9241-210)	7
.2	Deskriptives Kommunikationsdiagramm	9
.3	Präkriptives Kommunikationsdiagramm	10
.4	Architekturdiagramm zu dem System	14
.5	Projektplan zu Meilenstein 1 und 2	16

Tabellenverzeichnis

10 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] Norm, Ergonomie der Mensch-System-Interaktion - Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010), Stand Januar 2011 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=52075 - 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 - 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/ - 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