



Fachhochschule Köln
Cologne University of Applied Sciences

Spiel 21

Dokumentation

im Rahmen der Veranstaltung
„Entwicklungsprojekt interaktive Systeme“

Sommersemester 2015

Autor: Eduard König
eduard.koenig@smail.fh-koeln.de

Projektbetreunde Prof. Dr. Gerhard Hartmann
Professoren: Prof. Dr. Kristian Fischer

Projektbetreuer: Franz Jaspers, B. Sc.
Sheree Saßmannshausen, B. Sc.

Abgabedatum: 29.06.2015

II Inhaltsverzeichnis

II Inhaltsverzeichnis	II
II Abbildungsverzeichnis	V
III Tabellenverzeichnis	VI
IV Listing-Verzeichnis	VI
I MEILENSTEIN 1	1
1 Einleitung	2
1.1 Problemraum	2
1.2 Lösungsansatz	2
1.3 Marktrecherche (related-works)	2
1.4 Alleinstellungsmerkmal	3
II MEILENSTEIN 2	4
2 Zielhierarchie	5
2.1 Strategische Ziele (langfristig)	5
2.2 Taktische Ziele (mittelfristig)	5
2.3 Operative Ziele (kurzfristig)	6
3 MCI Vorgehensweise	7
3.1 Wahl des Vorgehensmodells	7
3.1.1 User Centered Design	7
3.1.2 Usability Methoden	7
3.1.3 Zusammenfassung	9
3.2 Nutzungskontextanalyse	9
3.2.1 Stakeholderanalyse	10
3.2.2 User Profiles	11
3.2.3 Personas	12
3.2.4 Szenarien	12
3.3 Anforderungsermittlung	14
3.3.1 Produktperspektive	14
3.3.2 Produktperspektive	14
3.3.3 Funktionale Anforderungen	15
3.3.4 Nichtfunktionale Anforderungen (qualitative)	15
3.4 Hierarchische Task Analysis (HTA) deskriptiv	16
3.5 Ermittlung von Anwendungsfällen (Use Cases)	19
4 Risiken	19
4.1 Akzeptanz	19

4.2 Entwicklung	20
4.3 Datenschutz	20
4.4 Technisch	20
4.5 Projektintern	21
III MEILENSTEIN 3	22
5 Spezifikation der PoCs	23
5.1 Testen der Netzwerkstruktur	23
5.2 Nachrichtenaustausch (Chat Applikation)	23
5.3 Standort	24
5.4 Wetter	24
5.5 Matchmaking	25
6 WBA-Modellierungen	25
6.1 Kommunikationsmodell	25
6.1.1 Deskriptives Modell	26
6.1.2 Präskriptives Modell	27
6.1.3 Architekturdiagramm	28
6.2 Systemarchitektur	28
IV MEILENSTEIN 4	29
6.3 Server	30
6.3.1 Datensicherheit und Datenschutz	30
6.3.2 Middleware	30
6.3.3 Vorläufiges Matchmaking (Pseudo-code)	31
6.4 Datenmodell	31
6.5 Endgeräte (Client)	35
6.5.1 Android Anwendung	35
6.5.2 Drittanbieter	35
7 Prototypen UI	35
7.1 Papierbasierte Prototypen	35
V MEILENSTEIN 5	40
7.2 Evaluationsprozess	41
7.3 Weitere Erkenntnisse	43
8 Narratives Konzept für filmische Präsentation	44
VI MEILENSTEIN 6	46
9 Prozessassessment	47

10 Fazit	48
11 Installationsanleitung	49
11.1 NodeJS Server	49
11.2 MongoDB Datenbank	49
11.3 Android Anwendung	50
12 Projektplan	51
13 Quellenverzeichnis	53
Anhang	I
A Spielregeln	I
B Marktrecherche (related-works)	II
B.1 Wetterdienste	II
B.2 Nachrichtenaustausch (Messenger)	II
B.3 Navigation (GPS-Datenübertragung)	II
C Fragebogen	IV
D Personas	VII
E Szenarien	VIII
F Use Cases	IX
G Screenshots	XII

II. Abbildungsverzeichnis

Abb. 1	MCI Vorgehensweise	9
Abb. 2	HTA-Dekomposition für den Task: Spiel 21 spielen	17
Abb. 3	deskriptives Kommunikationsmodell	26
Abb. 4	präskriptives Kommunikationsmodell	27
Abb. 5	Systemarchitektur	28
Abb. 6	MongoDB Datenmodell	32
Abb. 7	Datenstruktur der Benutzer	33
Abb. 8	Datenstruktur der Basketballplätze	34
Abb. 9	Datenstruktur der Matches	34
Abb. 10	Datenstruktur der Ergebnisse	34
Abb. 11	UC1: Start der Anwendung	36
Abb. 12	UC2: Registrierung des Benutzers	36
Abb. 13	UC3: Navigation und Menü	36
Abb. 14	UC4: Spielstream	37
Abb. 15	UC5: Herausforderungen	37
Abb. 16	UC6: Gegenspieler wählen	37
Abb. 17	UC7: Basketballplatz auswählen	38
Abb. 18	UC8: Nachrichten	38
Abb. 19	UC9: Wetter	38
Abb. 20	UC10: Einstellungen	39
Abb. 21	UC11: Popup Benachrichtigung	39
Abb. 22	UC12: Spielergebnis eintragen	39
Abb. 23	UC4: Spielergebnisse	42
Abb. 24	UC5: Herausfordern	42
Abb. 25	UC8.1: Nachrichten	43
Abb. 26	UC8.2: Unterhaltung	43
Abb. 27	Das Dreieck zu Zeit, Qualität und Kosten	48
Abb. 28	Schematische Darstellung des Basketballfeldes	I
Abb. 29	Logo Preview	XII
Abb. 30	Farbspektrum	XII
Abb. 31	GUI: Start der Anwendung	XIII
Abb. 32	GUI: Registrierung des Benutzers	XIII
Abb. 33	GUI: Navigation und Menü	XIII
Abb. 34	GUI: Spielergebnisse	XIII
Abb. 35	GUI: Herausfordern	XIII
Abb. 36	GUI: Basketballplatz auswählen	XIII
Abb. 37	GUI: Gegenspieler wählen	XIV
Abb. 38	GUI: Nachrichten versenden	XIV
Abb. 39	GUI: Wetter	XIV
Abb. 40	GUI: Einstellungen	XIV

III. Tabellenverzeichnis

Tab. 1	Bezug Benutzer	11
Tab. 2	Bezug Städte/Gemeinde	11
Tab. 3	Bezug Anwohner	12
Tab. 4	Bezug Datenschützer	12
Tab. 5	Nutzungskontextanalyse und deren Merkmale	13
Tab. 6	Ressourcen mit HTTP-Verben	33
Tab. 7	Auswertung des Fragebogens	VI
Tab. 8	Persona Markus	VII
Tab. 9	Persona Anna	VII
Tab. 10	Use Case: 1	IX
Tab. 11	Use Case: 2	IX
Tab. 12	Use Case: 3	IX
Tab. 13	Use Case: 4	X
Tab. 14	Use Case: 5	X
Tab. 15	Use Case: 6	X
Tab. 16	Use Case: 7	X
Tab. 17	Use Case: 8	X
Tab. 18	Use Case: 9	XI
Tab. 19	Use Case: 10	XI
Tab. 20	Use Case: 11	XI
Tab. 21	Use Case: 12	XI

IV. Listing-Verzeichnis

Lst. 1	Vorläufiges Matchmaking (Pseudo-code)	31
--------	---	----

Teil I.

MEILENSTEIN 1

1. Einleitung

Das Konzept dient der Entwicklung der Anwendung, dies ist eine spezielle Variante des Basketballspiels und wird kurz „Spiel 21“ gesprochen. Die Anwendung soll als verteiltes System entwickelt werden und der Anforderungen der speziellen Domäne entsprechen. Das System stellt eine Anwendung dar, welche den Benutzern hilft andere Benutzer in der Umgebung zu finden und diese dann auf ein Spiel einzuladen. Dieses Dokument beschreibt das Konzept, Prozess- sowie die Systemdokumentation der geplanten Anwendung. Dies umfasst die detaillierte Beschreibungen welche im Laufe des Projekts vorgenommen wurden.

1.1. Problemraum

Einige Menschen halten sich gerne im freien auf und spielen das Spiel 21. Dies ist ein Wettkampf Spiel, man kennt es vielleicht noch aus dem Sportunterricht. Die Regeln werden im Anhang A erläutert.

Ein Wettkampf (Match) kann aber nur dann ausgetragen werden, wenn der öffentliche Basketballplatz nicht belegt ist und nur bei schönen Wetter. Es wird nicht gespielt, wenn es regnet, da der Basketball nass ist und die Spieler können schnell beim Ausrutschen Verletzungen hinzuziehen. Außerdem ist es nicht einfach neue Leute in neuer Umgebung zu finden. Deshalb wäre es schön Gleichgesinnte, beispielsweise nach dem baldigen Feierabend auf ein Match herausfordern.

1.2. Lösungsansatz

Die Benutzer sollten auf einfache Weise das Finden und das Einladen von anderen Benutzern ermöglicht werden. Während der Einladung können Wetterdaten von einem Wetterdienst abgerufen und ausgewertet werden. Außerdem könnte man den anderen Benutzern zeigen, dass der Basketballplatz zu gegebenen Zeitpunkt belegt ist.

Es soll das Versenden von Textnachrichten unterstützen umso die Kommunikation zwischen den Benutzern zu erleichtern. Außerdem sollte das System die Benutzer zu den gewünschten Basketballplatz navigieren können.

Nach dem Spielende werden die Spielergebnisse von den Benutzern gespeichert, so kann man diese beispielsweise später für eine Matchmaking verwenden um zukünftig gleichstarke Gegenspieler zu finden.

1.3. Marktrecherche (related-works)

Um die Alleinstellungsmerkmale der Anwendung zu bestimmten, wurde eine Marktrecherche durchgeführt (Anhang B), umso die positiven wie die negativen Teileaspekte der Konkurrenzprodukte zu kennen.

1.4. Alleinstellungsmerkmal

Es gibt viele Sportanwendungen auf dem Markt, doch aktuell gibt es keine Anwendung, die speziell für Basketballfreunde und in den Fall für das „Spiel 21“ gedacht ist. Anhand der erläuterten Konkurrenzprodukte wurden die verschiedenen Teilfunktionen betrachtet. Viele der Anwendungen bieten vorweg gute Teilfunktionen zur Erfüllung des Nutzungsproblems. Jedoch kein ähnliches Konzept für das System. Als indirekte Konkurrenz könnten die sozialen Netzwerke, wie beispielsweise Facebook oder Ähnliches sein. Hierbei lässt es sich nicht konkret sagen wie viele, wo und welche Gemeinschaften für das „Spiel 21“ existieren.

Teil II.
MEILENSTEIN 2

2. Zielhierarchie

Aus der formulierten Problemstellung werden verschiedene Ziele aufgestellt und näher erläutert. Im Folgenden erfolgt die Gliederung in strategische, taktische und operative Ziele, dies sind Maßnahmen ohne technischen Hintergrund.

2.1. Strategische Ziele (langfristig)

1. Eine Gemeinschaft für das „Spiel 21“ aufzubauen sowie die Gewinnung neuer Benutzer.
2. Eine digitale Plattform für das „Spiel 21“ etablieren.
3. Für Benutzer, die optimale Spielgegebenheiten kenntlich machen.
4. Die Manipulation von Spielergebnissen reduzieren um somit eine bessere Akzeptanz zu schaffen oder damit sich das Spiel besser etabliert.
5. Den beschränkten Basketballplatz optimal nutzen (max. 8 Spieler auf dem Basketballplatz).
6. In Kontakt treten mit anderen Benutzern.

2.2. Taktische Ziele (mittelfristig)

- (1) Benutzer durch virtuelle Erfolge, sogenannte Achievements belohnen umso mehr Motivation zwischen den Benutzern zu schaffen und dadurch die Gemeinschaft zu stärken.
- (1/6) Neue Benutzer aus der Umgebung durch die Anwendung kennenzulernen und mit denen später in Kontakt treffen zu können.
- (2) Die Akzeptanz des Systems erhöhen.
- (2) Die Geschwindigkeit der Arbeit mit dem neuen System maximieren.
- (3) Wettervorhersage in der öffentlichen Umgebung angezeigt und einen Ratschlag gegeben ob es optimal wäre in der gewünschten Zeit zu spielen.
- (4) Durch die gegenseitige Kontrolle der Benutzer, kann man die Manipulation unterdrücken.
- (5) Durch das System sollte es möglich sein die Anzahl der Benutzer auf den öffentlichen Basketballplätzen zu identifizieren.

- (6) Den Benutzern erlauben Nachrichten zu versenden, um so mit anderen Benutzern in Kontakt treten zu können.
- (6) Die Zielvoraussetzung ist ein Korb und ein Ball pro Match, in einem Match sind mindestens 2 bis 4 Spieler.

2.3. Operative Ziele (kurzfristig)

- (1) Die Anwendung soll durch das Matchmaking sich positiv auf die Gemeinschaft auswirken.
- (1) Durch die Recherche kann festgestellt werden, wie viele Besitzer in der Domäne ein Smartphone mit Internetzugang haben.
- (2) Die Ausführung der notwendigen Schritte einer Funktion und die benötigte Zeit dafür minimieren.
- (2) Die Navigation sollte den Benutzern helfen einen bestimmten Basketballplatz zu finden.
- (2) Die Benutzer sollten den Weg zum Basketballplatz finden durch die Navigation oder durch ihr vorhandenes Wissen über die Umgebung. (Auto oder Fuß?)
- (3) Durch die Einbindung der Wetterinformationen kann das System genau vorhersagen ob ein Match ausgetragen werden kann oder nicht, denn beim Regen wird nicht gespielt.
- (3/5) Durch die Anmeldung der Benutzer im System kann man bei den gewählten Basketballplatz die Vakanz einsehen (der reale Tatbestand bleibt jedoch unberührt).
- (3) Durch die Eingabe des Benutzers kann bestimmt werden, ob der Spieler ein Profi oder ein Anfänger ist.
- (4) Durch das Versenden von negativen Feedback über den Benutzer, kann man die Manipulation der Spielergebnisse reduzieren.
- (6) Den Benutzern eine Chatfunktion anbieten, um so den Nachrichtenaustausch zu gewährleisten.
- (6) Durch Befragungen in der Domäne kann festgestellt werden, wie viele Benutzer bereit wären mit unbekannten Benutzer(n) zu spielen.
- (6) Die Benutzer sollten sich einigen beim Match, wer den Basketball zum Platz bringt.

3. MCI Vorgehensweise

Nachdem im Folgenden die Zielhierarchie und Marktanalyse durchgeführt worden ist, werden nachstehende methodologische Herangehensweise im Bereich Mensch-Computer-Interaktion vorgestellt. Die Überlegungen dienen später für das zu wählende Vorgehensmodell.

3.1. Wahl des Vorgehensmodells

Die Anwendung wird für den Freizeitbereich entwickelt. Es sollte daher ein hohes Maß an Gebrauchstauglichkeit aufweisen, damit es von der Gemeinschaft akzeptiert wird.

Daher sollten die Merkmale der Benutzer als zweckmäßiger Ausgangspunkt für das Konzept und die Entwicklung des interaktiven Systems betrachtet werden. Hier werden die verschiedenen Ansätze gegeneinander abgewogen und später im Fazit festgelegt.

3.1.1. User Centered Design

Dies ist keine konkrete Arbeitsmethode, sondern eine Grundhaltung, die sich darin äußert, dass der Benutzer in allen Phasen der Entwicklung mit einbezogen wird. Dadurch wird bestrebt, eine hohe Gebrauchstauglichkeit (Usability) zu erreichen.

Gründe die für die Einbeziehung der Benutzer sind meistens die verschiedenen Perspektiven. Der eine Entwickler hat nicht die gleichen Perspektiven wie ein Benutzer selbst. Deshalb wird ein Entwickler nicht schaffen die Perspektiven der Benutzer zu verbessern ohne seine Hilfe. Ein anderer Grund ist die Notwendigkeit, Produktqualität und Gebrauchstauglichkeit in einem Gesamtprozess als wichtige, aber eigenständige Aufgaben und Prozesse zu betreiben. So sollte der Benutzer mehr Aufmerksamkeit auf seine Tätigkeit werfen und nicht etwa von System ablenken lassen.

Bei der Entwicklung der interaktiven Systeme gehört ein iteratives Vorgehen mit Design, Prototypen- und Evaluationsprozess. Die Grundhaltung der Aktivitäten ist gut, da der Fokus auf den Benutzern selbst liegt und nicht etwa auf dem Nutzen des Systems. Außerdem gibt es hierzu verschiedene Rahmenbedingungen, die das Projekt maßgeblich beeinflussen.

3.1.2. Usability Methoden

Es stehen mehrere Modelle, mit unterschiedlichen Methoden und Techniken zur Auswahl, die den benutzungszentrierten Gestaltungsprozess unterstützen. Folgende Methoden kommen für das Projekt zum Einsatz:

Discount Usability Engineering Dieser Ansatz stammt von Jakob Nielsen[26]. Dies stellt eine schnelle und kostengünstige Überprüfung zur Verbesserung der Gebrauchstauglichkeit

mit den folgenden drei Techniken dar.

Bei den Szenarien (storyboard), stellt sich als Ziel, potenzielle Probleme mit der Usability zu identifizieren. Für das Projekt stellt sich als Nachteil heraus, dass keine Vollständigkeit gewährleistet wird und mit großen Zeitaufwand verbunden ist, deshalb wird es verworfen. Die Gestaltungslösungen (paperbased prototypes) haben eine geringe Vorläufigkeit und dienen nur zur Auflistung der Anforderungen. Für das Projekt stellt es insgesamt eine gute Lösung dar und somit wird ein erster visueller Eindruck über die Anwendung gewonnen. Hierbei existiert noch eine vereinfachte Technik (think aloud) das „laute Denken“. Diese Technik ist gut, da man die wichtigsten Usability-Probleme identifizieren kann.

Außerdem gibt es den letzten Punkt, Evaluation durch Heuristik (heuristic evaluation)[25]. Nach Nielson , können hier alle erfolgreich angewendet werden. Aber dies kann nicht von nur einer Person durchgeführt werden, da man sonst nie alle Fehler in einem System entdecken würde. Für das Projekt stellt dieses Modell eine mögliche Lösung dar. Dennoch würde die Ausarbeitung der Szenarientypen viel Zeit kosten und viele Nachteile in anderen Bereichen des Projekts mitbringen.

Aufgrund der finanziellen und zeitlichen Rahmenbedingungen wird die Vorgehensweise des Discount Usability gewählt.

Benutzerprofile (user profiles) Die Benutzerprofile stellen eine Liste von Merkmalen, samt Merkmalausprägungen zur Verfügung, welche für die Entwicklung benutzt werden. Der Entwickler kann sich so eine bessere Perspektive über weitere Benutzer schaffen. Hierbei wird iterativ geprüft und dokumentiert, wie der Benutzer mit der Anwendung zurechtkommen würde.

Für das Projekt stellt dieses Modell eine mögliche Lösung dar. Es bietet sich an, diesen Prozess also iterativ und über die gesamte Entwicklungsphase laufen zu lassen. Es würde hier die Ausarbeitung und die empirische Datenerhebung viel Zeit kosten. Beispielsweise durch Interviews, Umfragen, Feldstudien, Beobachtungen, etc. Dies könnte man bei dem Projekt begrenzen, beispielsweise auf Umfragen der Benutzer um somit die Zeitressource zu sparen.

DIN-Norm EN ISO 9241 (Teil 110/210) Die DIN-Norm mit dem Teil 110[21] stellt die Grundsätze der Dialoggestaltung dar, dies könnte bei dem Projekt sich als sehr nützlich Erweisen, beispielsweise bei der Entwicklung der Anwendung im Designbereich. Diese sind in sieben Anforderungen gegliedert und stellen keine Einschränkungen für das Projekt dar. Leider können aus zeitlichen Gründen nicht alle Anforderungen der DIN-Norm berücksichtigt werden.

Der Teil 210[22] stellt den Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme

dar. Hierbei handelt es sich um ein Rahmenwerk. Nachteil für dieses Projekt ist, dass keine Techniken oder Methoden zur Verfügung gestellt werden. Dies sollte möglichst mit anderen Modellen zur Vervollständigung beitragen. Dennoch bietet die Norm eine stabile Grundlage und sollte auf jeden Fall für das Projekt berücksichtigt werden.

3.1.3. Zusammenfassung

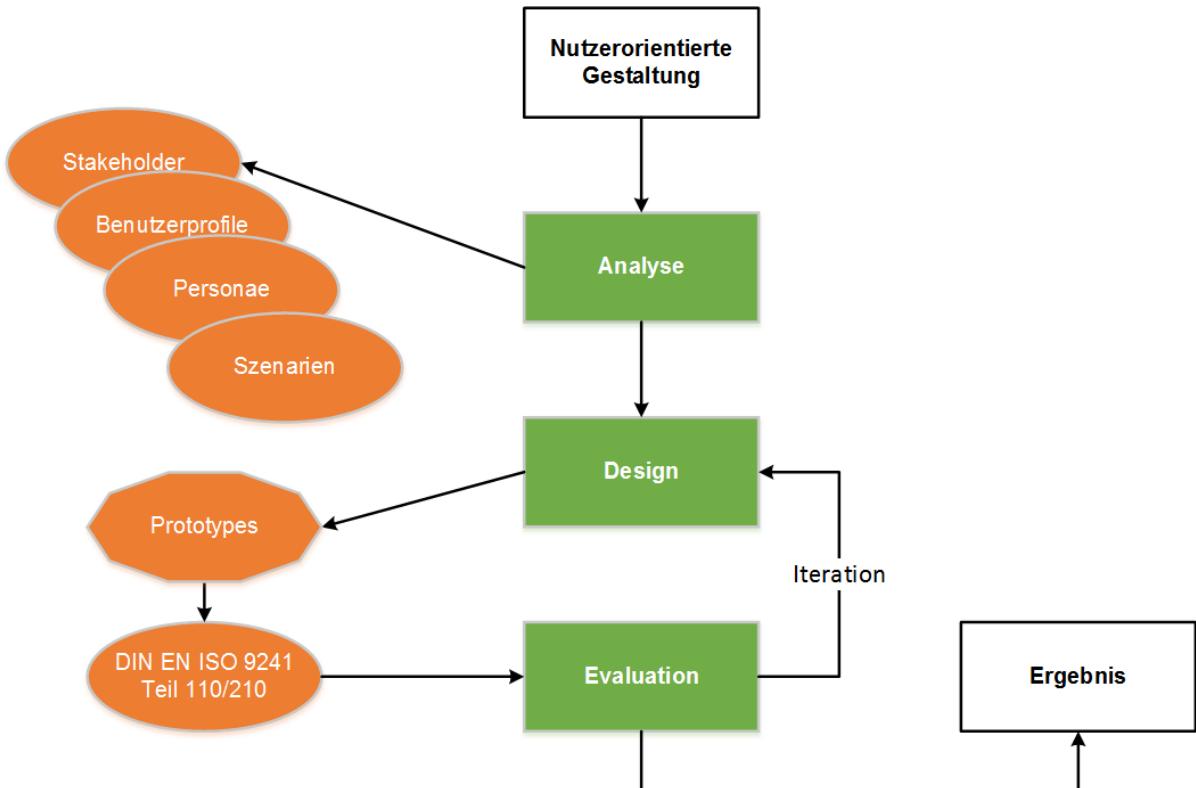


Abbildung 1: MCI Vorgehensweise

Eine benutzerorientierte Methodik sollte bereits in der Entwicklungsphase wichtige Lösungen aus der Benutzersicht liefern und später damit ein technisches Rahmenwerk schaffen. Als Grundhaltung sollte User Centered Design genommen werden mit der Kombination aus Discount Usability und den vorgestellten DIN-Normen.

Für das Projekt wurden bereits die Evaluationsmethoden abgewogen und die besten Lösungsansätze für das Projekt genannt. Durch die Iteration sollte man später zu einem Ergebnis gelangen, welches ein hohes Maß an Gebrauchstauglichkeit aufweist.

3.2. Nutzungskontextanalyse

Um bei der Nutzungskontextanalyse[22] mehr Verständnis in der Domäne zu bekommen, wurde eine Internetrecherche durchgeführt. Leider ist die Domäne so speziell, dass es kaum Informationen dazu gibt. Die einzige Erkenntnis sind die vielfältigen Regeln bei dem „Spiel

21“. Leider war es nicht möglich, Spieler aus der Szene zu sprechen.

Um dennoch Daten von Personen zu bekommen, werden Fragebögen verteilt, um so die Erfahrungen und die Meinungen der Personen als Ergebnisse auswerten zu können (Anhang C).

Dabei ist der Entwurf des Fragebogens sehr wichtig für die Qualität und die Beschaffenheit der Aussagen. Die Ergebnisse, der befragten fließen später in die Benutzermodellierung mit ein und begründen somit die wesentlichen Unterschiede zwischen den Benutzern.

Im Grunde wird hier die physiologischen Merkmale, Wohnortgegebenheiten sowie das Interesse an Freizeit abgefragt. Leider kann man zwischen einen Anfänger und Fortgeschrittenen nicht unterscheiden, da kein Maßstab gesetzt worden ist.

Die Analyse von Benutzer und Kontext hat ergeben, dass das Alter der befragten Personen im Durchschnitt bei 27 Jahren liegt. Die meisten Personen waren männlich. Es stand nur ein begrenzter Befragtenkreis zur Verfügung, deshalb kamen die meisten befragten aus Gummersbach und sind Studenten. Hier ist aber interessant zu betrachten, dass die meisten Personen entweder länger als 5 Jahre in den selben Ort wohnen oder sind ganz neu hinzugezogen. Dabei spielt die Transportmöglichkeit eine wichtige Rolle und die meisten besitzen ein Auto.

Was die technischen Merkmale angeht, so hat jeder ein internetfähiges Endgerät, beispielsweise ein Smartphone und besitzt hierbei eine Flatrate. Des Weiteren hat schon jeder Dritte nach Freizeitmöglichkeiten im Internet gesucht.

Im Durchschnitt haben die befragten Personen 26,15 Stunden in der Woche an Freizeit und sind sehr aktiv, deshalb kommt selten langweile auf. Wenn es um die Sportart geht, so gehen hier die Meinung weit auseinander und nur wenige beschäftigen sich mit Basketball, die Meisten trainieren lieber im Fitnessstudio. Außerdem kennen 15 von 32 Befragten die spezielle Variante des Basketballspiels.

Die gewonnenen Erkenntnisse werden eingesetzt und dienen damit zur Ermittlung der Anforderungen und tragen zur neuen Lösungen bei. Im folgende werden zunächst die Benutzer identifiziert.

3.2.1. Stakeholderanalyse

Im Falle der Stakeholderanalyse[22] konnten folgende Stakeholder anhand der Faktoren aus der DIN Norm infiziert werden:

- Benutzer: Spieler.
- Städte/Gemeinde: Ansehen der Gemeinde.
- Anwohner: Lärmelastigung.
- Datenschützer: sind besorgt um die missbräuchliche Datenverarbeitung.

Im Weiteren werden die Stakeholder begutachtet und weiter veranschaulicht anhand der Merkmale.

Bezug Benutzer	Objektbereich der Beziehung
Anrecht	Privatsphäre
Anteil	Speicherung von Endergebnissen. Auswertung von Statistiken. Erstellung / Einladung von Herausforderungen (Matches). Unterstützung der Anwendung durch Mundpropaganda.
Anspruch	Erhalt von Wettertipps. Erhalt von Basketballplatzinformationen, beispielsweise über den Zustand belegt (Ja/Nein). Erhalt von passenden Rivalen.
Interesse	Zum Basketballplatz zu gelangen, navigieren lassen.
Erfordernisse und Erwartungen	Nicht beim Regen zu spielen. Wissen, wie viele Benutzer auf den Basketballplatz sind. Mit anderen Benutzern in Verbindung treten, bspw. Chat. Nächstgelegenen Basketballplatz in der Nähe herausfinden. Passende Spieler zu finden. Geeignetes Endgerät um die Anwendung nutzen zu können.

Tabelle 1: Bezug Benutzer

Bezug Städte/Gemeinde	Objektbereich der Beziehung
Anrecht	Ansehen der Gemeinde.
Anteil	–
Anspruch	–
Interesse	Eine Gemeinschaft entstehen zu lassen, die das Ansehen der Umgebung bzw. des Stadtteils fördert.
Erfordernisse und Erwartungen	Steigerung des Freizeitangebotes. Friedlicher Ablauf.

Tabelle 2: Bezug Städte/Gemeinde

3.2.2. User Profiles

Nachdem die Stackeholder bekannt sind, werden nun diese detaillierter charakterisiert. Hierbei werden die Benutzer in Jugendliche und Erwachsene unterteilt, da diese nicht die gleichen Merkmale aufweisen.

Bezug Anwohner	Objektbereich der Beziehung
Anrecht	Privatsphäre, beispielsweise keine Abbildung der nächstgelegenen Häuser.
Anteil	–
Anspruch	–
Interesse	Sicherheit der Umgebung. Keine Lärmbelästigung.
Erfordernisse und Erwartungen	Sicherheit der gespeicherten Daten.

Tabelle 3: Bezug Anwohner

Bezug Datenschützer	Objektbereich der Beziehung
Anrecht	Privatsphäre
Anteil	–
Anspruch	–
Interesse	Sicherheit der Datenverarbeitung.
Erfordernisse und Erwartungen	Sicherheit der gespeicherten Daten.

Tabelle 4: Bezug Datenschützer

3.2.3. Personas

Bei den Personas nach Alan Cooper[20] handelt es sich um prototypische Benutzer und es stellt nur eine Maske dar, welche ihre unterschiedlichen Ziele, Verhalten und deren Eigenschaften darstellt. Deshalb ist es wichtig, diese lebensecht wirken zu lassen. Im folgenden werden zwei Personas beschrieben, die für zwei unterschiedliche Benutzergruppen stehen (Anhang D).

Während Markus ganz viel Freizeit hat und gerne was mit seinen Freunden unternimmt, arbeitet die Anna als Sachbearbeiterin in Vollzeit und hat daher weniger Freizeit. Die Personas zeigen die wichtigen Eigenschaften und die Bedürfnisse der Benutzer in Hinblick auf das geplante System.

3.2.4. Szenarien

Um die Perspektive der Benutzer einnehmen zu können, wurden die Personas und Szenarien erstellt. Nun kann man aus deren Sicht diskutieren und aus deren Entwurf eine neue Lösung schlagen für das zukünftige System. Als erste Prototypen können die Szenarien für das Projekt dienen. Im folgenden werden einige Problemszenarien und daraus die entstehenden User Needs beschrieben (Anhang E).

BENUTZER	Jugendlicher	Erwachsener		
Arbeitsaufgaben				
<ul style="list-style-type: none"> - Prüfung der Wetterdaten in Kombination mit der Verfügbarkeit von den Basketballplätzen. - Herausforderung der anderen Benutzer. - Eintragen der Spielergebnisse. 				
Ausrüstung				
Hardware:	Smartphone mit Internetzugang und GPS Verbindung			
Software:	die Anwendung „Spiel 21“			
Materialien:	Basketballplatz, Basketball			
Physische Umgebung				
Alter:	14-18 Jahre	18-60 Jahre		
Geschlecht:	männlich / weiblich	männlich / weiblich		
Beförderung:	eingeschränkt (Fuß, Fahrrad)	uneingeschränkt (Fuß, Auto,..)		
Körperlicher Zustand:	uneingeschränkt	uneingeschränkt		
Soziale Umgebung				
Berufliche Status:	Schüler, Auszubildender	Erwerbstätig o. ä.		
Herkunft:	uneingeschränkt	uneingeschränkt		
Religion:	uneingeschränkt	uneingeschränkt		
Psychologische Umgebung				
Geistiger Zustand:	uneingeschränkt	uneingeschränkt		
Grundkenntnisse				
Umgang mit Smartphone:	Grundkenntnisse (Herunterladen von Anwendungen, Installation und Konfiguration) erforderlich			
Motivation zur Nutzung				
<ul style="list-style-type: none"> - Neue Leute kennenlernen. - Abfragen der Wetterinformationen. - Abfragen von nächstgelegenen Basketballplätzen. - Abfragen von der Verfügbarkeit des Basketballplatzes. 				

Tabelle 5: Nutzungskontextanalyse und deren Merkmale

Szenario 1: Es gibt offensichtlich Personen, die gerne den Tag sportlich abklingen lassen. Außerdem kommt Langweile auf durch die monotone Sportart. Die Anwendung könnte neue Leute zusammen bringen und so eine neue Gemeinschaft entstehen lassen.

Szenario 2: Es gibt Leute, die sich über die Wettervorhersage nicht informieren und haben keinen Spaß beim schlechten Wetterbedingungen zu spielen. In der Anwendung könnten Warnungen ausgeben werden, beim Schlechtwetter. Außerdem gibt es Leute, die kein gutes Erinnerungsvermögen haben, dies könnte in der Anwendung unterstützt werden durch die persistente Speicherung.

Szenario 3 und 5: Es gibt offensichtlich Leute, die Andere für Basketballspielen begeistern wollen. Speziell die Jugendlichen kennen das Spiel aus dem Sportunterricht. In der Anwendung könnten das unterstützt werden, dass man andere Leute zu dem Spiel einlädt und begeistert. Außerdem könnten die gesammelten Spielergebnisse und die laufenden Spiele für jeden sichtbar sein.

Szenario 4: Es gibt Leute, die relativ neu in der Umgebung sind und haben mangelnde Kenntnisse über die Umgebung. Außerdem wollen einige sich sportlich aktiv halten ohne gleich eine Mitgliedschaft in Vereinen oder sonstigen Einrichtung abschließen zu müssen. Die Anwendung könnte bei der Navigation den Benutzer unterstützen. Es sollten keine Verträge oder Verpflichtungen mit der Anwendung abgeschlossen werden.

3.3. Anforderungsermittlung

In der Anforderung[23] ist eine Software zu erstellen.

3.3.1. Produktperspektive

Die Anwendung sollte die Planung und Durchführung von Matchen sowie die Navigation zum bestimmten Basketballplatz unterstützen soll. Wetterinformationen werden ausgewertet und an die Benutzer weiter geleitet sowie weitere Information über die Belegung des Basketballplatzes. Es wird ein Endgerät mit dem Android-Betriebssystem benötigt mit der passenden Anwendung. Es wird kein spezielles Fachwissen für die mobilen Endgeräte in der Domäne benötigt.

3.3.2. Produktperspektive

Für das Projekt wird das System vorerst nur deutschlandweit eingesetzt als zu testende Anwendung, um die mangelnde Akzeptanz auszuschließen und um letztendlich die Ressourcen zu sparen. Folgende Ressourcen sollten betrachtet werden: Kosten für die Übersetzer, Zeit für die mehrsprachige Programmierung und die Relevanz der speziellen Domäne. Des Weiteren ist eine zentrale Datenhaltung vorgesehen, um die Wartung des Systems zu erleichtern.

Zielgruppe des Produkts sind ausschließlich die Benutzer der Anwendung „Spiel 21“ selbst, die als Gemeinschaft agieren. Die Anwendung bezieht sich auf den Freizeitbereich. Außerdem wird bei dem Projekt in Erwägung gezogen, das Schalten von Werbung, dadurch sollten die laufenden Serverkosten finanziert werden. Im Folgenden werden die Anforderungen identifiziert und erläutert.

3.3.3. Funktionale Anforderungen

- F10** Der Benutzer muss die Möglichkeit haben sich im System identifizieren.
- F20** Der Benutzer soll die Wetterinformationen von seinem Bestimmungsort abrufen können.
- F30** Der Benutzer kann sich über die vorhandenen Basketballplätze im System informieren.
- F40** Der Benutzer kann andere Benutzer zum Match herausfordern.
- F50** Das System soll Benutzern Routen zum Basketballplatz bieten.
- F60** Der Benutzer sollte mit anderen Benutzern in Kontakt treten können.
- F70** Der Benutzer kann den Verlauf der gespielten Spiele ansehen.
- F80** Der Benutzer kann die Spielergebnisse notieren bzw. erfassen und diese einsehen.
- F90** Das System bietet die Möglichkeit in der Organisation des Spiels den Basketballträger zu bestimmen.
- F100** Das System soll den Benutzer automatisch passende Rivalen vorschlagen, bezieht sich auf F130.
- F110** Das System soll den Benutzer automatisch warnen, wenn die Wahrscheinlichkeit auf Regen sehr hoch ist bei einem bevorstehenden Match.
- F120** Das System sollte die Spielergebnisse der Benutzer persistent speichern.
- F130** Das System sollte den Standort des Benutzers lokalisieren.

3.3.4. Nichtfunktionale Anforderungen (qualitative)

- Q10** Die Anwendung soll eine hohe Gebrauchstauglichkeit aufweisen.
- Q20** Die Anwendung soll stets über die aktuellen Daten verfügen bis zum nächsten Spiel.
- Q30** Gute Navigierbarkeit durch das GPS, soweit in der GPS Verarbeitung möglich ist.
- Q40** Die erhobenen Daten über die Benutzer sollten anonymisiert behandelt werden.
- Q50** Die Manipulation von Spielergebnissen reduzieren.

Weitere Anforderungen können im Laufe des Projektes ermittelt werden.

3.4. Hierarchische Task Analysis (HTA) deskriptiv

Auf das Projekt bezogen nehmst man die Aufgabe bzw. Absicht das Spiel zu spielen und betrachtet diese, um in Anschluss die Möglichkeiten für das System zu betrachten. Dieser Ansatz wurde von Shepherd[27] definiert.

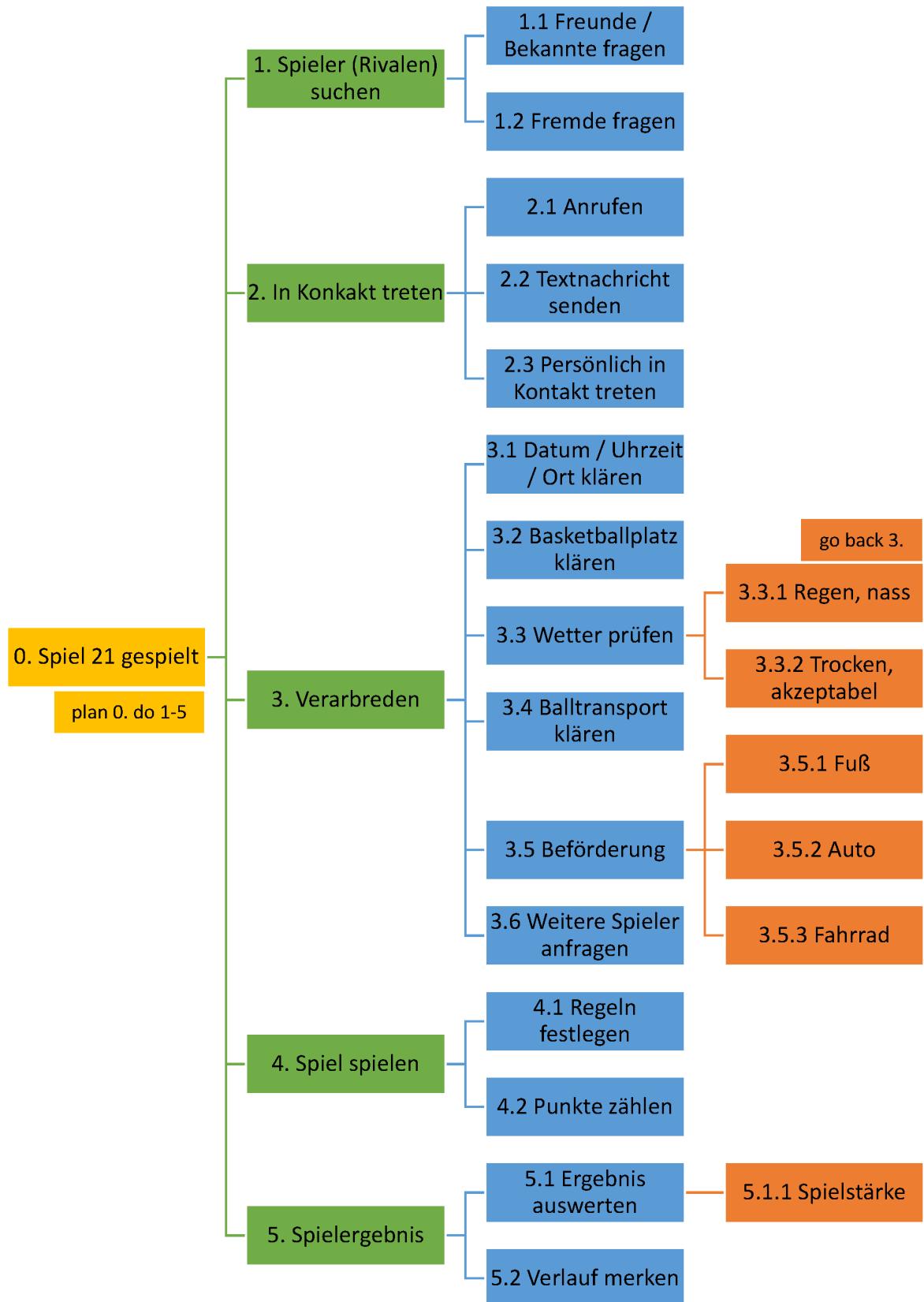


Abbildung 2: HTA-Dekomposition für den Task: Spiel 21 spielen

0. Spiel 21 gespielt
1. Spieler (Rivalen) suchen
 - 1.1 Freunde / Bekannte fragen
 - 1.2 Fremde fragen
2. In Kontakt treten
 - 2.1 Anrufen
 - 2.2 Textnachrichten senden
 - 2.3 Persönlich in Kontakt treten
3. Verabreden
 - 3.1 Datum / Uhrzeit / Ort klären
 - 3.2 Basketballplatz klären
 - 3.3 Wetter prüfen
 - 3.3.1 Regen, nass
 - 3.3.2 Trocken, akzeptabel
 - 3.4 Balltransport klären
 - 3.5 Beförderung
 - 3.5.1 Fuß
 - 3.5.2 Auto
 - 3.5.3 Fahrrad
 - 3.6 Weitere Spieler anfragen
4. Spiel spielen
 - 4.1 Regeln festlegen
 - 4.2 Punkte zählen
5. Spielergebnis
 - 5.1 Ergebnis auswerten
 - 5.1.1 Spielstärke
 - 5.2 Verlauf merken

Dies ist ein Beispiel und zeigt aus der HTA entstandenen Dokumentationen, wie die Durchführung einer HTA mit den Aufgabendiagramm (Abb. 2) und die dazugehörige Planliste.

Pläne:

- Plan 0: mach 1 – 2 – 3 – 4 – 5 in dieser Reihenfolge
- Plan 1: mach 1.1 – 1.2 in beliebiger Reihenfolge
- Plan 2: mach 2.1 – 2.2 – 2.3 in beliebiger Reihenfolge
- Plan 3: mach 3.1 – 3.2 – 3.3 – 3.3.2 – 3.4 – 3.5 in dieser Reihenfolge,
wenn gewünscht: mach 3.6, wenn 3.3.1 dann EXIT oder 3.1
- Plan 4: mach 4.1 – 4.2 in dieser Reihenfolge
- Plan 5: mach 5.1 – 5.2 in beliebiger Reihenfolge

Die Aufgabe wurden auf immer kleiner Teilaufgaben heruntergebrochen, um ein detailliertes Verstehen von den Nutzungsarten und den Stufen zu erlangen.

3.5. Ermittlung von Anwendungsfällen (Use Cases)

Aus den Anforderungen werden die Anwendungsfälle (Anhang F) ermittelt. Diese werden später bei der Gestaltungslösungen mit einbezogen. Die Anwendungsfälle beschreiben im Grunde die „geplanten“ Funktionen einer Anwendung gegenüber der „realen“ Welt[19]. Hier ist der Primärakteur immer der Benutzer und der Umfang (Scope) bezieht sich auf die Android-Anwendung, mit der Aufgabe das Spiel 21 zu spielen.

4. Risiken

In den meisten Fällen gefährden die Projektrisiken den Erfolg des Projektes. Diese sind aber die Grundlage für die Projektplanung und werden im Allgemeinen zur Identifikation und Bewertung eingesetzt, um frühzeitig diese Erkennen und vermeiden zu können. Die typischen Risiken bestehen meist aus der mangelnden Akzeptanz der Benutzer und die spät entdeckten Anforderungen im System.

4.1. Akzeptanz

Sollte die Anwendung von wenigen Benutzern genutzt werden, so entsteht ein hohes Risiko für die Anwendung selbst. Durch die fehlenden Daten, die zu qualitativ hochwertigen Statistiken und Rankingergebnissen führen könnten, kann kein Matchmaking stattfinden und somit keine geeigneten Gegenspieler für das Spiel. Folglich kann keine Gemeinschaft in dem Freizeitbereich entstehen. Anderes Szenario wäre die Unzuverlässigkeit der Anwendung. Hier würde der Benutzer im schlechtesten Fall die Anwendung überhaupt nicht mehr benutzen.

4.2. Entwicklung

Allein die Entwicklung des Systems verschlingt viel Zeit und Geld. Dies sollte generell bei jedem internen Datenverarbeitungssystem in Betracht gezogen werden. Die neuen und wenig bekannten Technologien bringen eine Herausforderung mit sich, beispielsweise die Einarbeitungsphase. Mögliche Lösung ist, ein Konzept zu entwickeln, wo die ganzen Problematiken, Techniken und Methoden festgehalten werden um die Entwicklung des Systems zu ermöglichen oder eine bessere Erkenntnis erlangen.

4.3. Datenschutz

Eventuell befürchten einige Benutzer die Speicherung der Routen durch das Navigationsfeature. Sollte man drauf verzichten, so kann keine Navigation stattfinden. Außerdem entstehen Sicherheitslücken durch den Entwicklungsprozess der Anwendung, wenn die Sicherheitsvorgaben nicht ausreichen berücksichtigt werden. Aber auch durch Sicherheitslücken im System oder in der Anwendung können Dritte an personenbezogene Daten herankommen und für eigene Zwecke missbrauchen.

Eine mögliche Lösung ist das GPS auf dem Smartphone auszuschalten. Das Abspeichern und Verarbeiten der Informationen sollte bei der Anwendung auf das Minimum reduziert werden. Bei den Sicherheitslücken im System oder der Anwendung müssen Updates durchgeführt werden um diese mit einem Patch bzw. Update zu schließen.

4.4. Technisch

Auf dem Smartphone laufen verschiedene Anwendungen, die die Batterie des mobilen Endgerätes belasten. Besonders durch das Aktivieren der GPS-Funktionalität, wodurch die Wetterinformationen und das Kartenmaterial für die Navigation geladen werden.

Eine Lösung ist es, den Benutzern zu erlauben durch die manuelle Eingabe ihren Standort auswählen zu können. Besonders interessant für die Benutzer, die sich in der Umgebung auskennen und wissen wo die jeweiligen Basketballplätze zu finden sind. Außerdem kann ein Ausfall von Drittanbietern, die Funktionalität der Anwendung beeinträchtigen. So beispielsweise können keine Wetterinformationen abgerufen werden und keine Navigation stattfinden.

Eine mögliche Lösung für diese Problematik wäre, dann die Daten des Wetterdienstes auf eigenen Servern zwischen zu speichern und bei den Navigationsdienst auf einen alternativen Drittanbieter zu setzen.

4.5. Projektintern

Durch die Bedrängnis der einzuhaltenden bevorstehenden Termine wird geradezu ein Zeitdruck aufgebaut. Außerdem könnte der Ausfall der Programmierer bzw. der Teammitglieder, beispielsweise durch eine Erkrankung o. ä. das Projekt zum Scheitern bringen.

Teil III.
MEILENSTEIN 3

5. Spezifikation der PoCs

Bei Proof of Concept (Machbarkeitsnachweis) werden die Funktionalitäten des Systems geprüft und konkrete Bedingungen sowie konkrete Alternativen beschreiben und anschließen prototypisch umgesetzt.

5.1. Testen der Netzwerkstruktur

Risiko: Asynchrone Datenverarbeitung zwischen der Android-Anwendung und NodeJS-Server. Anschließend sollten die Daten in die Datenbank gespeichert werden, beispielsweise ein einfaches Login.

Ziel: Es sollte eine Android-Anwendung prototypisch geschrieben werden, welche mittels POST-Methode die JSON-Daten von der Anwendung an den Server sendet und diese dort in der Datenbank vergleicht, beispielsweise ein Login.

Folgende Schritte sollten ergriffen werden:

- Eine Android-Anwendung sollte erstellt werden.
- Die Anwendung sollte auf einem Testgerät ausführbar sein.
- Der Server sollte JSON-Daten empfangen und diese persistent speichern können.
- Allgemein: das Erlernen der Android Entwicklungsumgebung.

Exit / Fail Kriterium: Nur wenn der Ausbildungs- und Lernaufwand in den Projektrahmen minimal gehalten werden und somit ein passables Ergebnis erzielt wird, gilt die Prüfung als bestanden.

Alternative Fallback: Leider führt hier kein Weg vorbei, man sollte sich die Zeit nehmen und sich mit den verschiedenen Entwicklungstools beschäftigen. Als Informatiker sollte man sich mit den neuen Technologien aneignen und sich weiter entwickeln.

Status: Die Entwicklungsumgebung „Android Studio“ wurde eingerichtet. Die ersten Schritte wurden ergriffen und es ist gelungen die Kommunikation zwischen den NodeJS-Server und der MongoDB-Datenbank herzustellen inklusive Datenaustausch. Anschließend wurde eine prototypische Android-Anwendung geschrieben und auf dem Testgerät erfolgreich getestet.

5.2. Nachrichtenaustausch (Chat Applikation)

Risiko: Es besteht zur Zeit keine Möglichkeit mit den Benutzern in Verbindung zu treten um unvorhersehbare Ereignisse zu diskutieren oder den Benutzer kennenzulernen. Man sollte die Netzwerkverbindung zwischen zwei Clients aufbauen.

Ziel: Schaffung offener Verbindung, die zum Nachrichtenaustausch zwischen den Benutzern dient, beispielsweise durch einen Chat.

Exit / Fail Kriterium: Nur wenn eine ständig offenen Verbindung zwischen den Client und einem anderen Client über den Server besteht und diese miteinander kommunizieren können, so gilt die Prüfung als bestanden. Ausnahme ist, wenn der Client nicht kommuniziert wie gewünscht mit dem Server und es findet kein Nachrichtenaustausch statt.

Alternative Fallback: Eine mögliche Alternative wäre eine Handynummer im Profil des Benutzers zu hinterlegen und somit die persönliche Kommunikation zwischen den Benutzern erlauben. Dies sollte dann als Pflichtfeldeingabe im Datenmodell benutzt werden.

Status: Es wurde eine Chat Anwendung mithilfe von NodeJS und Socket.io realisiert. Später sollte die Funktion in die Android-Anwendung implementiert werden.

5.3. Standort

Risiko: Die Benutzer verlieren das Interesse an dem System, wenn schon zu Beginn viele Abfragen gestellt werden, beispielsweise über den aktuellen Standort.

Ziel: Um die Benutzer nicht unnötig mit Standortfrage zu belästigen, sollte eine die Lokalisierung so einfach wie möglich gehalten werden. Mittels GPS sollte der Standort des Benutzers ermittelt werden.

Exit / Fail Kriterium: Nur wenn die GPS Position mittels der Android-Anwendung bestimmt werden kann, so gilt die Prüfung als bestanden, ansonsten greift man auf die Alternative zurück.

Alternative Fallback: Die Alternative sieht vor eine manuelle Eingabe des Standorts.

Status: Die GPS Position kann mithilfe des Android Gerätes ermittelt werden und steht für weitere Entwicklung zur Verfügung.

5.4. Wetter

Risiko: Es ist den Spielern auf den Basketballplatz nicht bewusst, dass es im Laufe des Tages regnen könnte.

Ziel: Wetterinformationen sollen die Benutzer vor Regen warnen.

Exit / Fail Kriterium: Wenn die Wetterinformationen bei den Benutzern in seiner Umgebung (manuelle Standorteingabe) korrekt als Warnungen ausgegeben werden können. Ausnahme ist eventuell die Anbindung der Schnittstelle bei den Drittanbietern.

Alternative Fallback: Es gibt mehrere Dienstleister am Markt die, die Wetterinformationen und die passenden APIs zur Verfügung stellen. Sollten die Wetterwarnungen nicht funktionieren kann man einfach die Wettervorhersagen für die nächsten zwei Wochen zur Verfügung stellen ohne die Warnungen.

Status: Das Vorgehen hat gezeigt, das die Daten durch einen HTTP-Request durchgeführt werden können. Hier wird auf den „openweathermap[12]“ Wetterdienst zugegriffen und nicht wie zuvor beschrieben auf „wetter.com“, da man sonst zwei Wochen warten müsste,

bis ein Antrag für die Schnittstelle bewilligt wird. Außerdem wird hier über XML und nicht über JSON kommuniziert. Die Warnungen im Regenfall konnten leider noch nicht umgesetzt werden, deshalb gilt der PoC als fehlgeschlagen. Als Alternative wurde die Ausgabe von den nächsten zwei Wochen zur Verfügung gestellt.

5.5. Matchmaking

Risiko: Ohne Matchmaking Algorithmus wird das System für den Benutzer uninteressant und bietet daher kaum einen Mehrwert in der Domäne.

Ziel: Implementierung der Bewertungsgrundlagen anhand der Spielergebnisse schaffen, dann mit Pseudodaten anreichern und testen. Matchmaking verknüpfen mit den Wetterinformationen und Basketballplätzen.

Exit / Fail Kriterium: Nur wenn der Algorithmus einwandfrei funktioniert und die Funktionalitäten der Benutzer bestmöglich erfüllt, so gilt es als erfolgreich. Ausnahme ist, wenn der Algorithmus für jeden Benutzer, der die gleichen Ergebnisse liefert oder der Algorithmus bezieht alle Ergebnisse der Benutzer mit ein, auch diejenigen die nicht in der Umgebung leben.

Alternative Fallback: Es gibt kaum eine Alternative dafür, außer man macht es in einer kleinen Domäne, wo jeder Jeden kennt.

Status: Das Matchmaking wird zu diesen Zeitpunkt nur als Pseudocode entwickelt, da hier das Testen der Netzwerkstruktur Vorrang hat und muss zuerst als erfolgreich abgeschlossen werden um später die Implementierung im System zu gestatten.

6. WBA-Modellierungen

6.1. Kommunikationsmodell

Die Anwendung wirkt als verteiltes System mit verschiedenen Teilkomponenten. Im Folgenden wird der Ablauf der Kommunikation genauer erläutert, ohne auf die technischen Details einzugehen.

6.1.1. Deskriptives Modell

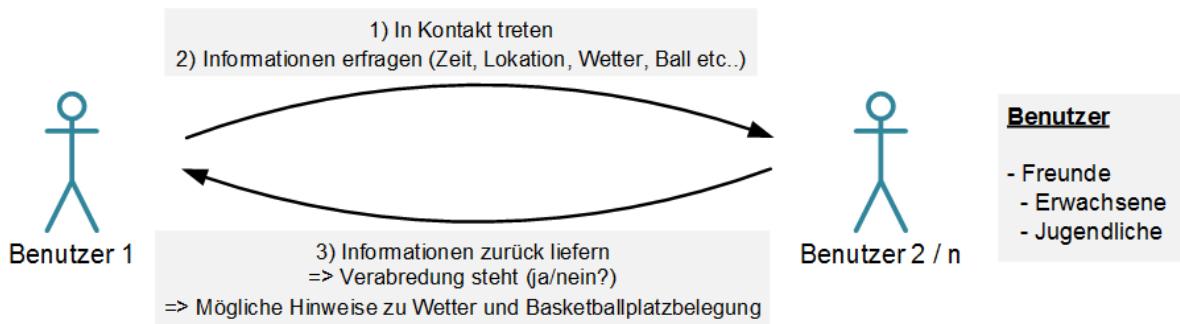


Abbildung 3: deskriptives Kommunikationsmodell

Die Abbildung 3 zeigt die momentane Situation zwischen den Benutzern. Derzeit verläuft die Kommunikation zum größten Teil über die Sprachebene, indem der Benutzer 1 mit dem Benutzer 2 in Kontakt tritt. Sie tauschen verschiedene Informationen untereinander aus. Beispielsweise wird erfragt:

- Wann und wo wird sich getroffen?
- Eventuell wird das momentane Wetter geprüft, regnet es?
- Wer bringt den Basketball zu den verabredeten Platz mit?

Des Weiteren bleibt jedoch die Problematik bestehen, ob ein Basketballplatz belegt ist oder nicht. Die Benutzer können verschiedene Personen sein, sind jedoch in den meisten Fällen nur Freunde aus der Umgebung die sich gut kennen.

6.1.2. Präskriptives Modell

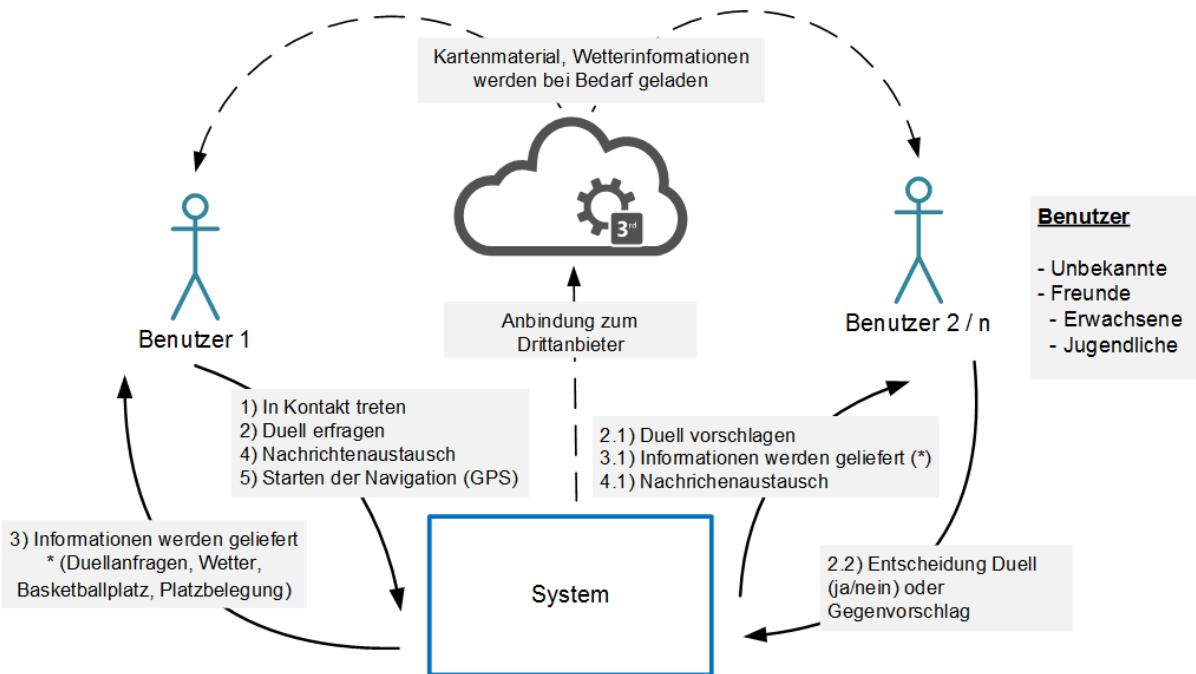


Abbildung 4: präskriptives Kommunikationsmodell

In der Zukunft sollte möglichst der größte Teil der Kommunikation über das System erfolgen (Abb. 4). Ein Benutzer kann in Kontakt mit dem System treten und nach einen Match anfragen. Daraufhin werden die möglichen Informationen, wie beispielsweise andere Matchanfragen, Wetterinformationen, Basketballplatz geliefert. Anhand der Informationen kann der Benutzer 1 mit anderen Benutzer(n), die sich vorher im System registriert haben in Kontakt treten. Hierbei findet ein Nachrichtenaustausch statt.

Sollte sich ein Match ergeben, so werden den Benutzer 2 möglichen Matches vorgeschlagen und die passenden Informationen geliefert, wie beispielsweise Datum, Uhrzeit, Ort, etc. Der Benutzer kann anhand dieser Informationen selbst entscheiden, ob man mit dem Vorschlag einverstanden ist oder einen Gegenvorschlag unterbreiten möchte.

Hierbei kann die Kommunikation zwischen den verschiedenen Benutzern in verschiedene Richtungen laufen. Das präskriptive Kommunikationsmodell sollte nur einen Fallbeispiel zeigen, wie so was in der Anwendung ablaufen könnte. Die Benutzer können verschiedene Personen sein, die man eventuell aus der Umgebung nicht kennt. Dadurch könnten sich neue Freundschaften und Gemeinschaften bilden.

Drittanbieter liefern die Wetterdaten und das Kartenmaterial, um Wettervorhersagen und Navigationsfunktionalitäten zur Verfügung zu stellen.

6.1.3. Architekturdiagramm

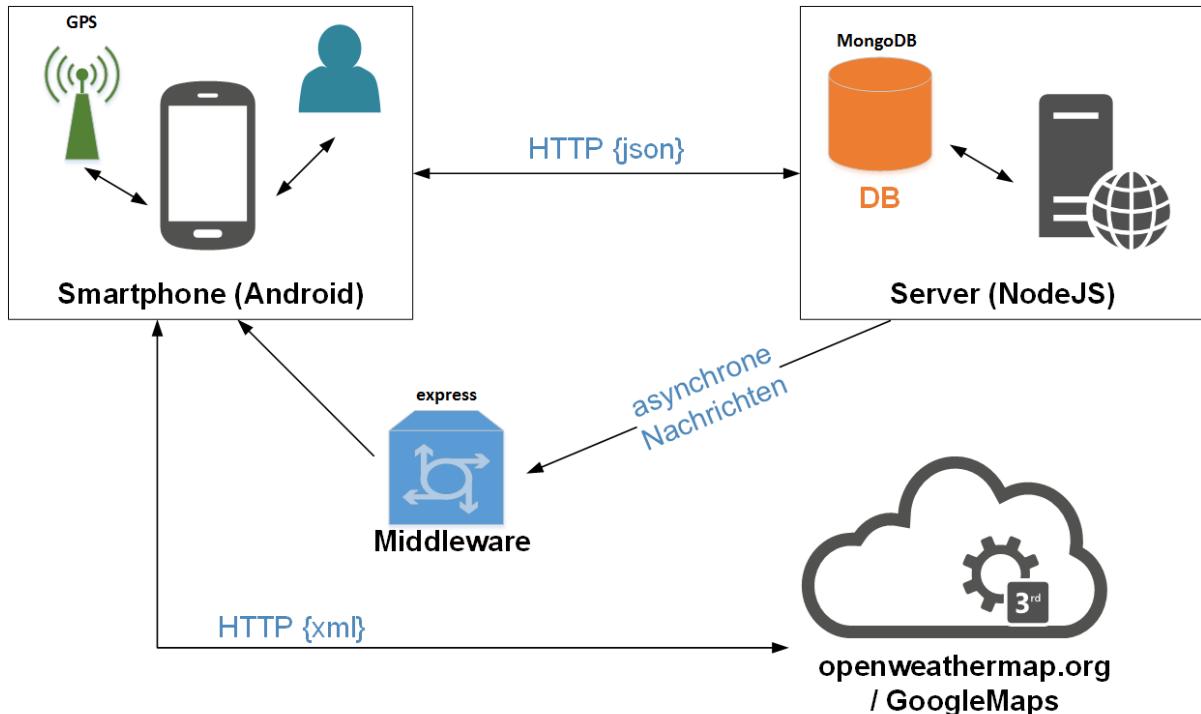


Abbildung 5: Systemarchitektur

6.2. Systemarchitektur

Die Systemarchitektur (Abb. 5) sollte über einen Server mit angebundener Datenbank kommunizieren. Der Client besteht aus einer Präsentations- und Anwendungslogik. Bei dem Endgerät spielt die GPS Funktion eine wichtige Rolle, um beispielsweise die Wetterinformationen für bestimmte Ortschaft abrufen zu können oder sich zum Basketballplatz navigieren zu lassen. Hierbei wird auf die meteorologischen Wetterinformationen von einem Drittanbieter „openweathermap.org[12]“ zugegriffen und über eine XML Struktur abgerufen. Außerdem sollte das System in der Lage sein, die Textnachrichten zwischen den Benutzern in einer asynchronen Verbindung auszutauschen, dies wurde bereits in den PoCs realisiert.

Obwohl es eine Menge voneinander unabhängiger Computer sind, so sollte es für den Benutzer als eine einzelnes zusammenwirkendes System erscheinen. Im weiteren Punkten werden die Abwägungen für die Systeme erläutert.

Teil IV.
MEILENSTEIN 4

6.3. Server

Auf dem Server sollen verschiedene Aufgaben durchgeführt werden, unter anderem das Matchmaking. Die Abwägung des Servers fiel auf die Implementierung von NodeJS[11]. NodeJS bietet gegenüber PHP[13] schnellere und performante Ausführung, weniger Arbeitsspeichernutzung und ermöglicht zudem das Verwenden von gemeinsamen Code auf der Client und Serverseite. Dies ist eine Plattform, basierend auf JavaScript-Laufzeitbibliotheken, welches einen schnellen Bau von Netzwerkanwendungen erlaubt. Ein weiterer Aspekt ist die asynchrone Kommunikation bei Ein- und Ausgabe der Daten. Im aktuellen Projekt werden die Daten über die REST-Grundprinzipien[28] versendet und empfangen. Hier kann die Anwendung POST und GET-Anfragen an die URI-Ressourcen des Servers versenden und empfangen. So werden beispielsweise bei der Registrierung die Daten Repräsentationsformat JSON abgelegt und anschließend wird die Client-Anwendung darüber informiert ob der Vorgang erfolgreich war oder nicht.

6.3.1. Datensicherheit und Datenschutz

Leider findet die derzeitige Kommunikation des Prototypen über das HTTP-Protokoll im Klartext statt, hier werden die Daten unverschlüsselt übertragen. In Zukunft sollte man jedoch die verschlüsselte Form der Kommunikation durch HTTPS anstreben.

GoogleMaps ist ein Dienst des US-amerikanischen Unternehmens Google Inc und unterliegt nicht den deutschen Datenschutzbestimmungen. So wird beispielsweise die Ermittlung des eigenen Standorts mit der Funktion „Mein Standort“ durch die GoogleMaps API[5] abgefragt, wodurch die Positionsangabe durch Längen- und Breitengrad des GPS bestimmt wird. Leider werden die Informationen über die empfangenen Mobilfunknetze miteinbezogen.

6.3.2. Middleware

Die Middleware stellte eine Komponente dar, die neben der verteilten Anwendung auf verschiedenen Systemen laufen kann. Bei der Middleware fällt die Wahl auf das NodeJS Modul „express“. Dies ist ein kleines und flexibles NodeJS-Webframework, dessen Funktionsumfang sich mit ergänzenden Modulen je nach Bedarf erweitern lässt.

6.3.3. Vorläufiges Matchmaking (Pseudo-code)

```

1 function matchmaking () {
2     if (aktuelleGpsPosition < 10 km) &&
3         (benutzerAlter > -3 || benutzerAlter < 3) {
4             return Benutzer (spielerNamen());
5         } else {
6             return 'keine Spieler gefunden!';
7         }
8 }
```

Listing 1: Vorläufiges Matchmaking (Pseudo-code)

Unter dem Menüpunkt „Herausfordern“ könnte ein Matchmaking-Algorithmus auf dem Server im Hintergrund ausgeführt werden, sobald der Benutzer neue Spieler herausfordert. Es werden als erstes die Spieler angezeigt, die sich in der umliegenden Umgebung befinden, beispielsweise im Radius von 10 km. Anschließend wird das Alter der Benutzer berechnet und geschaut, ob die Benutzer 3 Jahre älter oder junger sind. Danach werden die Spielernamen als Vorschlag ausgegeben und können entsprechend ausgewählt werden. Natürlich sollte man auch weiterhin seine Gegenspieler selbst bestimmen können und beispielsweise in einer Suchliste auswählen.

6.4. Datenmodell

Für die persistente Datenhaltung wird die Datenbank MongoDB[8]“ verwendet. Diese bietet gegenüber den relationalen Datenbank (bspw. MySQL[9]) ein flexibles Datenmodell und erlaubt die Daten in JSON-Format abzuspeichern. Im aktuellen Projekt verwaltet die Datenbank vier Collections und ist über die jeweiligen ObjectID's verknüpft.

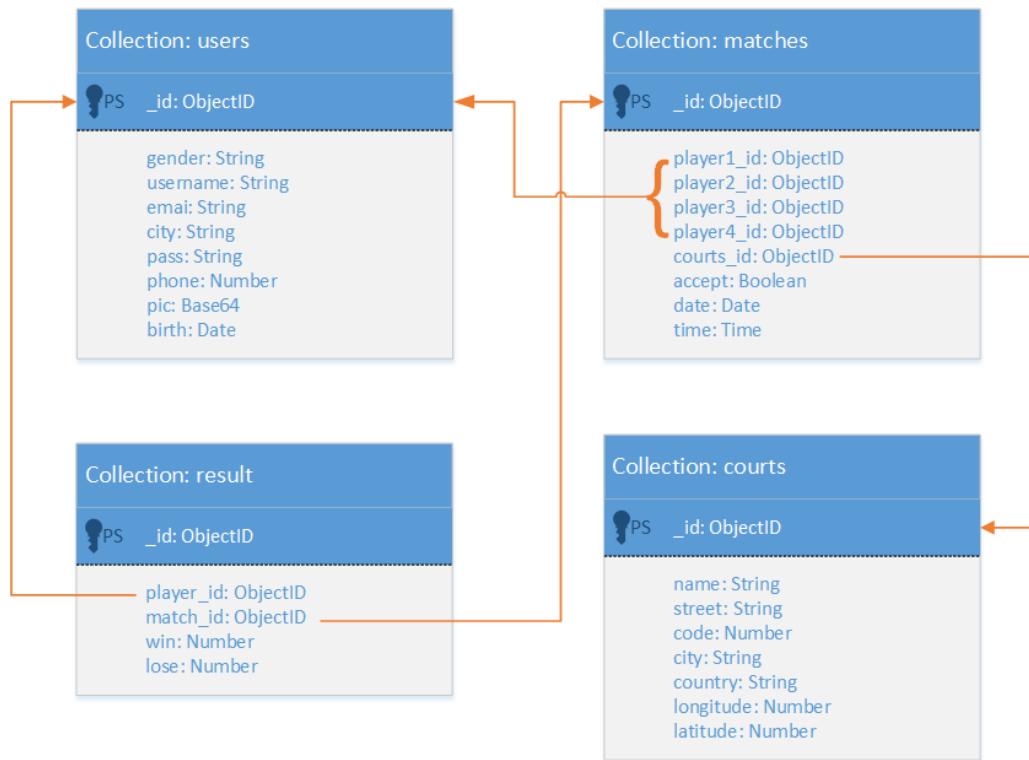


Abbildung 6: MongoDB Datenmodell

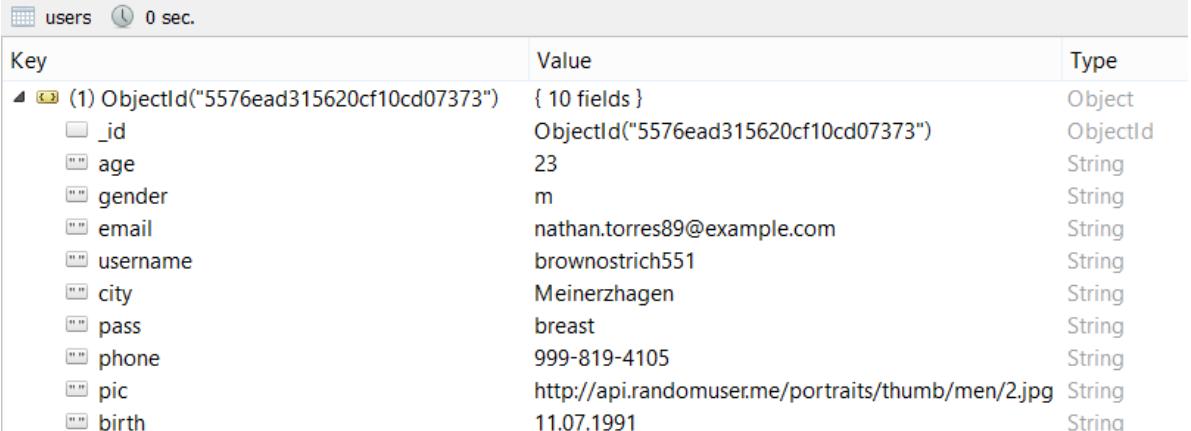
Vorerst wurden die Collections mit Beispieldaten (Mock-Up) gefüllt, damit später verschiedene Berechnungen durchgeführt werden können. Im folgenden werden die einzelnen URI's mit den jeweiligen Verben beschrieben, dabei wird der Payload in Form eines JSON-Objekts übertragen und ist hauptsächlich identisch mit der Datenstrukturen.

URI	Verben	Beschreibung
/users	GET	Benutzer abrufen
/users	POST	Benutzer erstellen
/users/:id	GET	Informationen zu einem Benutzer anzeigen
/users/:id	PUT	Benutzer bearbeiten
/users/:id	DELETE	Benutzer löschen
/users/playerresult	GET	Gesamtergebnisse zu allen Benutzern
/users/playserresult/:id	GET	Ergebnisse zu einem Benutzer
/users/login	POST	Benutzer Anmeldung
/users/chat	GET	Benutzer Chat (Textnachrichten)
/courts	GET	Basketballplätze abrufen
/courts	POST	Basketballplätze erstellen
/courts/:id	GET	Einen Basketballplatz abrufen
/courts/:id	PUT	Einen Basketballplatz bearbeiten
/courts/:id	DELETE	Einen Basketballplatz löschen
/matches	GET	Matches abrufen
/matches	POST	Match erstellen
/matches/:id	GET	Einen Match abrufen
/matches/:id	PUT	Einen Match bearbeiten
/matches/:id	DELETE	Einen Match löschen

Tabelle 6: Ressourcen mit HTTP-Verben

Users

Die Collection „users“ liefert die wichtigen Informationen zu den Benutzern.



Key	Value	Type
« 1 ObjectID("5576ead315620cf10cd07373")	{ 10 fields }	Object
_id	ObjectID("5576ead315620cf10cd07373")	ObjectID
age	23	String
gender	m	String
email	nathan.torres89@example.com	String
username	brownstrich551	String
city	Meinerzhagen	String
pass	breast	String
phone	999-819-4105	String
pic	http://api.randomuser.me/portraits/thumb/men/2.jpg	String
birth	11.07.1991	String

Abbildung 7: Datenstruktur der Benutzer

Courts

Basketballplätze, beinhaltetet Informationen über die Plätze und deren Standort.

Key	Value	Type
« ③ (1) ObjectId("5576f84815620cf10cd0737c")	{ 8 fields }	Object
└ _id	ObjectId("5576f84815620cf10cd0737c")	ObjectId
└ name	Lindengymnasium	String
└ street	Moltkestraße 50	String
└ code	51643	String
└ city	Gummerbach	String
└ country	Deutschland	String
└ longitude	7.570.996	String
└ latitude	51.032.045	String

Abbildung 8: Datenstruktur der Basketballplätze

Matches

Spielergebnisse, beinhaltet die gespielten Spiele mit den dazugehörigen Benutzern, Basketballplätzen sowie Datum und Uhrzeit.

Key	Value	Type
« ③ (1) ObjectId("5576fae315620cf10cd0738b")	{ 8 fields }	Object
└ _id	ObjectId("5576fae315620cf10cd0738b")	ObjectId
└ player1_id	5576ead315620cf10cd07373	String
└ player2_id	5576ead315620cf10cd07379	String
└ player3_id	5576ead315620cf10cd0731f	String
└ player4_id		String
└ courts_id	5576f84815620cf10cd0737c	String
└ date	09.06.2015	String
└ time	09:58	String

Abbildung 9: Datenstruktur der Matches

Result

Ist das Ergebnis der einzelnen Benutzer, dort stehen die Siege und die Niederlagen der einzelnen Matches.

Key	Value	Type
« ③ (1) ObjectId("557739f115620cf10cd073ee")	{ 5 fields }	Object
└ _id	ObjectId("557739f115620cf10cd073ee")	ObjectId
└ player_id	5576ead315620cf10cd07373	String
└ match_id	5576fae315620cf10cd0738b	String
└ win	2	String
└ lose	1	String

Abbildung 10: Datenstruktur der Ergebnisse

6.5. Endgeräte (Client)

Hierbei stellt sich die Frage zwischen den verschiedenen Betriebssystemen. Ein der beliebtesten auf dem Markt ist das Android OS von Google. Es hat zurzeit den größten Marktanteil gegenüber den anderen Systemen und wird deshalb für das Projekt priorisiert.

6.5.1. Android Anwendung

Man sollte so flexibel wie möglich sein und relative Größen definieren, um die verschiedenen Geräte mit verschiedenen Auflösungen abzudecken. Die Größen werden in dp (dip per inch) definiert, dies ist wichtig, da man die Elemente mit dem Finger anklickt und nicht wie etwa auf einem Computer mit der Maus. Finger haben einen bestimmten Durchmesser, deshalb gibt es seit neuesten Richtlinien[1] für die Metrics und Grids aus dem Hause Google.

6.5.2. Drittanbieter

Die Wetterinformationen werden von einem Drittanbieter „openweathermap.org“ abgerufen. Die Anfragen an die Schnittstelle erfolgen über ein einfaches REST Format. Als Ausgabeformat wird JSON mit geringen Overhead und damit geringen Übertragungszeiten unterstützt. Die Systemkommunikation zwischen den Drittanbieter soll von dem Server ausgehen, dass die Anwendungslogik zum Abrufen der Wetterinformationen auf der Benutzerseite stattfindet.

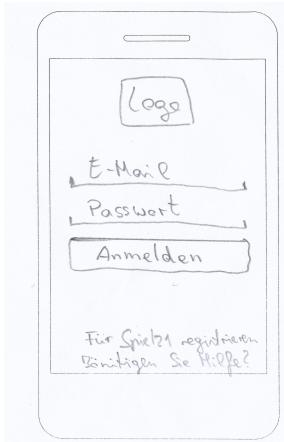
Das Kartenmaterial kommt von einem Drittanbieter GoogleMaps API[5]. Hier kann man für Endgeräte die Standordienste und Karten hinzufügen. Dies basiert auf einer JavaScript API und ist von Google gut auf der Webseite dokumentiert. Durch die GPS-Funktion wird der Standort ermittelt und auf Wunsch können Routen berechnet werden.

7. Prototypen UI

Nach dem die Anforderungen für das System erstellt wurden, folgen nun die vorläufigen Prototypen für das System.

7.1. Papierbasierte Prototypen

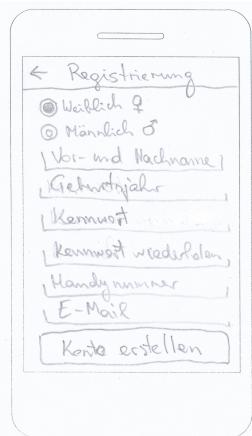
Bei einem papierbasierten Prototyp wird mit einfachen Mitteln nur grob eine Skizze gezeichnet und dient bei der Entwicklung neuer Produktkonzepte. Einer der wichtigen Vorteile ist, dass es keine Grenzen bei der Kreativität durch Technik oder sonstiges gesetzt werden. Hier wird der Fokus auf die vereinfachte Darstellung und leichte Zugänglichkeit für den Benutzer gelegt, so ist beispielsweise die beste Navigation die, die man nicht wahrnimmt. Die einzelnen Anforderungen, sowie die Normen werden hier in Abhängigkeit gezogen und in Iteration umgesetzt.



Start der Anwendung

Oben sollte das Logo des Spiels erscheinen, um speziell die Domäne zu erkennen. Zwei Optionen sollten auf dem Startbildschirm zu Verfügung gestellt werden. Die Anmeldung (Use Case 1) mit E-Mail und Passwort, dafür gibt es einen Button. Nach erfolgreichen Einloggen sollte der Benutzer zum Menü weiter geleitet werden. Ein anderer Punkt ist die Registrierung (Use Case 2), dafür gibt es einen Text zum Anklicken.

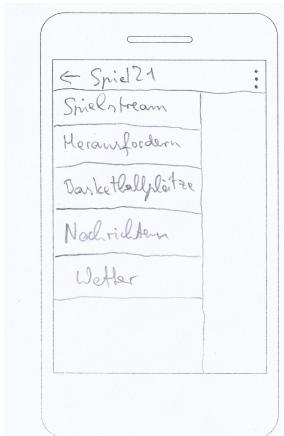
Abbildung 11: UC1



Registrierung des Benutzers

Bei der Registrierung werden die wichtigen Daten von den Benutzer abgefragt. Anschließend nach dem Ausfüllen und Validation aller Felder die Daten persistent in der Datenbank gespeichert. Um das Konto zu erstellen gibt es einen entsprechenden Button dafür. Nach erfolgreicher Registrierung sollte der Benutzer eingeloggt werden (Use Case 1) und zum Menü (Use Case 3) weiter geleitet werden.

Abbildung 12: UC2



Navigation und Menü

Nach dem erfolgreichen Einloggen sollte der Benutzer das Navigationsmenü zu sehen bekommen und darüber die wichtigen Menüpunkte (Use Case 4, 5, 7, 8, 9 und 12) ansteuern können.

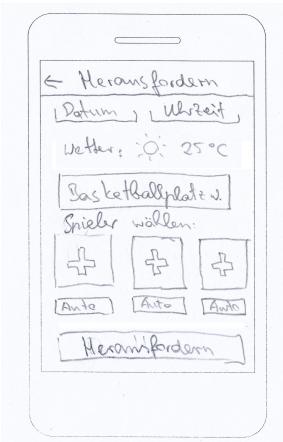
Abbildung 13: UC3



Spielstream

Im dem Menüpunkt Spielstreams hat der Benutzer die Ansicht mit den gespielten Spielen. Hier sind hierarchisch die gespielten Matches zu sehen zwischen den einzelnen Benutzer mit dem entsprechenden Spielergebnis und den Basketballplatz. Des Weiteren sollte oben eine ActionBar zu sehen sein um somit die Navigation zu erleichtern und um weitere Einstellungen (Use Case 10) vornehmen zu können

Abbildung 14: UC4



Herausforderungen

Die Überschrift zeigt immer wo sich zur Zeit der Benutzer befindet. Hier kann der Benutzer andere Spieler zu einem Match herausfordern, dabei kann er das Datum und die Uhrzeit bestimmen, daraufhin wird automatisch das passende Wetter für den spielenden Tag angezeigt. Weiterhin wählt er einen Basketballplatz (Use Case 7) und weitere Spieler (Use Case 6) aus. Das System kann die Spieler automatisch vorschlagen durch das Matchmaking.

Abbildung 15: UC5



Gegenspieler wählen

Im Menü Herausfordern kann der Benutzer manuell seinen Gegenspieler wählen. Durch das Plus-Symbol wird in einer Itemliste vorgeschlagen mit den registrierten Spielern im System, dort kann er seinen Gegenspieler auswählen und kehrt anschließend wieder in das Herausforderungen Menü (Use Case 5) zurück.

Abbildung 16: UC6



Basketballplatz auswählen

Im Menü Herausforderungen (Use Case 5) kann man auf einer Karte einen Basketballplatz auswählen, seinen eigenen Standort bestimmen, sich zu den Basketballplatz navigieren lassen und einen neuen hinzufügen. So kann hier beispielsweise ein Basketballplatz gewählt werden und man kehrt anschließend wieder in das Herausforderungen Menü (Use Case 5) zurück. Die Kartenansicht sollte über das Menü (Use Case 3) direkt ansteuerbar sein.

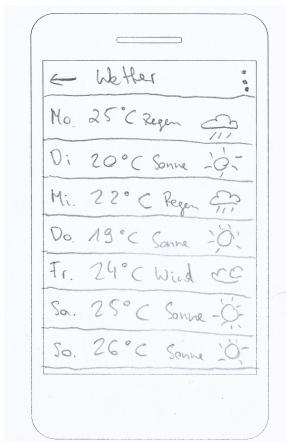
Abbildung 17: UC7



Nachrichten

Im Menü Nachrichten können die Benutzer verschiedenen Textinformationen austauschen und andere Benutzer kennen lernen. Hier kann beispielsweise ausgemacht werden, wer den Basketball zum Platz bringen soll.

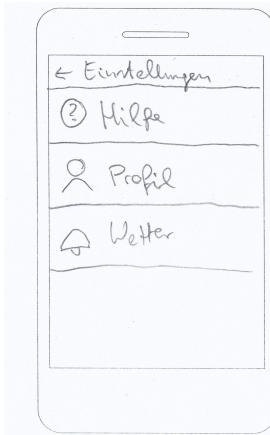
Abbildung 18: UC8



Wetter

Einer wohl der wichtigsten Nutzungsprobleme bei dem Spiel ist wohl die Wetterlage. Im Menü Wetter wird eine Vorhersage des Wetters bekannt gegeben und der Benutzer kann sich über die Wetterlage informieren und kann somit besser mit anderen Benutzern verabreden.

Abbildung 19: UC9



Einstellungen

Die Option Einstellungen wird nach den erfolgreichen einloggen sichtbar und beinhaltet Auskunftsinformationen über das Spiel unter dem Punkt „Hilfe“, dort kann der Benutzer sich beispielsweise über die Regeln des Spiels informieren. Unten den Punkt „Profil“, kann der Benutzer seine registrierten Informationen im System einsehen. Letzter Punkt ist das „Wetter“, dort kann der Benutzer den Standort manuell bestimmen oder durch das GPS automatisch bestimmen lassen.

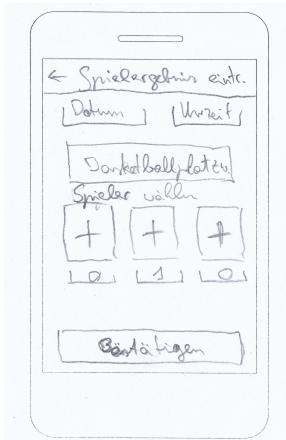
Abbildung 20: UC10



Popup Benachrichtigung

Die Popup Benachrichtigung taucht dann auf, wenn ein Benutzer einen anderen zum Spiel herausfordert (Use Case 5). Hier werden die Grundlegende Informationen (Name, Datum, Uhrzeit, Wetter) für den Benutzer gezeigt. Hier kann der Benutzer selbst die Entscheidung treffen, ob er die Herausforderung annimmt oder ablehnt.

Abbildung 21: UC11



Spielergebnis eintragen

Sollte es zu einer Herausforderung (Use Case 5) kommen und diese wurde angenommen (Use Case 11). So kann der Benutzer nach dem gespielten Spiel sein Spielergebnis eintragen, der Gegenspieler muss das gleiche tun. Das System verifiziert dann die Ergebnisse und zeigt diese später in den Spielstream (Use Case 3) an.

Abbildung 22: UC12

Teil V.
MEILENSTEIN 5

7.2. Evaluationsprozess

Durch die Iteration und die „Think Aloud“ Technik der Probanden wurde ersichtlich, dass das Design überarbeitet werden sollte. Nur die Use Cases 3 und 9 wurden als selbsterklärend und verständlich empfunden. Bei der Methodik wurden die Prototypen einzeln angeschaut, ob die gewünschte Aktion mit der Funktion eingeht. Die Mängel wurden zur Kenntnis genommen und im folgenden verschiedene Gestaltungslösungen erarbeitet.

Use Case 1:

Nachdem der Proband die Anwendung gestartet hat, war im nicht ersichtlich, dass man auf den Schriftzug „Für Spiel 21 registrieren“ klicken kann und somit ein Benutzerkonto anlegen kann. Bemängelt wurde der Schriftzug „Benötigen Sie Hilfe“, hier erwartet der Proband noch keine Hilfe, da die Maske selbsterklärend ist.

Deshalb wird der erste Schriftzug als Button „Registrieren“ abgewandelt und die Hilfestellung wird in die Hauptnavigation ausgelagert.

Use Case 2:

Auf der Registrierungsmaske wurde dem Probanden nicht ersichtlich welche Felder man ausfüllen muss und welche sind optional. Des Weiteren ist hier aufgefallen, dass der Wohnort nicht abgefragt wird, dabei ist dieser wichtig für spätere Lokalisierung der Benutzer. Außerdem möchte man seinen Vor- und Nachnamen ungern preisgeben.

Für das Projekt wird das Feld „Vor- und Nachname“ in „Benutzername“ umbenannt, Handynummer nur als optionales Feld hinzugezogen und ein extra Feld für den Wohnort wird abgefragt. Eventuell kann man schon hier, oben rechts in der Ecke eine Hilfestellung für den Benutzer anbieten.

Use Case 4:

Auf dem Prototypen erkannt der Proband nicht, dass es hierbei um gespielte Spiele handelt, da der Schriftzug „Spielstream“ irreführend ist.

Deshalb wird der Schriftzug in „Spielergebnisse“ umbenannt. Außerdem ist aufgefallen, dass hier nur ein Spieler gegen einen anderen angezeigt wird (Duell). Es kann jedoch maximal mit bis zu vier Spielern (Match) das Basketballspiel gespielt werden, deshalb sollte hier der Prototyp noch mal überarbeitet werden. Als Vorschlag wurde Ranking erwähnt mit den folgenden Fragen, wo oder wer hat die meisten Spiele gespielt?

Im Hauptmenü (Use Case 3) soll neben den Schriftzug ein kleiner Schriftzug mit der Anzahl der neuen Herausforderungen erscheinen.



Abbildung 23: UC4

Use Case 5:

In dem Menü „Herausfordern“ erkannte der Proband nicht, welche Spieler man bereits herausgefordert hat und welche nicht. Außerdem wurde der Button „Auto“ bemängelt, dies sollte ursprünglich den Spieler anhand des Matchmaking einen passenden Spieler aus der Umgebung vorschlagen.

Für das Projekt wurde abgewogen, vorher eine Itemliste zu erstellen mit den herausgeforderten Spieler und deren Antwortstatus. Somit könnte die Popup Benachrichtigung (Use Case 11) passend in das Menü ausgelagert werden. Außerdem bekommt das Menü (Use Case 3) ein klein Nachrichtenzahl im Menü, so kann der Benutzer sehen, ob neue Matches bevorstehen.

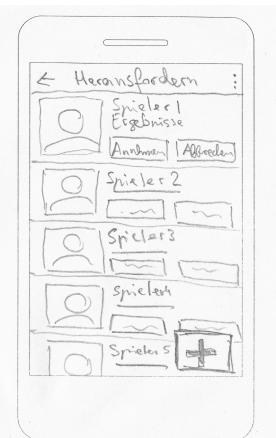


Abbildung 24: UC5

Use Case 6-7

Beim „Gegner wählen“ (Use Case 6) wird automatisch im Hintergrund anhand des Matchmaking die passenden Spieler vorgeschlagen. Sollten keine passenden Spieler gefunden werden, so wird der Radius des Wohnorts erweitert. Außerdem wird eine Suchfunktion eingebaut, so kann der Benutzer anhand des Benutzernamens den passenden Spieler finden und herausfordern.

Beim „Basketballplatz wählen“ (Use Case 7) wurde dem Probanden nicht ersichtlich, wie man zurück zu der vorherigen Maske kommt. Deshalb wird hier in die Sprechblase, neben den Informationen zu den Basketballplatz ein Button mit dem Schriftzug „hinzufügen“ eingebaut.

Use Case 8:

Bei dem Chat wurde bemängelt, dass es zu viele Textnachrichten gibt, je nachdem wie viele Spieler sich in dem Chat aufhalten werden.

Deshalb wird noch weitere Überlegungen abgewogen, ob man diese Funktionen auf die einzelnen Benutzer personalisieren sollte oder komplett wegfallen lässt. Da der Benutzer eine Handynummer ins Profil eintragen kann (alternatives Fallback 5.2) und ist somit für andere Benutzer erreichbar.

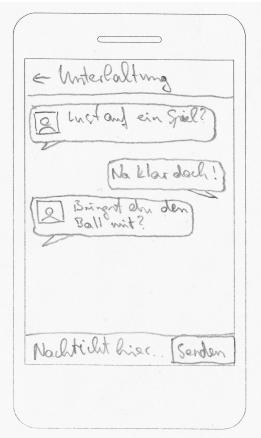


Abbildung 25: UC8.1 Abbildung 26: UC8.2

Use Case 10:

Hier wurde den Probanden nicht ersichtlich warum man dort auf die Wetttereinstellungen zugreifen kann und außerdem konnte man sich nirgendwo aus der Anwendung ausloggen. Somit wird der Menüpunkt „Wetter“ gestrichen und ist nur unter dem Use Case 9 sichtbar. Außerdem wird ein neuer Menüpunkt hinzugefügt „Logout“.

Use Case 12:

Nach dem der Spieler das Spiel 21 gespielt hat, war es den Probanden nicht klar, wie man das Spielergebnis eintragen soll.

Hier wurde der Menüpunkt in der Navigation bei dem Prototypen vergessen. Durch die Iteration hat sich jedoch gezeigt, dass man diesen jedoch bei den „Spielergebnissen“ platzieren sollte. Durch das Plus-Symbol sollte es den Benutzer ersichtlich sein, dass man dort ein Spielergebnis platzieren kann.

7.3. Weitere Erkenntnisse

Anbei werden weitere Erkenntnisse in Bezug auf DIN-Norm 9241, Teil 210 beschrieben. So werden beispielsweise die Selbstbeschreibungsfähigkeit, Lernförderlichkeit und Fehlertoleranz betrachtet.

Selbstbeschreibungsfähigkeit Die Anwendung verwendet einheitliche Steuerelemente, so wird beispielsweise die Zustandsänderung im Menü der Navigation auf der ActionBar angezeigt. Dadurch weiß der Benutzer immer, wo er sich gerade befindet.

Durch die einfache Menüstruktur entfällt die Einführung beim Prototypen. Benötigt der Benutzer jedoch weitere Hilfe, wie beispielsweise die Regeln zum Spiel, so kann er diese unter Hilfe abrufen. Außerdem wurde eine weitere Erkenntnis bei der Evaluation sichtbar. Die Benutzer wollen Schalflächen (Buttons) haben mit selbstbeschreibenden Titeln. Dies stellt sich als wichtig heraus, um einen störungsfreien Ablauf zu gewährleisten.

Lernförderlichkeit Für die neuen Benutzer können erklärende Texte oder Tutorials zur Verfügung gestellt werden. Außerdem erlaubt das Menü einfache Navigation durch die einzelnen Menüeinträge, so können die Benutzer durch ausprobieren die Anwendungsfunktionen schnell erlernen.

Fehlertoleranz Hier sollte das System in der Lage sein die fehlerhaften Eingaben automatisch zu erkennen und diese den Benutzer entsprechend mitteilen. Beispielsweise bei der Registrierung sollten keine leeren Eingabefelder akzeptiert werden. Hier werden die Passwörter doppelt abgefragt um die Einprägsamkeit zu steigern und die Fehler durch Falscheingabe zu vermeiden. Im weiteren betrifft es auch weitere Eingabefelder, wie beispielsweise E-Mail Adresse.

Ein weiteres Beispiel ist die Herausforderung der Benutzer zu einem Match. Hier muss gewährleistet werden, dass das Datum, Uhrzeit, Basketballplatz sowie mindestens ein Spieler ausgewählt werden, bevor eine Herausforderung versendet werden kann.

8. Narratives Konzept für filmische Präsentation

Es wird ein Kurzfilm anhand von einiger Folien und Screenshots erzählt, diese werden entsprechend animiert und betitelt.

Am Anfang wird zu der Einführung mit thematisch bezogener Einleitung als Kurzform textuell dargestellt. So wird beispielsweise Name des Projektes, Campusort und die Anwendung an sich beschrieben.

Als Aufmacher sollte der folgende Titel verwendet werden:

**Du willst raus in die Sonne, Basketballspiel 21 spielen?
Die Anwendung „Spiel 21“ soll dir helfen, Spieler und Basketballplätze
in deiner Umgebung zu finden.**

Somit werden alle wichtigen Merkmale der Anwendung auf den Punkt gebracht und der Benutzer bekommt eine gute Vorstellung davon.

Für das Fachpublikum wird später die Systemarchitektur gezeigt und die Client/Server Verbindung mit den jeweiligen Funktionen gezeigt. Dabei wird auf die Datenübertragung mit GET/POST eingegangen und auf das Austauschformat JSON/XML. Praktischerweise werden die einzelnen Menüpunkte angesprochen und daraus die einzelnen Szenen generiert.

Szene: Herausfordern Unter dem Menüpunkt „Herausfordern“ kann man den passenden Gegner (Freunde oder neue Leute) in seiner Umgebung raus suchen, die Spielstärke prüfen und anschließend zum Match herausfordern.

Szene: Karte Mithilfe der Kartenfunktion sieht man, wo in der Umgebung Basketballplätze gibt. Außerdem kann man eine direkte Route zum Basketballplatz in seiner Umgebung erhalten.

Szene: Spielergebnisse Die Ergebnisse können nach dem gespielten Spiel unter dem Menüpunkt „Spielergebnisse“ eingetragen werden. Daraufhin wird im Vergleich gezeigt, wie gut man ist gegenüber den anderen Spielern. Außerdem können die Statistiken (Siege und Niederlagen) im Profil eingesehen werden.

Szene: Wetter Es wird auf die Benutzerproblematik hingewiesen, beim Schlechtwetter, wie beispielsweise Regen kann kein Spiel gespielt werden. Unter dem Menüpunkt kann man entsprechend die Wettervorhersage für die nächsten 1 bzw. 2 Wochen abgerufen.

Die wichtigsten Funktionen des Prototypen ist der Menüpunkt „Herausfordern“, hier wird beispielsweise gezeigt wie man andere Benutzer zu einem Spiel einlädt mit den passenden Basketballplatz. Hierbei muss der Benutzer Datum und Uhrzeit eingeben, dabei wird im Hintergrund von einem Drittanbieter für den Tag, Uhrzeit und Ort eine Wettervorhersage gezeigt. Daraufhin werden die herausgeforderten Benutzer über den Terminvorschlag informiert und können entsprechend die Herausforderung akzeptieren oder ablehnen.

Nach dem erfolgreichen Match werden die Ergebnisse in der Ergebnisliste eingetragen, natürlich nur, wenn das Spiel vorher gespielt worden ist. Am Schluss werden entsprechenden Quellen und Credits gezeigt, wie beispielsweise die verwendete Hintergrundmusik.

Teil VI.
MEILENSTEIN 6

9. Prozessassessment

Die einzelnen Prozesse stellen eine Herausforderung dar in Hinsicht auf die Effektivität und die Effizienz des Projektes. Als Erstes lag der Fokus auf dem MCI-Teil und wurde in der Vorgehensweise abgewogen und weitgehend auf das Projekt begrenzt. Hier war es gut die Merkmale der Benutzer als Ausgangspunkt für die Entwicklung des Systems zu verwenden.

Um die Ressource Zeit zu sparen, wurden der Ansatz von Nielsen, Discount Usability Engineering gewählt. Dabei haben sich die „paperbased“ Prototypen mit der „think aloud“ Technik als sehr nützlich erwiesen. Durch die einfache Umsetzung wurden einige Mängel sichtbar und konnten durch weitere Iteration beseitigt werden, dies wurde in den Evaluationsprozess beschrieben. Im späteren Verlauf sollten auch die „screen“ Prototypen zur Evaluation hinzugezogen werden.

Es wurde versucht die DIN-Norm EN ISO 9241, Teil 110 und 210 bei der Entwicklung des Systems zu berücksichtigen. Leider konnte man aus Zeitgründen nicht explizit auf die einzelnen Grundsätze eingehen, deshalb stellt dieser Punkt Verbesserungspotenzial dar. Bei der Analyse wurden Stakeholder, Benutzerprofile, Personas sowie einige Szenarien erstellt. Anhand dessen und der anonymen Personenbefragung hatte man Erkenntnisse gezogen zu der speziellen Domäne. Insbesondere der Fragebogen ist als kritisch anzusehen, da nur ein beschränkter Personenkreis zur Verfügung stand. Durch die Analyse konnten Stakeholder identifiziert werden und der Objektbereich der Beziehungen. Mithilfe der Szenarien konnte man aus der Sicht der Benutzer diskutieren und daraus die entsprechenden User Needs beschreiben.

Aus den Bedürfnissen der Benutzer resultieren die Anforderungen mit einige Kritikpunkten. So sollte man diese genau formulieren, da dies ein wichtiges Hauptmerkmal darstellt. Außerdem sollten die technischen Aspekte im späteren Verlauf des Projektes beschreiben werden. Um besseren Einblick zu bekommen wurde eine Aufgabe „Spiel 21 gespielt in kleinere Teilaufgaben heruntergebrochen durch die deskriptive HTA. Hier wurde es deutlich, dass man einige Schwachstellen in den Anforderungen hatte, vor allem sollte man die Anforderungen priorisieren.

Die Identifikation der Risiken spielte für das Projekt eine wichtige Rolle. Auf Basis dessen konnte eine Spezifikation der PoCs erstellt werden und die Funktionalität des Systems geprüft werden. Als Verbesserungspotenzial wurden die Alternative genannt.

Im WBA-Teil lag der Fokus auf den verschiedenen Modellen. So wurde versucht, ein optimales präskriptives Modell mit Bezug auf das System zu erzeugen. Als Basis wurde ein deskriptives Modell erzeugt und später die verschiedenen Prozesse.

Daraus resultierte die Systemarchitektur mit verschiedenen Abwägungen in Bezug auf das

Architekturdiagramm. Kritischer Punkt ist die Datenbankabwägung. Hier gab es auf der technischen Seite einige Schwierigkeiten mit der Datenbank MongoDB. Es ist nicht einfach mehrere Collections miteinander durch die jeweiligen ID's zu verknüpfen, wie beispielsweise bei der MySQL Datenbank durch Joint Operationen. Hierfür wurden mehrere JavaScript Codezeilen geschrieben, damit dies möglich wurde. Objektiv betrachtet ist es nicht sehr professionell. Außerdem gab es Komplikationen mit dem NodeJS Modul „mongoose“, dieses Modul verursachte Fehler auf der Plattform und kam deshalb nicht zum Einsatz. Das Datenmodell mit den einzelnen Collection ist soweit gut, dennoch sollte es kritisch betrachtet werden. Im Verlauf des Projekts können eventuell weitere Attribute und URI's hinzugefügt werden, beispielsweise durch neue prototypische Funktionen.

Deshalb wurden einige Punkte noch nicht komplett ausgearbeitet, insbesondere die Bereiche der technischen Umsetzung und des vertikalen Prototypen fehlen einige Funktionen. So wurde beispielsweise nur ein vorläufiges Matchmaking mit Pseudo-Code erstellt.

Im Allgemeinen besteht immer die Herausforderung, in bestimmter Zeit und mit bestimmten Budget die geforderte Qualität zu liefern. Doch leider ist es nicht möglich, alle drei Anforderungen zu erreichen.

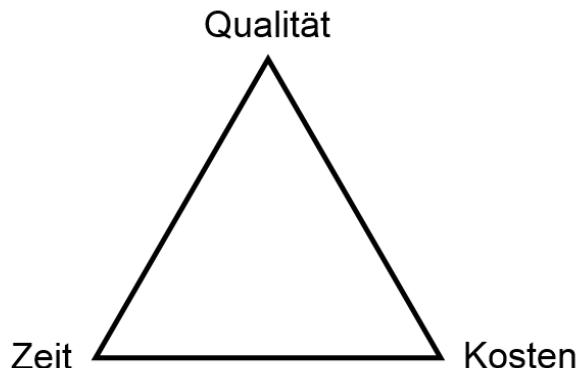


Abbildung 27: Das Dreieck zu Zeit, Qualität und Kosten[24]

10. Fazit

Auf das Projekt bezogen habe ich von den Projektbetreuern eine konstruktive Kritik bekommen, vielen Dank dafür. Bezogen auf die Alleinarbeit erleichterte es einige Entscheidungen bei den Artefakten und sorgte zugleich für Diskussion und bessere Überlegungen. Die Dokumentation versuchte ich systematisch aufzubauen und dabei die verschiedenen Artefakte zu ordnen. Leider wurde zum Schluss die Vorgabe für das Projekt geändert. Laut dem Projektplan habe ich mehr Zeit investiert als geplant, da die einzelnen Artefakte nicht leicht zu erreichen gewesen sind. Besonders das Abwägen bestimmter Entscheidungen viel

mir schwer. In der Dokumentation habe ich versucht meinen weitläufigen Formulierungsstil zu begrenzen.

In Bezug auf den vertikalen Prototypen wurden die meisten Ziele erreicht. Beispielsweise kann sich der Benutzer im System Registrieren, Anmeldung, andere Benutzer zum Match einladen, Spielergebnisse ansehen und Wetterinformationen abrufen. Aus zeitlichen Gründen konnte keine Benachrichtigung zu den einzelnen Matches realisiert werden. Für die Zukunft sollte es jedoch möglich sein, die einzelnen Ziele zu erreichen und zu vervollständigen.

11. Installationsanleitung

Für einen ordnungsgemäßen Gebrauch mit dem System werden verschiedene Voraussetzungen erwartet. Beispielsweise wird ein Smartphone mit Android als Betriebssystem (min. Version 4.4, KitKat), Internetverbindung und GPS Sensoren erwartet. Bei den Servern können die gängigen Systeme, wie beispielsweise Windows, OSX oder Linux zum Einsatz kommen.

11.1. NodeJS Server

- Man lädt die serverseitige Plattform NodeJS auf den Computer für das jeweilige Betriebssystem <https://nodejs.org/download/>.
- Anschließend installiert man die Plattform NodeJS, einfach den Installationsanweisungen folgen.
- Nun kopiert man die „Spiel 21“ Serverdateien auf den Computer.
- Durch die Eingabe in der Konsole `npm install` werden die benötigten NodeJS Module aus dem Internet geladen und automatisch installiert.
- Zum Schluss startet man in der Konsole den NodeJS-Server durch den Befehl `node spiel21.js`.

11.2. MongoDB Datenbank

- Man lädt die schemafreie Datenbank MongoDB auf den Computer für das jeweilige Betriebssystem <https://www.mongodb.org/downloads>.
- Anschließend installiert man die Datenbank MongoDB, einfach den Installationsanweisungen folgen.
- Nun erstellt man ein Verzeichnis für die Datenbank.
- Nun startet man die Datenbank mit der Angabe des erstellten Verzeichnisses mit folgenden Befehl `mongod dbpath --[Pfad+Verzeichnis]`.

- Optional: kann man die Datenbank mit Beispieldaten (Moke-Up) befüllen. Dazu sollten die entsprechenden JSON-Dateien aus dem Verzeichnis „database“ genommen und als Collection in MongoDB eingefügt werden, beispielsweise durch Robomongo <http://robomongo.org>.

11.3. Android Anwendung

- Man lädt die Android Anwendung „com.spiel21.application.apk“ herunter.
- Anschließend installieren man die Anwendung, einfach den Installationsassistent folgen.
- Hinweis: Die eingetragene URL des Server steht standardmäßig auf <http://10.0.2.2:3000> und sollte entsprechend angepasst werden.

12. Projektplan

Name	Abgabe	Aktivität (Artefakte)	1. Unteraktivität	2. Unteraktivität	Workload geplant	Workload tatsächlich
laufend		Beratung			6,50	7,00
		Workshops	3x		2,50	2,50
		Beratungstermine	12x		4,00	4,50
MS1	13.04.2015	Expose			14,00	22,00
		Idee finden			2,00	5,00
			Brainstorming		4,00	5,00
			Recherche		6,00	8,00
		Ausarbeitung	Ergebnis dokumentieren		2,00	4,00
MS2	27.04.2015	Dokumentation			46,00	70,00
		Recherche	Einleitung		1,00	1,50
				Problemraum	1,00	1,50
				Lösungsansatz	0,50	1,00
		Zielhierarchie	langfristige		2,00	1,00
			mittelfristige		2,00	2,50
			kurzfristige		2,00	3,00
		Markrecherche	Wetterdienste finden		1,00	1,50
			Nachrichtenaustausch (Messenger)		1,00	1,50
			Navigation (GPS Funktionalitäten)		1,00	2,50
MS3	11.05.2015	Alleinstellungsmerkmale			1,00	3,00
		Meth. Rahmen (MC)	Nutzungskontextanalyse		1,00	1,50
			Stekholderanalyse		2,00	2,00
			Anforderungsermittlung	Zielbestimmung und	3,00	4,00
				Funktionale Anford.	2,00	2,00
				Nichtfunk. Anforderungen	1,00	2,00
			Wahl des Vorgehensmodells		4,00	5,00
				User Centered Design	2,00	2,00
		Kommunikationsmodell	Deskriptives		2,00	3,00
			Prässktiptives		2,00	3,00
MS4	01.06.2015	Risiken	Kontinuität		1,00	2,00
			Datenschutz		1,00	2,00
			Werbung		1,00	2,00
			Technisch		1,00	2,00
			Projektinterne		1,00	2,00
		Spezifikation der PoCs	Matchmaking		1,50	2,00
			Client		1,50	2,00
			Server		1,50	2,00
		Architekturdiagramm	Systemarchitektur		1,00	2,00
		(inkl. Architekturbegründung)	Endgeräte		1,00	2,00
MS5	18.06.2015		Server		1,00	2,50
			Middleware		1,00	2,00
			Drittanbieter		1,00	2,00
		Dokumentation			53,50	62,50
		Nachbearbeitung der Artefakte			3,00	4,00
			Überarbeitung der PoCs inkl. Dokumentation		4,00	4,50
			Überarbeitung der MCI Methoden		3,00	2,00
			Personas		2,00	3,00
			Wahl des vorgehensmodells		2,00	2,00
			MCI Vorgehensweise		4,00	5,00
MS6	25.06.2015		Überarbeitung des Architekturdiagramms		1,50	2,00
		Einarbeiten in Android Studio inkl Installation			12,00	18,00
		Aufbau vom Server			4,00	3,00
			Testen der Netzwerkstruktur		5,00	7,00
			Datenaustausch, Login		4,00	5,00
			Nachrichtenaustausch (Chat)		4,00	3,00
			Implementierung der Wetterdaten		5,00	4,00
		Dokumentation, Programmierung			56,00	56,00

		Projektplan überarbeiten		5,00	4,00
		Nachbearbeitung der Artefakte		2,00	2,00
		Überarbeitung MCI Rahmenteil		4,00	6,00
		Überarbeitung Aufgabenmodellierung		2,00	4,00
		Einfügen der HTA		2,00	3,00
		Umfragebogen inkl Auswertung		4,00	6,00
		Server schreiben		6,00	10,00
		Datenstrukturen		10,00	4,00
		Datenbankmodell		2,00	3,00
		Matchmaking Logik erstellen		2,00	1,00
		WBA-Modellierungen		12,00	6,00
		Prototypen UI		10,00	11,00
MS5	15.06.2015	Dokumentation, Programmierung		82,00	87,00
		Projektplan überarbeiten (MS5 und MS6 fein planen)		5,00	4,00
		funktionale Prototypen		40,00	30,00
		Login-Bereich		2,00	3,00
		Registrierung Bereich		3,00	4,00
		Datenmodell (inkl Mock-Up Daten)		3,00	5,00
		Ref der Daten		2,00	8,00
		Server schreiben	GET-Methode	1,00	1,00
			Android APK, GET-Methode Implement	5,00	8,00
			Aufbau, Verfeinerung der Datenstruktur	2,00	4,00
		Kürzung der Dokumentation		6,00	6,00
		Evaluationsergebnisse UI		7,00	8,00
		narratives Konzept für filmische Präsentation		6,00	6,00
MS6	29.06.2015	Dokumentation, Programmierung		37,00	46,50
		Code refactoring		5,00	7,00
		Implementationspräsentation		3,00	4,00
		Dokumentantation mit Latex ausarbeiten		8,00	12,00
			Prozessassessment	3,00	5,00
			Evaluationsergebnisse UI	8,00	8,00
			Fazit	2,00	2,00
			Installationsdoku	2,00	2,50
		Implementationsabgabe		2,00	2,00
		Filmszenen vorbereiten, ausarbeiten		4,00	4,00
	06.07.2015	Präsentation		23,00	0,00
		Projektpräsentation		2,00	0,00
		Video drehen, vertonen		8,00	0,00
		Video schreiben, rendern		12,00	0,00
		Abgabe des Films		1,00	0,00
Gesamtzeiten				318,00	351,00

13. Quellenverzeichnis

- [1] *Androide Style Guides.* <https://developer.android.com/design/style/index.html>. – Zugriff: 22.05.2015
- [2] *App Stroe.* <https://itunes.apple.com/de/genre/ios/id36?mt=8>. – Zugriff: 08.06.2015
- [3] *Facebook Messenger.* <https://play.google.com/store/apps/details?id=com.facebook.orca>. – Zugriff: 20.04.2015
- [4] *Google Maps.* <https://play.google.com/store/apps/details?id=com.google.android.apps.maps>. – Zugriff: 20.04.2015
- [5] *Google Maps API.* https://www.google.com/intx/en_uk/work/mapsearch/products/mapsapi.html. – Zugriff: 12.06.2015
- [6] *Google Play Store.* <https://play.google.com/store>. – Zugriff: 08.06.2015
- [7] *MAPS.ME Offline Karte und Routen.* <https://play.google.com/store/apps/details?id=com.mapswithme.maps.pro>. – Zugriff: 20.04.2015
- [8] *mongoDB.* <https://www.mongodb.org/>. – Zugriff: 21.04.2015
- [9] *MySQL relationale Datenbankverwaltungssystem.* <https://www.mysql.de/>. – Zugriff: 25.06.2015
- [10] *NAVIGON select.* https://play.google.com/store/apps/details?id=com.navigon.navigator_select. – Zugriff: 20.04.2015
- [11] *Node.js.* <https://nodejs.org/>. – Zugriff: 21.04.2015
- [12] *OpenWeather API.* <http://openweathermap.org/api>. – Zugriff: 01.05.2015
- [13] *PHP: Hypertext Preprocessor.* <http://php.net/>. – Zugriff: 25.06.2015
- [14] *TomTom GPS-Navigation Traffic.* <https://play.google.com/store/apps/details?id=com.tomtom.gplay.navapp>. – Zugriff: 20.04.2015
- [15] *Wetter Online.* <https://play.google.com/store/apps/details?id=de.wetteronline.wetterapp>. – Zugriff: 20.04.2015
- [16] *wetter.com.* <https://play.google.com/store/apps/details?id=com.wetter.androidclient>. – Zugriff: 20.04.2015
- [17] *WhatsApp Messenger.* <https://play.google.com/store/apps/details?id=com.whatsapp>. – Zugriff: 20.04.2015

- [18] CAVELTI, M. : *Spielen: Basketball Twenty-One*. <http://www.sportunterricht.ch/lektion/spielen/spielen80.php>. – Zugriff: 17.06.2015
- [19] COCKBURN, A. : *Use Cases effektiv erstellen*. MITP-Verlag, 2003
- [20] COOPER, A. : *About Face: Interface und Interaction Design*. MITP-Verlag, Bonn, 2009
- [21] DIN: *Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung (ISO 9241-110:2006)*. DIN Deutsches Institut für Normung, 2008
- [22] DIN: *Ergonomie der Mensch-System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010)*. DIN Deutsches Institut für Normung, 2011
- [23] IEEE: *610.12-1990 - IEEE Computer Society*. IEEE Standard Glossary of Software Engineering Terminology, 1990
- [24] KNEUPER, R. : *CMMI: Verbesserung von Softwareprozessen mit Capability Maturity Model Integration*. dpunkt Verlag, 2007
- [25] NIELSEN, J. : *10 Usability Heuristics for User Interface Design*. <http://www.nngroup.com/articles/ten-usability-heuristics/>. – Zugriff: 27.04.2015
- [26] NIELSEN, J. : *Discount Usability: 20 Years*. <http://www.nngroup.com/articles/discount-usability-20-years/>. – Zugriff: 27.04.2015
- [27] SHEPHERD, A. : *Hierarchical Task Analysis*. Taylor and Francis, London and New York, 2001
- [28] TILKOV, S. : *REST und HTTP: Entwicklung und Integration nach dem Architekturstil des Web*. dpunkt Verlag, 2015

Anhang

A. Spielregeln

Das Spiel kann zu zweit oder maximal mit vier Spielern gespielt werden in der Regel[18]. Bei mehr Spielern werden die Wartezeiten länger. Dabei spielt Jeder gegen Jeden. Für das Spiel werden ein Basketball und ein Basketballkorb benötigt. Im folgenden wird das Spiel anhand Spieler A und Spieler B erläutert.

Zu Spielbeginn wirft Spieler A den Ball von der Freiwurflinie aus auf den Korb. Trifft er, erhält dieser zwei Punkte und kann sogleich einen weiteren Wurf tätigen. Trifft er nicht fängt Spieler B den Ball in der Luft oder nachdem dieser maximal einmal auf dem Boden aufgeprallt ist. Spieler B muss nun genau von dieser Position einen Korbwurf tätigen. Trifft er, erhält er einen Punkt und darf den nächsten Wurf von der Freiwurflinie aus machen usw. Zwei Punkte erhält man dann, wenn der Ball von der Freiwurflinie (Abb. 28) aus geworfen wird.

Berührt ein Ball beim Wurf weder das Brett noch den Korb, so ist dies ein Airball. Dies bedeutet, dass der Spieler wieder bei Null anfangen muss. Wer zuerst 21 Punkte erreicht, gewinnt.

Natürlich kann das Spiel nicht nur in dieser Variante gespielt werden. Es ist jederzeit möglich, die Regeln auf das Können der Spieler anzupassen. So kann beispielsweise bei den Fortgeschrittenen Spielern der letzte Wurf hinter der Dreipunktelinie aus erfolgen.

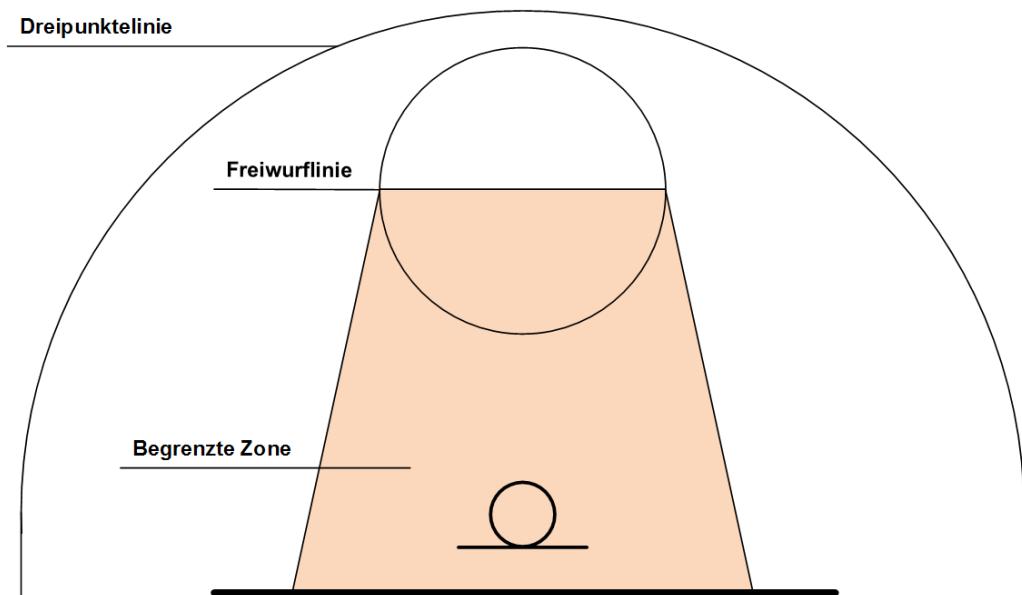


Abbildung 28: Schematische Darstellung des Basketballfeldes

B. Marktrecherche (related-works)

Im Rahmen des Systems werden die Anwendung im Google[6] und Apple[2] Store betrachtet, dabei wird nicht auf die Webanwendungen weiter eingegangen, da diese keine Erfordernis für das jetzige System darstellen. Im Mittelpunkt stehen Konkurrenzprodukte, die einige Teilespektive für das Projekt darstellen. Wie beispielsweise eine Übermittlung von Wetterinformationen, Navigation sowie eine Kommunikationsmöglichkeit zwischen den Benutzern.

B.1. Wetterdienste

Ein Wetterdienst wie beispielsweise [wetter.com](#)[16] und [Wetter Online](#)[15] bietet Wettervorhersagen an. Hierbei kann die Lokalisierung entweder durch den Namen, PLZ oder über die GPS Ortung erfolgen. Außerdem können Regenwarnungen ausgegeben werden. Die Dienste nutzen automatisch denn zur Verfügung gestellten Netzwerk, entweder den mobilen oder einen stabilen WLAN Datentarif.

Deutliche Nachteile der Anwendung sind nicht feststellbar. Außerdem bietet [wetter.com](#) eine kostenlose Programmierschnittstelle für nicht kommerzielle Projekte. Daher eignet es sich gut zur Implementierung in das System als Drittanbieter.

B.2. Nachrichtenaustausch (Messenger)

Es gibt diverse Messenger, eines der Beliebtesten davon sind WhatsApp[17] und Facebook[3]. Der größte Vorteil solcher Messenger sind die geringen Gebühren zum Vergleich für SMS. Zusätzlich bieten die Messenger eigene Profilverwaltung mit eigenen Namen und Profilbild. So können die Benutzer auf einfache Weise in Kommunikation treten und Textnachrichten versenden.

Einige Nachteile solcher Dienste sind meist die Geschäftsbedingungen, da diese den Datenschutz nicht genau wahrnehmen. Bei WhatsApp handelt es sich um eine kostenlose Testversion für ein Jahr, danach muss ein Abo abgeschlossen werden. Leider sind solche Lösungen von den Messenger eher weniger geeignet, da diese keine Programmierschnittstelle bieten. Alternativ kann man eigene Nachrichtenaustauschfunktion einbauen implementieren, wie beispielsweise einen Chat.

B.3. Navigation (GPS-Datenübertragung)

Die Navigationsprodukte unterscheiden sich meist in kostenlosen und kostenpflichtigen Varianten, sowie offline und online Navigation. Einige der kostenpflichtigen sind beispielsweise NAVIGON select[10] und TomTom GPS-Navigation Traffic[14]. Diese bieten sehr viele Vorteile und sehr viele Funktionen, wie beispielsweise die Echtzeit-Verkehrsinformationen.

Nachteil ist wohl die fehlende Programmierschnittstelle für das angedachte System, deshalb wird hier nicht mehr weiter auf die kostenpflichtigen Produkte eingegangen.

Eine Alternative sind die kostenlosen Produkte, wie beispielsweise GoogleMaps[4] und MAPS.ME Offline Karte & Routen[7]. Bei GoogleMaps handelt es sich um eine online Navigation und ist unter Umständen mit hohen Datenaufkommen verbunden. Einer der größten Vorteile ist die offene Programmierschnittstelle.

Bei MAPS.ME dagegen handelt es sich um eine offline Navigation und hat den Vorteil, dass es keine weiteren Karten geladen werden müssen. Nachteilig ist dann die Speichergröße der Anwendung bzw. der Karten die man vorher installieren müsste.

In den Rahmen des Projekts kommt daher nur die online Navigation in Frage.

C. Fragebogen

Fragebogen Spiel 21

WICHTIG! Der Fragebogen wird anonym ausgewertet.

Wie alt sind Sie?

____ Jahre weiblich männlich

Was machen Sie beruflich?

Wo wohnen Sie?

Wie lange wohnen Sie da?

weniger als 1 Jahr 1 Jahr 5 Jahre länger

Haben Sie ein eigenes Auto?

ja nein

Besitzen Sie ein internetfähiges Endgerät (bspw. Smartphone)?

ja nein

Wenn ja, habe Sie einen Volumentarif oder Flatrate?

Volumen Flat

Schätzen Sie, wie viel Freizeit habe Sie in der Woche?

____ Stunden

Haben Sie schon mal Ihr Endgerät zum suchen von Feizeitmöglichkeiten verwendet?

ja nein

Haben Sie am Wochenende Langeweile?

() nie () selten () oft () immer

Nutzen Sie Angebote von Sportklubs oder ähnlichen Einrichtungen?

() ja () nein

Wie wichtig ist für Sie Sport?

() nicht wichtig () wenig wichtig () wichtig () sehr wichtig

Machen Sie Sport in ihrer Freizeit?

() ja () nein

Wenn ja, welche Sportart

() Fitnessstudio () Fußball () Basketball
() Tischtennis () Schwimmen () Volleyball
() Andere _____

Kennen Sie das Basketballspiel 21 (Twenty-One)?

() ja () nein

Vielen Dank, dass du dir die Zeit genommen hast, den Fragebogen auszufüllen.

Text für die Online Umfrage:

Umfrage im Rahmen der Veranstaltung „Entwicklungsprojekt interaktive Systeme“.

Im Rahmen der Veranstaltung „Entwicklungsprojekt interaktive Systeme“ möchte ich gerne mithilfe Eurer Meinung wichtige Informationen zu meinem Projekt dazu ermitteln.

Ich würde mich sehr freuen, wenn ihr Euch etwa 10 Minuten für die Umfrage zeitnehmen könntet. Die Befragung erfolgt anonym und es sind keinerlei Rückschlüsse auf einzelne Personen möglich.

Zur Umfrage gelangt ihr über diesen Link: <http://goo.gl/forms/vziaD5ebwb>

Über Eure Unterstützung würde ich mich sehr freuen!

Vielen Dank im Voraus.

Auswertung des Fragebogen

Fragen	Zeitstempel: 14.05-17.05.2015	Befragtenanzahl: 32 Personen
Wie alt sind Sie?	Weiblich: 27,85 Männlich: 27,85 Durchschnitt: 26,56	
Welches Geschlecht?	Weiblich: 8 Männlich: 24	
Was machen Sie beruflich?	Es wurden meistens Studenten gefragt	
Wo wohnen Sie?	Die meisten befragten kommen aus Gummersbach	
Wie lange wohnen Sie da?	weniger als 1 Jahr: 3 1 Jahr: 10 5 Jahr: 6 Länger: 13	
Haben Sie ein eigenes Auto?	Ja: 19 Nein: 13	
Besitzen Sie ein internetfähiges Endgerät (bspw. Smartphone)?	Ja: 32 Nein: 0	
Wenn ja, habe Sie einen Volumentarif oder Flatrate?	Flatrate: 20 Volumen: 12	
Schätzen Sie, wie viel Freizeit habe Sie in der Woche (in Stunden)?	Durchschnitt: 26,15	
Haben Sie schon mal Ihr Endgerät zum suchen von Feizeitmöglichkeiten verwendet?	Ja: 23 Nein: 9	
Haben Sie am Wochenende Langeweile?	Nie: 5 Selten: 25 Oft: 2 Immer: 0	
Wenn ja, welche Sportart?	Basketball: 2, Rest: 30 Am meisten: Fitnessstudio	
Machen Sie Sport in ihrer Freizeit?	Ja: 27 Nein: 5	
Kennen Sie das Basketballspiel 21 (Twenty-One)?	Ja: 15 Nein: 17	

Tabelle 7: Auswertung des Fragebogens

D. Personas



Alter: 17
Beschäftigung: Azubi
Status: single
Sport: Basketball, Fußball
Mobilität: Fahrrad

Markus

Er ist 17 Jahre alt, ein Jugendlicher und arbeitet bei der Fa. Saturn in Gummersbach. In seiner Freizeit spielt er gerne Fußball mit seinen Freunden. Dennoch würde er gerne Basketball spielen. Sowie früher, das Spiel 21, kennt er noch aus der Realschule.

Er versucht immer wieder seine Freunde zu überreden aber hat leider keinen Erfolg. Dabei würde er so gerne ein paar Körbe werfen.

Er hat Abend immer viel Zeit und geht gerne raus beim schönen Wetter, wenn es nicht regnet. Dann kann man am besten das Wetter genießen bei einem Spielchen unter Freunden.

Am liebsten verabredet er sich mit seinen Freunden über das Smartphone.

Tabelle 8: Persona Markus



Alter: 34
Beschäftigung:
Sachbearbeiterin
Status: verheiratet
Sport: Jogging, Basketball
Mobilität: Auto, Fahrrad

Anna

Sie ist eine Mutter mit einem Kind im Alter von 6 Jahren.

Sie hat einen neuen Job als Sachbearbeiterin in Gummersbach bei der Stadt. In der Freizeit beschäftigt sie sich gerne mit Sport.

Sie versucht sich selbst immer zum Sport zu animieren, doch es fällt leider schwer die sportlichen Aktivitäten alleine auszuüben. Den einer muss immer auf das Kind auf aufpassen, deshalb kann der Ehemann nicht mitkommen.

Am liebsten spielt sie Basketball. Doch leider kann sie keine Freunde finden in der Umgebung. Nach dem Umzug kennt sie sich in Gummersbach noch nicht so gut aus.

Die Zeit ist knapp und sie würde gerne neue Leute kennenlernen die genau soviel Spaß haben an sportlichen Aktivitäten wie sie.

Sie benutzt ihr Smartphone um mit Ihren Freunden aus der alten Umgebung in Kontakt zu bleiben.

Tabelle 9: Persona Anna

E. Szenarien

Szenario 1: Langweile Es ist 15:00 Uhr und Henry D. macht heute früher Feierabend. Er würde gerne noch irgend eine sportliche Aktivität machen, bevor er komplett für heute Abend abschaltet. Er langweilt sich nur immer, das seine Kollegen immer wieder nur Fußball spielen wollen. Dabei gibt es doch so viele andere Sportarten. Er würde gerne was neues ausprobieren, wie beispielsweise Basketball.

Szenario 2: Wechselhafte Wetterbedingungen Daniel und Markus sind beste Freunde und spielen gerne das Basketballspiel 21 zusammen. Durch das wechselhafte Wetter in letzter Zeit haben die beiden Jungs mittlerweile kein Lust zum spielen und bleiben lieber gerne zu Hause und spielen Computerspiele. Des Weiteren kommen immer wieder Streitereien zwischen den beiden, da sie sich den Punktestand nicht merken, wer zuletzt gewonnen hat. Dabei wünschen sich die beiden nicht sehnlicher als eine Anwendung, wo sie ihren Punktestand aufschreiben können und durch den bevorstehenden Regen gewarnt werden.

Szenario 3: Begeisterter Basketballspieler Peter M. ist ein begeisterter Basketballspieler. Schon seit längeren möchte er gerne mehr Aufmerksamkeit auf die Sportart lenken und versucht die jungen Leute zu begeistern. Eines seiner Konzepte ist das Organisieren von Wettkämpfen und das sammeln von Statistiken aus den Spielbegegnungen. Dabei würden einige der Spieler und Zuschauer begrüßen, wenn die Ergebnisse des Spiels öffentlich zugänglich wären.

Szenario 4: Neu in der Umgebung Maria K. studiert seit zwei Semestern an der FH-Köln und ist relativ neu in der Umgebung. Sie wohnt in einer WG und hat ab Freitag bis Sonntag Freizeit und würde gerne sportlich aktiv sein, ohne sich in ein Sportverein einschreiben zu müssen. Seit dem Umzug hat sie leider immer noch keine Kenntnis von der Umgebung und kennt sehr wenige Leute aus der Umgebung. Dabei würde sie so gerne neue Leute kennen lernen, ohne irgendwelche Fitness-Verträge abschließen zu müssen.

Szenario 5: Sportunterricht Die Lisa S. ist die Klassenbeste im Sportunterricht. Sie liebt es Basketball zu spielen, und kennt noch das Spiel 21 aus dem Sportunterricht. Dies findet allerdings leider äußerst selten statt, da das jetzige Sportprogramm dies nicht mehr vorsieht. Ebenso kann sie ihre Klassenkameraden nur schwer für das spezielle Variante des Basketballspiel begeistern. Nur unter größerem Aufwand und viel Überredungskunst ist es jedoch spontan möglich, einige Partien unter den Schülern zu starten.

F. Use Cases

Nummer, Titel:	1. Start der Anwendung.
Vorbedingung:	Die Anwendung wurde installiert und gestartet.
Standardablauf:	<p>Der Benutzer möchte die Anwendung nutzen. Hier stehen zwei Optionen zur Verfügung:</p> <ol style="list-style-type: none"> 1. Login 2. Registrierung
Nachbedingung:	<ol style="list-style-type: none"> 1. Besteht bereits ein Account so kann der Benutzer erfolgreich sich einloggen und landet im Menü (Use Case 3). 2. Besteht kein Account, so muss sich der Benutzer vorher registrieren (Use Case 2).

Tabelle 10: Use Case: 1

Nummer, Titel:	2. Registrierung des Benutzers.
Vorbedingung:	Der Benutzer trägt seine persönlichen Daten (Name, Geschlecht, Email, Passwort, etc) ein.
Standardablauf:	Stimmen die Daten überein und sind vollständig, so können diese dann abgesendet werden.
Nachbedingung:	Benutzer sendet die eigenen Daten ab und wird wieder zum Login (Use Case 1) weitergeleitet.

Tabelle 11: Use Case: 2

Nummer, Titel:	3. Navigation und Menü.
Vorbedingung:	Use Case 1 muss erfüllt sein.
Standardablauf:	<ol style="list-style-type: none"> 1. Benutzer kann die verschiedenen Menüpunkte ansteuern. 2. Die Überschrift zeigt an, wo sich der Benutzer zur Zeit befindet.
Nachbedingung:	Trifft der Benutzer eine bestimmte Auswahl im Menü, so wird dieser zu den jeweiligen Punkt navigiert (Use Case 4 bis 12).

Tabelle 12: Use Case: 3

Nummer, Titel:	4. Spielergebnisse
Vorbedingung:	Use Case 1 muss erfüllt sein.
Standardablauf:	1. Der Benutzer kann die zuletzt gespielten Spiele und deren Spielergebnisse einsehen. 2. Der Benutzer kann seine eigenen Ergebnisse filtern.
Nachbedingung:	Eine Detailansicht der zuletzt gespielten Spiele wird geladen.

Tabelle 13: Use Case: 4

Nummer, Titel:	5. Herausforderungen.
Vorbedingung:	Use Case 1 muss erfüllt sein.
Standardablauf:	1. Der Benutzer wählt das Datum und die Uhrzeit aus. 2. Zu den ausgewählten Zeitpunkt werden Wetterinformationen angezeigt. 3. Benutzer wählt ein Basketballplatz aus (Use Case 7). 4. Benutzer wählt seine Gegenspieler aus (Use Case 6) oder lässt sich durch das System automatisch welche vorschlagen. 5. Anschließend fordert man die Spieler heraus.
Nachbedingung:	Die Daten wurden korrekt eingegeben und die herausforderten Spieler werden benachrichtigt (Use Case 11).

Tabelle 14: Use Case: 5

Nummer, Titel:	6. Gegenspieler wählen.
Vorbedingung:	Use Case 5 muss erfüllt sein.
Standardablauf:	Der Benutzer kann einen Gegenspieler suchen und auswählen oder einen von System sich vorschlagen lassen.
Nachbedingung:	Eine Detailansicht über die jeweiligen Spieler im System werden angezeigt.

Tabelle 15: Use Case: 6

Nummer, Titel:	7. Basketballplatz auswählen.
Vorbedingung:	Use Case 5 muss erfüllt sein.
Standardablauf:	Der Benutzer kann einen Basketballplatz auswählen oder selbst auf der Karte hinzufügen.
Nachbedingung:	Der gewählte Basketballplatz wird angezeigt.

Tabelle 16: Use Case: 7

Nummer, Titel:	8. Nachrichten versenden.
Vorbedingung:	Use Case 1 muss erfüllt sein.
Standardablauf:	Der Benutzer kann mit den anderen Spielern über den Chat mit Textnachrichten kommunizieren.
Nachbedingung:	Der Benutzer sieht die ein und ausgehende Textnachrichten.

Tabelle 17: Use Case: 8

Nummer, Titel:	9. Wetter
Vorbedingung:	Use Case 1 muss erfüllt sein.
Standardablauf:	1. Der Benutzer kann Wetterinformationen über seinen Standort abrufen, diese kann automatisch über GPS lokalisiert werden. 2. Der Benutzer kann den Standort manuell bestimmten und die Vorhersage kontrollieren in den Einstellungen.
Nachbedingung:	Eine Detailansicht mit den Wetterinformationen mit den gewünschten Einstellungsergebnissen wird angezeigt.

Tabelle 18: Use Case: 9

Nummer, Titel:	10. Einstellungen.
Vorbedingung:	Use Case 1 und 3 muss erfüllt sein.
Standardablauf:	Der Benutzer kann in den Einstellungen: 1. Hilfe: Informationen anzeigen lassen, beispielsweise die Regeln für das Spiel. 2. Profil: Informationen über seinen Account. 3. Wetter: den Standort manuell oder automatisch durch GPS bestimmen und die Vorhersage für 1 Woche wählen.
Nachbedingung:	Die Daten werden korrekt angezeigt und gegebenenfalls gespeichert, beispielsweise manueller Standort.

Tabelle 19: Use Case: 10

Nummer, Titel:	11. Popup Benachrichtigung.
Vorbedingung:	Use Case 1, 5, 6 und 7 muss erfüllt sein.
Standardablauf:	Ein Benutzer im System hat einen anderen herausgefordert. Dieser kann darauf interagieren und die Einladung annehmen oder ablehnen.
Nachbedingung:	Popup wird korrekt dargestellt und bei den Gegenspieler wiedergegeben.
Trigger:	Der Benutzer lehnt die Herausforderung ab.

Tabelle 20: Use Case: 11

Nummer, Titel:	12. Spielergebnis eintragen.
Vorbedingung:	Use Case 1, 5, und (11 wenn angenommen) muss erfüllt sein.
Standardablauf:	Der Benutzer kann nach dem gespielten Spiel das Spielergebnis eintragen.
Nachbedingung:	Das gespielte Spielergebnis wird in den Spielergebnissen angezeigt (Use Case 4).

Tabelle 21: Use Case: 12

G. Screenshots

Logo

Als Logo wurde ein Basketball mit der Nummer 21 am unteren rechten Rand erstellt. Durch dieses Logo können sich die Benutzer in der Zukunft identifizieren. Je nach Endgerät und deren Auflösung existieren verschiedene Größen.

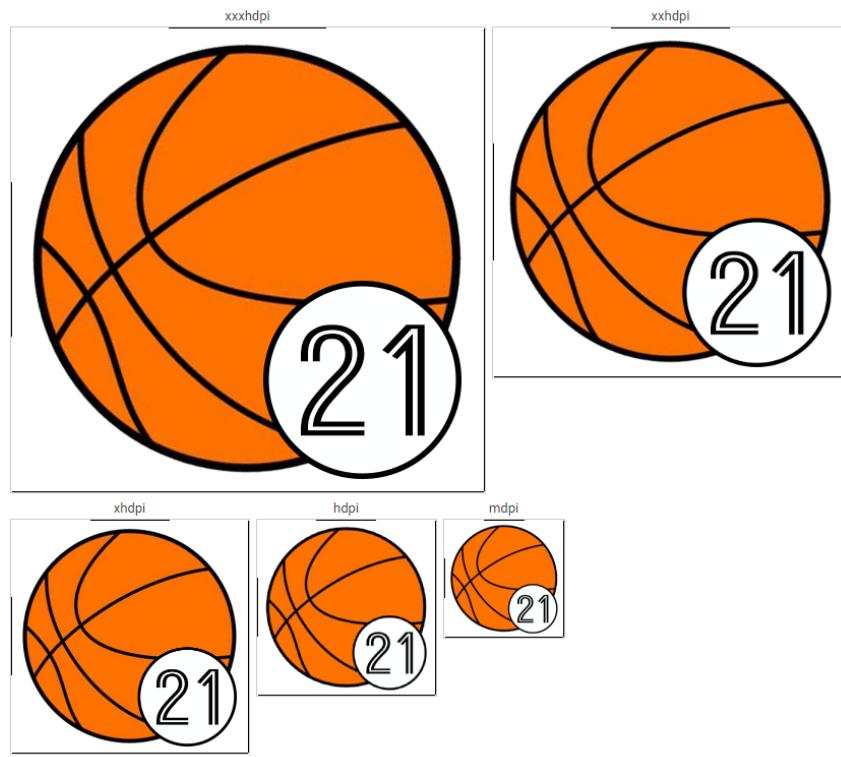


Abbildung 29: Logo Preview

Farben

Folgender Farbspektrum sollte für die Zukunft verwendet werden.

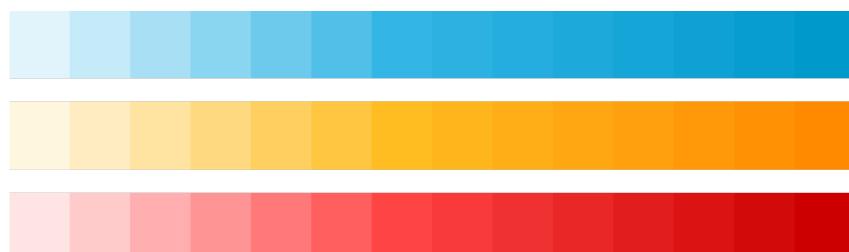


Abbildung 30: Farbspektrum

GUI Prototypen

Hier sind einige Screenshots aus dem vertikalen Prototypen zu sehen.

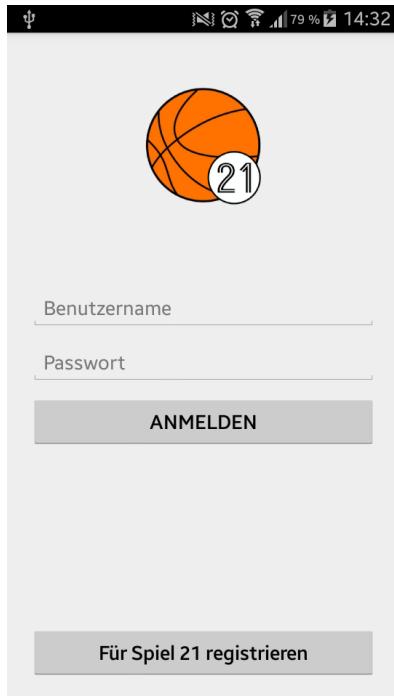


Abbildung 31: UC1



Abbildung 32: UC2

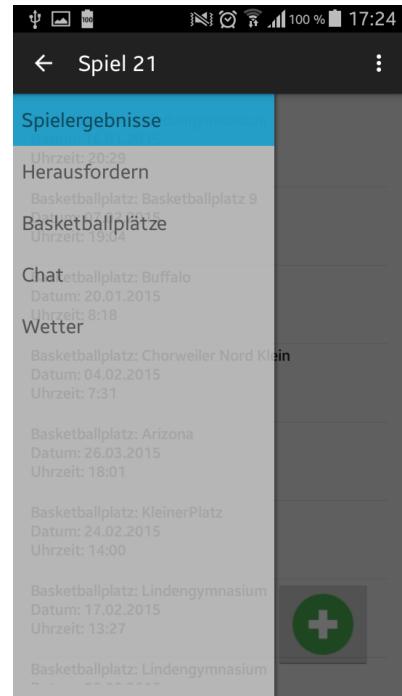


Abbildung 33: UC3



Abbildung 34: UC4



Abbildung 35: UC5



Abbildung 36: UC7

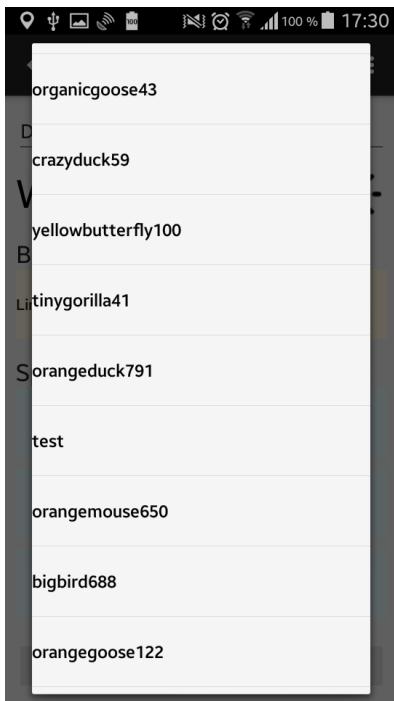


Abbildung 37: UC6

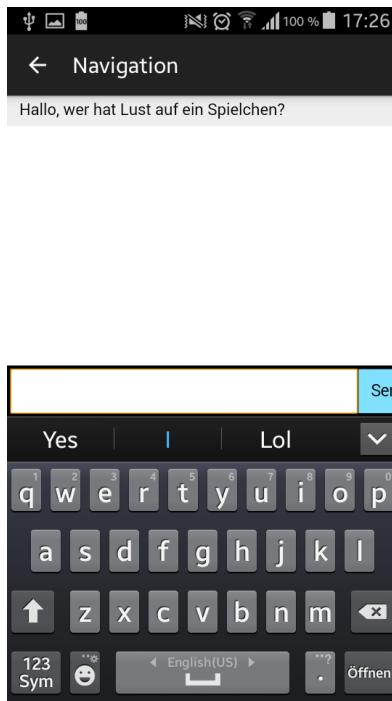


Abbildung 38: UC8



Abbildung 39: UC9

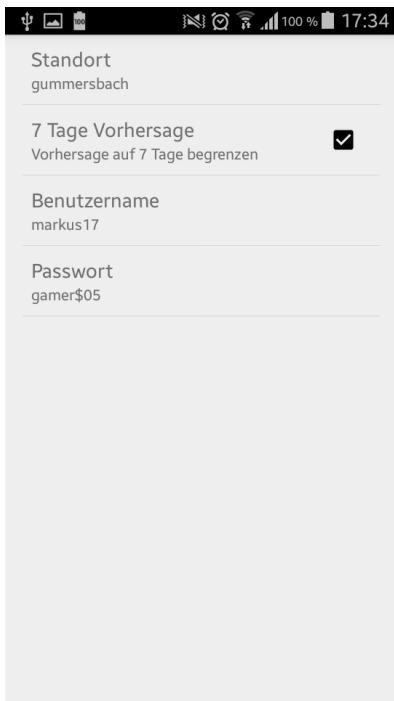


Abbildung 40: UC10