

# **Racketsports Manager**

**Semesterarbeit ZHAW**



**Autor: Raphael Marques**

**Betreuer: Michael Reiser**

13. Juli 2015

# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Inhalt der Arbeit</b>                    | <b>4</b>  |
| 1.1      | Ausgangslage . . . . .                      | 4         |
| 1.2      | Aufgabenstellung . . . . .                  | 4         |
| 1.3      | Erwartete Resultat . . . . .                | 5         |
| 1.4      | Zielsetzung der Arbeit . . . . .            | 5         |
| <b>2</b> | <b>Analyse</b>                              | <b>6</b>  |
| 2.1      | Marktumfeld . . . . .                       | 6         |
| 2.1.1    | Identifizierung von Partnern . . . . .      | 6         |
| 2.1.2    | Terminvereinbarung . . . . .                | 7         |
| 2.1.3    | Amateur-Liga Management . . . . .           | 7         |
| 2.1.4    | Protokollierung eines Spiels . . . . .      | 7         |
| 2.2      | Benutzergruppen . . . . .                   | 7         |
| 2.2.1    | Gelegenheitsspieler . . . . .               | 8         |
| 2.2.2    | Regelmässige Spieler . . . . .              | 8         |
| 2.2.3    | Clubmitglieder . . . . .                    | 9         |
| 2.3      | Use Cases . . . . .                         | 10        |
| 2.3.1    | UC1: Identifizierung von Partnern . . . . . | 10        |
| 2.3.2    | UC2: Terminvereinbarung . . . . .           | 11        |
| 2.3.3    | UC 3: Spiel Protokollieren . . . . .        | 13        |
| 2.3.4    | UC4: Liga Management . . . . .              | 14        |
| 2.4      | Anforderungsanalyse . . . . .               | 15        |
| 2.4.1    | Anforderungen für UC1 . . . . .             | 15        |
| 2.4.2    | Anforderungen für UC2 . . . . .             | 16        |
| 2.4.3    | Anforderungen für UC3 . . . . .             | 17        |
| 2.4.4    | Anforderungen für UC4 . . . . .             | 17        |
| <b>3</b> | <b>Projektmanagement</b>                    | <b>19</b> |
| 3.1      | Projektablauf . . . . .                     | 19        |
| 3.1.1    | Grobplanung . . . . .                       | 19        |
| <b>4</b> | <b>Konzeption</b>                           | <b>20</b> |
| 4.1      | Technologiestack . . . . .                  | 20        |
| 4.2      | Applikationsdesign . . . . .                | 21        |

---

|          |                                 |           |
|----------|---------------------------------|-----------|
| 4.3      | DB Design . . . . .             | 21        |
| 4.4      | GUI Entwurf . . . . .           | 21        |
| <b>5</b> | <b>Implementation</b>           | <b>22</b> |
| 5.1      | REST API . . . . .              | 22        |
| 5.2      | Web Applikation . . . . .       | 22        |
| 5.3      | Android Applikation . . . . .   | 22        |
| <b>6</b> | <b>Test</b>                     | <b>23</b> |
| 6.1      | Unit Tests . . . . .            | 23        |
| 6.2      | System Tests . . . . .          | 23        |
| 6.3      | User Acceptance Tests . . . . . | 23        |
| <b>7</b> | <b>Reflektion</b>               | <b>24</b> |

# 1 Inhalt der Arbeit

Racket-Sportarten haben ein Problem. Man muss die richtige Zeit und den richtigen Partner finden. Die Applikation, welche ich für die Semester-Arbeit erstelle sollte dieses Problem lösen. Durch einen "Doodle-Style Termin und Spieler-Finder sowie eine Liga-Verwaltung werden die Teilnehmenden motiviert, öfters zu squashen.

## 1.1 Ausgangslage

Einen Partner für Racket-Sportarten zu finden, ist nicht immer sehr einfach. Wenn man schliesslich jemanden gefunden hat, ist es immer schwer, einen Termin zu finden. Spielt man regelmässig gegen die gleiche Person, stellt man sich oft auf diese ein und lernt mit der Zeit nur noch wenig dazu.

Diese App sollte diese Probleme lösen, indem man einfach miteinander Termine vereinbaren kann und das Termine vom System für die User automatisch generiert werden.

## 1.2 Aufgabenstellung

Um die Ziele zu erreichen, muss methodisch vorgegangen werden. In der Dokumentation müssen angewendete Technologien begründet werden sowie Konzepte und Architekturen der Applikation festgehalten werden.

1. Informieren über den momentanen Markt und Organisation rund um Racketsport
  - Andere Applikationen rund um Racketsport identifizieren
  - Zielgruppen- und Anwendungsfallanalyse
2. Anforderungsanalyse, welche Funktionalitäten dem Nutzer einen echten Mehrwert bieten
  - Basierend auf (1) eine Anforderungsanalyse erstellen
  - User Stories erstellen
  - Sprints planen
3. Konzeption der App, Technologien wählen
  - Entscheidungsmatrix, welche Technologien verwendet werden
  - DB Design
  - Klassen-Diagramm
  - GUI-Entwurf
4. Die Applikation umsetzen

- Applikation programmieren
- Unit Tests erstellen/ausführen
- 5. Systemtests durchführen
  - Systemtests konzipieren
  - Systemtests durchführen
  - End-to-End Testing
- 6. Benutzertests durchführen
  - Benutzer einladen für Alpha-Test
  - Feedback von Benutzer verlangen

## 1.3 Erwartete Resultat

Folgende Resultate werden erwartet:

- Marktanalyse & Benutzerverhaltensanalyse zur vereinbarung von Spielen
- Anforderungsanalyse
- Recherche für einzusetzende Techniken
- Konzepte und Implementationstechniken
- User Stories und Projektplanung
- Android App und Webapplikation
- Testplan und Umsetzung
- Benutzerfeedback des Alpha-Tests
- Fazit

## 1.4 Zielsetzung der Arbeit

Ziel der Arbeit ist eine WebApplikation sowie eine Android App mit bestimmten Funktionalitäten zu erstellen. Der Autor der Arbeit soll so Zugang zu neuen Technologien im Webbereich erhalten und Erfahrung in der Webapplikations-Programmierung sammeln.

Aufgabenstellung:

- Dokumentation über Entscheidungen und Implementationstechniken
- Webapplikation mit folgender Funktionalität:
  - RestAPI um alle untenstehenden Funktionalitäten
  - Liga erstellen und löschen
  - Einer Liga beitreten und eine Liga verlassen
  - Spiel vereinbaren
  - Termin für Spiel finden
  - Spielresultat eintragen
  - Rangliste der Liga berechnen
  - Automatische Spielvorschläge innerhalb der Liga (z.B. jeder Spieler spielt jede zweite Woche ein Spiel)
- Android App, welche auf die WebAPI zugreift mit gleicher Funktionalität wie WebApplikation

## 2 Analyse

Das Ziel der Software ist, den User in der Terminfindung, Protokollierung und Partnerfindung optimal zu unterstützen. Dieser Teil der Dokumentation dient zur Findung exakter Anforderungen an die Software die den User optimal unterstützen. Zusätzlich wird Anfangs untersucht wie das Marktumfeld rund um die geplante Applikation aussieht, um einen möglichen Erfolg einer solchen Applikation zu schätzen sowie mögliche Synergieeffekte zu identifizieren.

### 2.1 Marktumfeld

Um mögliche Synergien oder Wettbewerber zu identifizieren, müssen zuerst die geplanten Basis Funktionalitäten aufgelistet werden:

- Vereinfachung zur Identifizierung von Partnern
- Vereinfachung zur Terminvereinbarung
- Vereinfachung eines Amateur-Liga Management
- Vereinfachung zu Protokollierung eines Spiels

#### 2.1.1 Identifizierung von Partnern

**GlobalTennisNetwork.com** ist eine Community zur Identifizierung von Tennispartnern. Die Website ist fokussiert auf Tennis. Andere Racket Sportarten werden nicht behandelt. Dadurch sieht der Autor dieser Service nicht als direkte Konkurrenz. Eine Verbindung mit dem Service um eine Grössere Population von Potenziellen Tennispartnern zu erreichen wäre vorstellbar.

**Spontacts** ist eine Android Applikation, welche Spontane Terminvereinbarungen ermöglicht. Es kann zusätzlich als Identifizierung von Partnern für jegliche Freizeitaktivitäten dienen. Es gibt jedoch keine Fokussierung auf Racket Sport. Nichts desto trotz kann Spontacts als konkurrenz zu der geplanten Applikation angesehen werden, da zwei Funktionalitäten (auch wenn der Fokus nicht auf Sport liegt teilweise abgedeckt werden können. Jedoch wird die Konkurrenz nicht als erheblich eingeschätzt.

**sport42.com** bietet auch eine Möglichkeit zur Identifizierung eines Potenziellen Sportpartners. Racket Sportarten werden hier auch abgedeckt. Nach Recherchen stellte sich jedoch heraus, das der Fokus der Applikation auf dem US Markt liegt. Bei einer potenziellen Expansion der geplanten Applikation müsse die Konkurrenz neu evaluiert werden. Da der Fokus

jedoch im Moment auf den deutschsprachigen Raum gerichtet ist, wird dieser Service nicht als Konkurrenz eingeschätzt.

**sportpartner.com** ist ein auf Sport fokussierter Service um Partner zu identifizieren. Der Service deckt auch Racketsportarten wie Tennis, Squash sowie Badminton ab. Bis jetzt gibt es allerdings nur wenig aktive Spieler in Europa.

### 2.1.2 Terminvereinbarung

**Doodle** ist der bekannteste Terminvereinbarungs Service. Ein ähnlicher Algorithmus, welcher in der geplanten Applikation verwendet wird, wurde von den Ersteller dieser Applikation entwickelt. Jedoch gibt es bei Doodle keinerlei Fokus auf Inhalt zum Termin. Es ist möglich Business wie auch Freizeit Termine zu erstellen. Die geplante Applikation kann davon profitieren einen ähnlichen, auf Sport Termine fokussierte Terminvereinbarungs Algorithmus zu entwickeln. Sollte dieser genügend intuitiv umgesetzt sein, wird Doodle nicht als Konkurrenz betrachtet.

**Moreganize.ch** wie Doodle benützt auch Moreganize einen Terminvereinbarungs Algorithmus.

### 2.1.3 Amateur-Liga Management

**LeagueManager Wordpress Plugin** ist eine Zusatzsoftware für Wordpress, mit welchen man Teams und Matches zusammenstellen kann und so Ligen erstellen kann. Der Fokus liegt hier grösstenteils auf Teamsports.

**Excel** von Microsoft wird oft als Tracking und Organisation für Ligen verwendet. Die Einfachheit und das breite Skillset spricht für diese Lösung. Jedoch ist Zusammenarbeit und Organisation umständlich.

### 2.1.4 Protokollierung eines Spiels

**Excel** von Microsoft wird oft als Tracking und Organisation für Ligen verwendet. Die Einfachheit und das breite Skillset spricht für diese Lösung. Jedoch ist Zusammenarbeit und Organisation umständlich. Als alternative, welche die Zusammenarbeit vereinfacht bietet sich auch **Google Docs** an.

## 2.2 Benutzergruppen

Die geplante Applikation wird nicht für alle Benutzergruppen interessant sein. Um das Zielklientel zu finden wird erstellt diese Kapitel eine Kategorisierung in verschiedene Benutzergruppen. Anschliessend wird abgeschätzt, wie welche Benutzergruppe von der Applikation profitieren kann.

## 2.2.1 Gelegenheitsspieler

### 2.2.1.1 Analyse

Gelegenheitsspieler spielen nicht regelmässig (wöchentlich) Racketsport. Sie vereinbaren mit - meistens wenige Personen in engem oder erweitertem privaten Umfeld - unregelmässig Spiele. Oft pausieren Gelegenheitsspieler mehrere Monate und haben anschliessend intensivere Phasen mit mehreren Spielen innerhalb wenigen Wochen. Der Aufwand um ein Spiel zu vereinbaren und einen passenden Court zu finden ist dementsprechend gross. Viel fehlt auch an einem passenden Partner im privaten Umfeld, insbesondere wenn der Spieler Ambitionen hegt sich zu verbessern. Schlussendlich ist die Motivation zu einer „gewissen Regelmässigkeit“ nicht hoch, da die Vereinbarung eines Spieles aufwändig ist, sowie eine Verbesserung des Spiels gegenüber anderen Spielern nicht schnell ersichtlich ist.

Zusammengefasst lassen sich folgende Charakteristiken zusammenfassen:

- Einzelne oder wenige Partner im privatem Umfeld
- Oft längere Pausen gefolgt von intensiveren Phasen
- Grosse Aufwand um Spiel zu vereinbaren
- Fehlende Partner bei gewollter Verbesserung im Spiel
- Fehlende Vergleichsmöglichkeiten, da wenige Partner und meist keine Protokollierung der Ergebnisse
- Wenig Anreize um nächstes Spiel zu organisieren

### 2.2.1.2 Use Cases

Gelegenheitsspieler können ausserordentlich von der geplante Applikation profitieren.

Neue **Partner** in der Nähe können unkompliziert mit der Applikation identifiziert werden. So ist es möglich längere Pausen, wenn der originale Partner in den Ferien ist oder schlicht verhindert ist, zu verhindern.

**Courts und Spielzeiten** können schnell vereinbart werden. Dies verringert den Aufwand um ein Spiel durchzuführen und erhöht somit die Motivation öfters Squash zu spielen. Spiele können **protokolliert** werden und so kann eine Statistik erstellt werden, welche wiederum als Anreiz dienen kann um besser zu werden.

Durch ein **Liga Management** können zusätzliche Anreize, neue Partner sowie eine Regelmässigkeit gefunden werden. Möglicherweise sind ein Grossteil der Gelegenheitsspieler nicht an einer solchen Liga interessiert, jedoch gibt es ein gewisses Potenzial dafür.

## 2.2.2 Regelmässige Spieler

### 2.2.2.1 Analyse

Regelmässige Spieler spielen regelmässig zu einem vereinbarten Termin mit einem oder mehreren Partner einen Racketsport. Meist werden genannte vereinbarte Termine mit den gleichen Partnern ausgetragen. Bei den vereinbarten Terminen gibt es gewisse Variationen. So



kann der Zeitpunkt und auch der Wochentag variieren.

Zusätzlich zu den regelmässigen Terminen, kommen meist noch unregelmässige Termine in der Charakteristik der Gelegenheitsspieler hinzu.

Durch die Regelmässigkeit ist diese Spielergruppe meist einiges ambitionierter als Gelegenheitsspieler. Dadurch das Sie regelmässig mit den gleichen Spielern spielen, wollen Sie besser sein als die anderen. Da Sie jedoch nicht in einer Liga spielen gibt es meist kein Protokoll der Resultate oder ein Ranking.

#### 2.2.2.2 Use Cases

Regelmässige Spieler können durch die geplante Applikation sehr unkompliziert ihre Spiele **protokollieren**.

Eine **Liga** kann zwischen mehreren regelmässigen Spielern arrangiert werden. Diese kann durch regelmässige **Terminvereinbarungen vom System** dazu beitragen das es keine Pausen gibt bei Krankheit oder Abwesenheit. Spieler die ausserhalb der eigenen Stammgruppe spielen wollen, können gleichzeitig bei **öffentlichen Ligen** weitere Erfahrungen sammeln.

### 2.2.3 Clubmitglieder

#### 2.2.3.1 Analyse

Clubmitglieder Spielen regelmässig und haben dafür fix Vereinbarte Clubtrainings. Diese sind nicht flexibel. Zusätzlich besteht auch eine Infrastruktur für Ranking über die Liga sowie eine Protokollierung von Spielen in den meisten Fällen. Clubmitglieder sind somit nicht im Hauptfokus dieser Applikation. Höchstens Ausserhalb des Clublebens ist es gut möglich das ein Clubmitglied diese Applikation braucht um neue Spiele zu vereinbaren oder in einer privaten Liga mitzuspielen.

### 2.2.3.2 Use Cases

Neue Clubs ohne Infrastruktur könnten diese Applikation für ihre Administration verwenden.

## 2.3 Use Cases

### 2.3.1 UC1: Identifizierung von Partnern

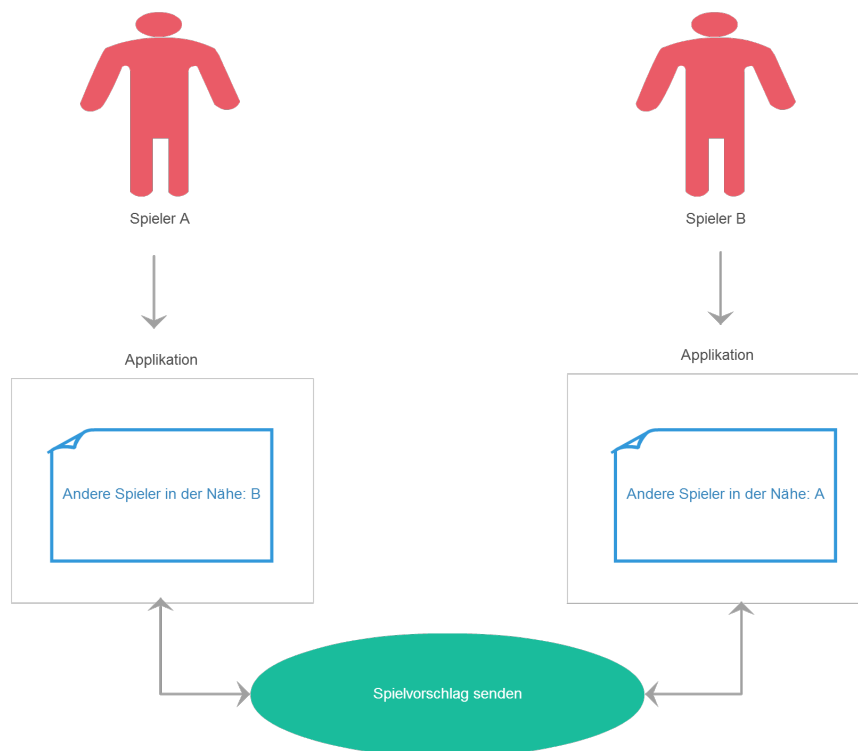


Abbildung 2.1: Grobplanung für Applikation

Der erste Use Case beschreibt die Identifizierung von Partnern. Zwei Partner - A und B - wollen einen bestimmten Racketsport in einer bestimmten Region spielen. Der User A oder B kann Spielvorschläge an andere User senden. Dafür kann er nach Region filtern.

## 2.3.2 UC2: Terminvereinbarung

Für den User Case Terminvereinbarung wurde in zwei Teil Use Cases unterteilt.

### 2.3.2.1 UC 2.1: Gelegenheitsspiele

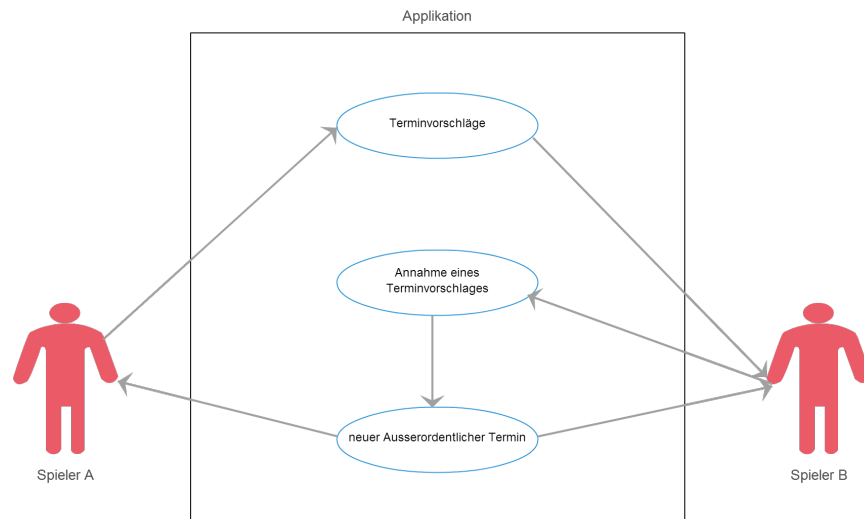


Abbildung 2.2: Grobplanung für Applikation

Bei Gelegenheitsspielen kann der Spieler A oder B einen oder mehrere Spielvorschläge senden. Anschliessend kann der Spieler B einer dieser Spielvorschläge **annehmen**, **ablehnen** oder **neue Spielvorschläge senden**. Sobald beide Nutzer einen Spielvorschlag angenommen hat, wurde der Termin vereinbart.

### 2.3.3 UC 3: Spiel Protokollieren

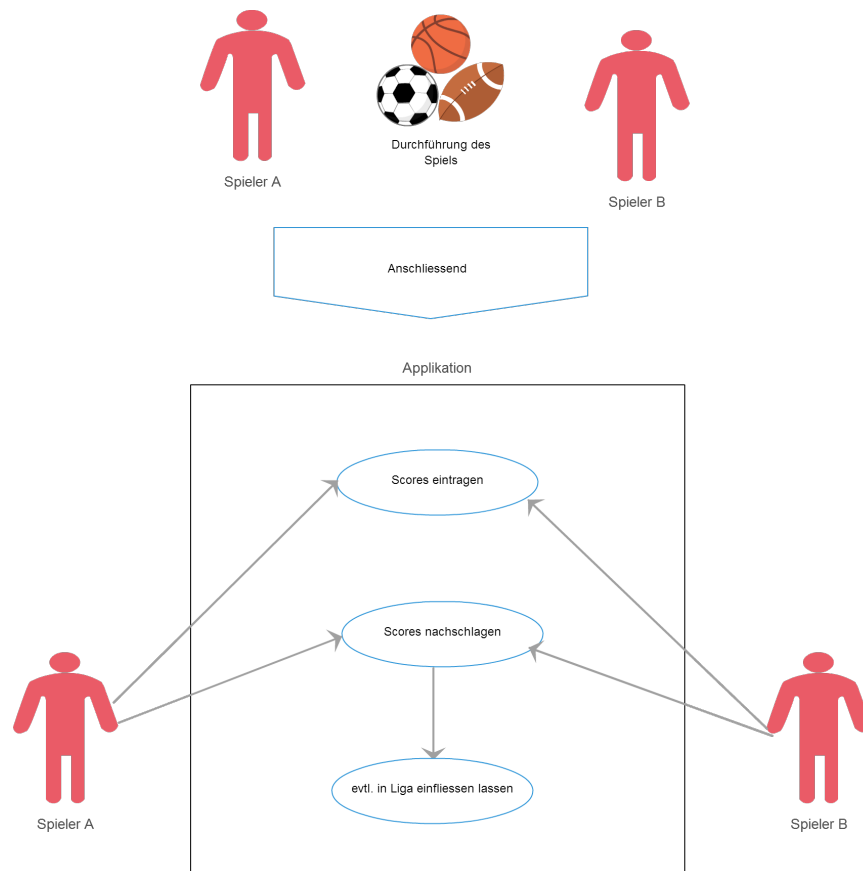


Abbildung 2.3: Grobplanung für Applikation

Nachdem ein Spiel durchgeführt wurde, kann der Spieler B sowie der Spieler A die Scores eintragen oder bestätigen. Anschliessend können beide Spieler die jeweiligen vergangenen Spiele jederzeit in der Applikation nachschlagen. Optional wird das protokollierte Spiel in einer Liga gewertet (falls vor dem Spiel so deklariert.)

## 2.3.4 UC4: Liga Management

### 2.3.4.1 UC 4.1: Liga erstellen

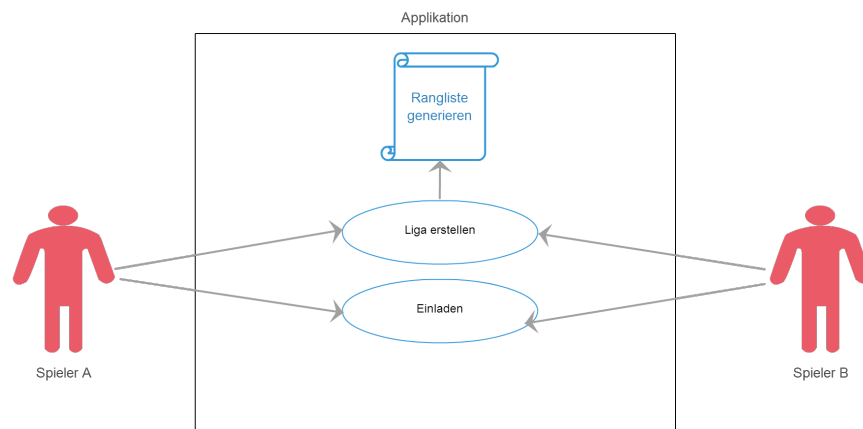


Abbildung 2.4: Grobplanung für Applikation

Jeder Spieler kann eine Liga erstellen. Diese Liga generiert eine Rangliste. Der Spieler kann anschliessend andere Spieler in die Liege einladen. Spieler können zusätzlich bei öffentlichen Ligen beitreten.

### 2.3.4.2 Rangliste durch Spiele aktualisieren

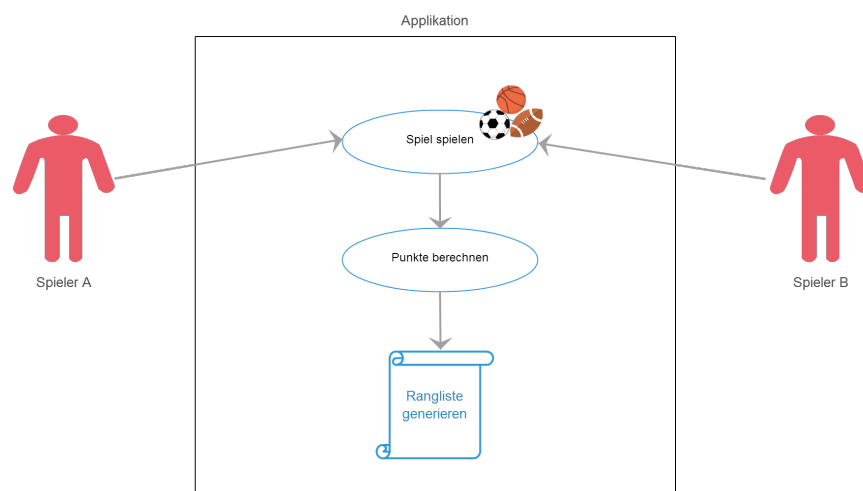


Abbildung 2.5: Grobplanung für Applikation

Nachdem ein Spiel gespielt ist, können beide Spieler ihre Scores eintragen. Stimmt diese überein oder bestätigt ein Spieler der eingetragene Score des anderen Spielers, wird dieser zur Rangliste hinzugerechnet.

### 2.3.4.3 Spiele automatisch anfordern

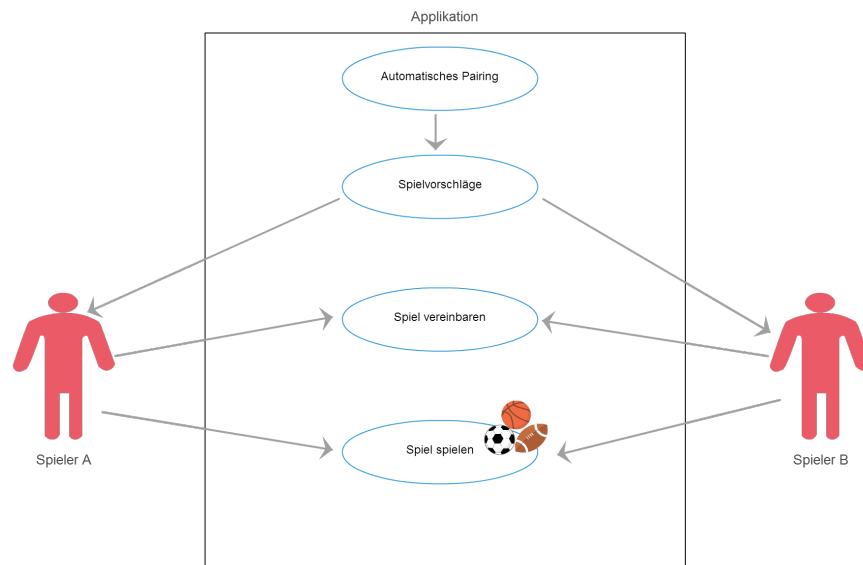


Abbildung 2.6: Grobplanung für Applikation

Bei der Erstellung der Liga kann einen Algorithmus zur automatischen Spielvereinbarung ausgewählt werden. Dieser Algorithmus schlägt jedem Spieler nach der definierten Regelmässigkeit Terminvorschläge vor und stellt so sicher, dass alle Spieler regelmässig gegen alle andere Spieler spielen. Die Spieler können sich auf einen der vorgeschlagenen Termine - oder ein anderer individuell vorgeschlagener Termin festlegen.

## 2.4 Anforderungsanalyse

Die Anforderung in diesem Abschnitt werden direkt in User Stories übertragen in das Tool Yodiz: <https://app.yodiz.com/plan/pages/task-board.vz?cid=14274&pid=1&iid=1>

### 2.4.1 Anforderungen für UC1

#### 2.4.1.1 REQ1.1.x - Zuteilung zu Regionen

Der User muss zu einer Region zugeordnet werden können. Regionen sind im Racketsport Zentren, die der Spieler vorzugsweise benutzt. Geographische nähen sind in diesem Use Case viel weniger präzise, da je nach Verkehrsanbindung oder Lebensumstände nicht Racketsport Zentren in der unmittelbaren Geografischen nähe referenziert werden.

**REQ1.1.1 - Racketsport Zentren registrieren:** Der User der Applikation kann Racketsport Zentren registrieren.

**REQ1.1.2 - Zuordnung zu Racketsport Zentren:** Der User kann sich zu Racketsport Zentren zuordnen und so anderen Spielern signalisieren, dass er da bevorzugt spielt.

**REQ1.1.3 - Suche nach Spielern per Racketsport Zentrum:** User können andere Spieler suchen die sich an ein Racketsport Zentrum zugeordnet haben.

#### **2.4.1.2 REQ1.2 - Broadcast Herausforderung für Region**

Der Spieler kann für eine Region einen Broadcast Spielvorschlag versenden für eine bestimmte Zeit.

#### **2.4.1.3 REQ1.3 - Privatsphäre Einstellungen für Herausforderungen**

Der Spieler kann in der Applikation einstellen ob er Broadcasts (REQ1.2) empfangen will und ob er von fremden Spielern Spielvorschläge empfangen will.

#### **2.4.1.4 REQ1.4 - Details zur Spielerselektion**

Der Spieler soll von anderen Spielern verschiedene Merkmale herauslesen können:

- Username
- Spielstärke
- Freiwillige Beschreibung
- Absichten (Regelmässige Spielpartner, Gelegenheitspartner, Spass spiele, Training, ...)
- Sportzeiten (z.b. nur abends, flexibel, nur Montags - Mittwoch)

#### **2.4.1.5 REQ1.5 - Friend System**

User können andere User zu ihrer Freundesliste hinzufügen. Privatsphäre Einstellungen können anschliessend drauf abgestimmt werden.

### **2.4.2 Anforderungen für UC2**

#### **2.4.2.1 REQ2.1 - Spiel erstellen**

Der User kann ein Spiel erstellen. Darin wird der Gegner, die Spielart sowie Terminvorschläge definiert.

#### **2.4.2.2 REQ2.2 - Spielterminvorschläge erstellen**

Der User kann mehrere Terminvorschläge für ein Spiel registrieren (Terminvorschlag besteht aus einer Lokation und einem Datum/Uhrzeit). Diese Terminvorschläge werden einem Spiel zugeordnet.

#### **2.4.2.3 REQ2.3 - Spielterminvorschläge annehmen**

Der Empfänger der Terminvorschläge kann einen Spielterminvorschlag annehmen.

#### **2.4.2.4 REQ2.4 - Spielterminvorschläge ablehnen**

Der Empfänger der Terminvorschläge kann alle Vorschläge ablehnen und neue Vorschläge erstellen.

#### **2.4.2.5 REQ2.5 - Regelmässige Spiele erstellen**

Der User kann ein regelmässiges Spiel mit einem Gegner registrieren. Beide User können Vorschläge zur Abweichung einreichen

### **2.4.3 Anforderungen für UC3**

#### **2.4.3.1 REQ3.1 - Ergebnisse des Spiels eintragen**

Der User kann das erzielte Ergebnis des Spiels eintragen. Informationen über Sätze sowie Punkte werden in der App für das jeweilige Spiel eingetragen.

#### **2.4.3.2 REQ3.2 - Bestätigung der Ergebnisse**

Der Gegner kann das eingetragene Ergebnis (REQ3.1) bestätigen oder korrigieren. Bei Korrektur wird das Ergebnis dem originalen User zur Bestätigung zurückgesendet.

### **2.4.4 Anforderungen für UC4**

#### **2.4.4.1 REQ4.1 - Liga erstellen**

Der User kann eine neue Liga erstellen. Die Liga kann öffentlich (jeder kann sich einschreiben) oder privat (Einschreibung nur per Einladung möglich) sein. Bei der Erstellung der Liga wird eine Rangliste erstellt. Die Rangliste beinhaltet alle User, welche der Liga zugeordnet sind sowie Punkte.

#### **2.4.4.2 REQ4.2 - User zur Liga einladen**

Der User kann einen oder mehrere andere User zur Liga einladen. Diese User können die Einladung anschliessend annehmen oder ablehnen.

#### **2.4.4.3 REQ4.3 - User kann Liga beitreten**

Der User kann eine Übersicht der Liga abrufen und einer gewünschten Liga beitreten.



#### **2.4.4.4 REQ4.4 - Spiele zu Liga zuordnen**

Der User kann ein Spiel- bevor gespielt wird - einer Liga zuordnen.

#### **2.4.4.5 REQ4.5 - Aktualisierung er Rangliste bei gespieltem Spiel**

Bei einem beendeten Spiel wird die Rangliste der Liga neu berechnet um die korrekte Punktezahl zu erhalten.

#### **2.4.4.6 REQ4.6 - Automatisch Spielaufforderungen**

In einem bestimmten Turnus werden alle Spieler aufgefordert ein Spiel mit einem anderen Spieler aus der Liga zu Spielen

#### **2.4.4.7 REQ4.7 - Abwesenheit**

Bei längeren Abwesenheiten kann einer Liga signalisiert werden, das der User aus der Automatischen Spielaufforderung herausgenommen werden soll.

## 3 Projektmanagement

### 3.1 Projektablauf

#### 3.1.1 Grobplanung

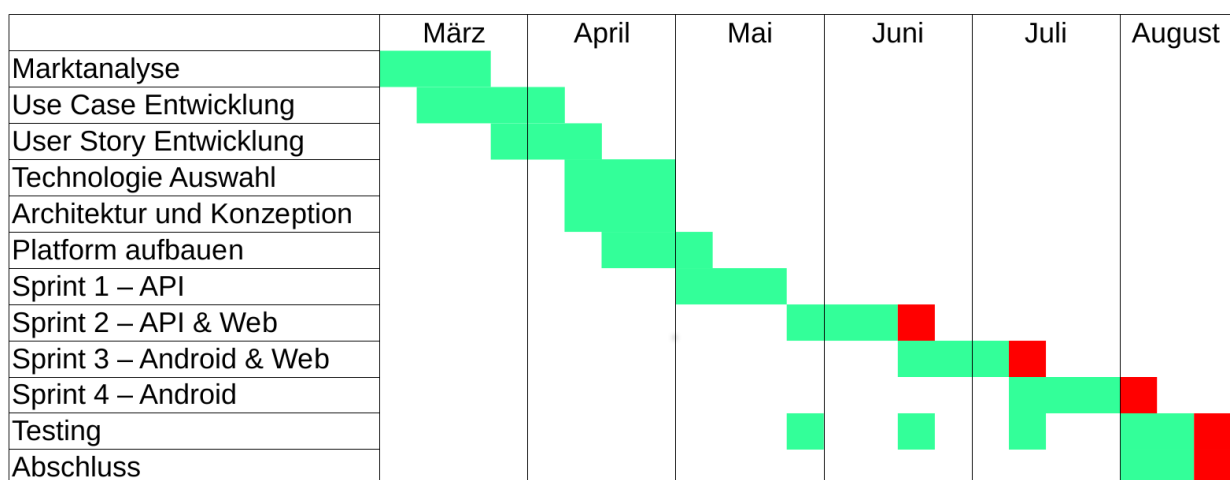


Abbildung 3.1: Grobplanung für Applikation

## 4 Konzeption

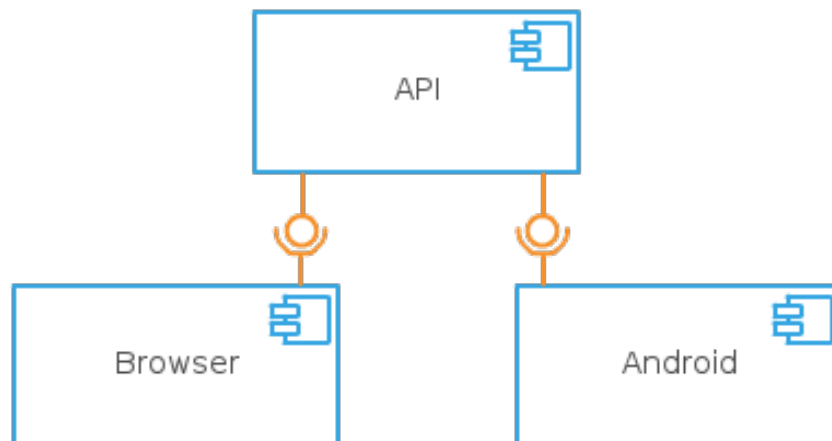


Abbildung 4.1: Grobkonzept für Applikation

Die Applikation besteht aus drei Teilen. Einem Webserver, der eine API und statische Clientfiles zur Verfügung stellt, der Client im Browser, welcher die API konsumiert sowie eine Android Applikation die auch auf die API zugreift.

### 4.1 Technologiestack

Wie in dem Grobkonzept beschrieben, wird für die Applikation einen Technologiestack gebraucht, welcher einfach eine skalierbare API sowie eine gute Integration der API mit einer Browser Frontend Anwendung bietet. Folgende Anforderungen werden an den Technologiestack gestellt:

- Skalierbare API
- Einfacher und schneller Umgang mit AJAX
- Gute Integration zwischen API und HTML/JS Client
- Responsive Design, integration mit OAUTH für fremdauthentisierungen
- Persistance Layer (Datenbankunterstützung)
- Integration mit Android APIs

---

## **4.2 Applikationsdesign**

### **4.3 DB Design**

### **4.4 GUI Entwurf**

## **5 Implementation**

### **5.1 REST API**

### **5.2 Web Applikation**

### **5.3 Android Applikation**

## **6 Test**

### **6.1 Unit Tests**

### **6.2 System Tests**

### **6.3 User Acceptance Tests**

## 7 Reflektion

# Abbildungsverzeichnis

|     |                                       |    |
|-----|---------------------------------------|----|
| 2.1 | Grobplanung für Applikation . . . . . | 10 |
| 2.2 | Grobplanung für Applikation . . . . . | 11 |
| 2.3 | Grobplanung für Applikation . . . . . | 12 |
| 2.4 | Grobplanung für Applikation . . . . . | 13 |
| 2.5 | Grobplanung für Applikation . . . . . | 14 |
| 2.6 | Grobplanung für Applikation . . . . . | 14 |
| 2.7 | Grobplanung für Applikation . . . . . | 15 |
| 3.1 | Grobplanung für Applikation . . . . . | 19 |
| 4.1 | Grobkonzept für Applikation . . . . . | 20 |