

Racketsports Manager

Semesterarbeit ZHAW



Autor: Raphael Marques

Betreuer: Michael Reiser

24. März 2015

Inhaltsverzeichnis

1	Inhalt der Arbeit	3
1.1	Ausgangslage	3
1.2	Aufgabenstellung	3
1.3	Erwartete Resultat	4
1.4	Zielsetzung der Arbeit	4
2	Analyse	5
2.1	Marktumfeld	5
2.2	Verhalten Spieler bei Terminfindung	5
2.3	Anforderungsanalyse	5
3	Projektmanagement	6
3.1	Projektablauf	6
3.2	User Stories	6
4	Konzeption	7
4.1	Technologiestack	7
4.2	Applikationsdesign	7
4.3	DB Design	7
4.4	GUI Entwurf	7
5	Implementation	8
5.1	REST API	8
5.2	Web Applikation	8
5.3	Android Applikation	8
6	Test	9
6.1	Unit Tests	9
6.2	System Tests	9
6.3	User Acceptance Tests	9
7	Reflektion	10

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
 - Eine Umfrage erstellen, wie sich Racket-Sportler für Spiele verabreden
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

2.1 Marktumfeld

2.2 Verhalten Spieler bei Terminfindung

2.3 Anforderungsanalyse

3 Projektmanagement

3.1 Projektablauf

3.2 User Stories

4 Konzeption

4.1 Technologiestack

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