

FitnesswithFriends

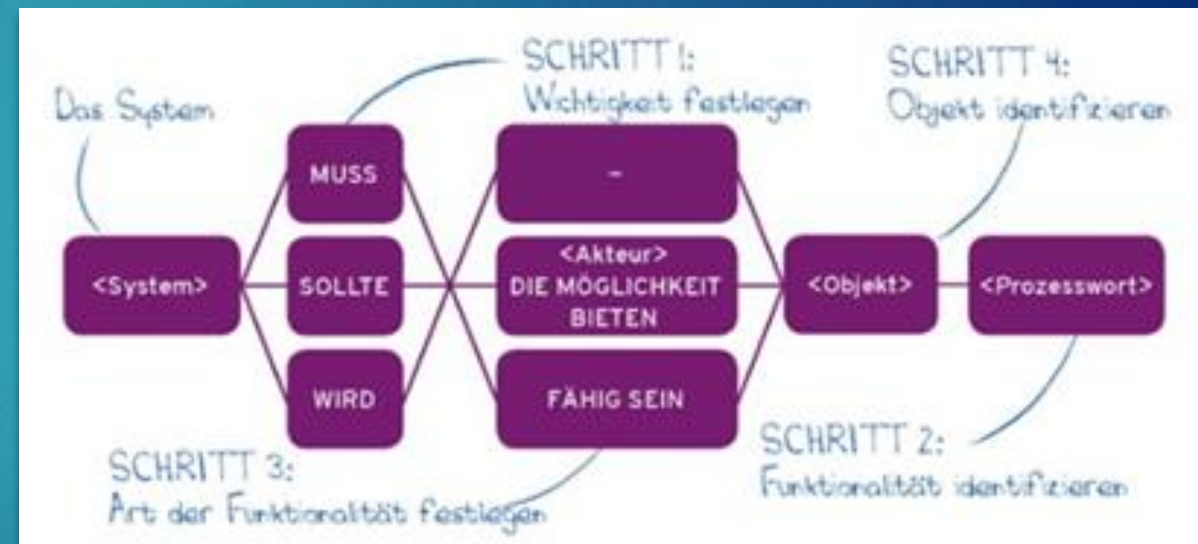


Inhaltsverzeichnis

- ▶ Anforderungen
- ▶ Alleinstellungsmerkmale
- ▶ Proof of Concept
- ▶ Risiken
- ▶ Evaluation
- ▶ User Interface Prototyp
- ▶ Verhaltensmodellierung
- ▶ Rest-tabelle
- ▶ Fazit
- ▶ Prozessassessment
- ▶ Architekturdiagramm

Anforderungen

- ▶ Hier ist für uns wichtig zu beachten, was der Unterschied zwischen Muss-Sollte-Wird ist.
- ▶ „Muss“ sind die Mindestanforderungen, die das System beinhalten muss.
- ▶ „Soll“ bedeutet, dass die Funktionen nicht zwingend notwendig sind und die man in Betracht ziehen könnte.
- ▶ „Wird“ stellt einen Ausblick dar.



Anforderungen

► Funktionale Anforderungen

1. Das System muss fehlerfrei den Standortermitteln können.
2. Das System muss dem Nutzer die Möglichkeit bieten Account zu erstellen.
3. Das System muss eine Liste von allen Nutzern in der Umgebung anzeigen, damit der Nutzer einen Überblick hat.
4. Das System muss den Nutzer die Möglichkeit bieten Herausforderungen an der persönlichen Leistung zu definieren.
5. Das System muss fehlerfrei die Statistik ausgeben können
6. Das System muss die Statistik nach einer Herausforderung fehlerfrei vergleichen können, um einen Gewinner ermitteln zu können.
7. Das System muss einen Gewinner ermitteln, damit die Punkte gutgeschrieben werden können
8. Das System wird den Nutzer die Möglichkeit bieten mit anderen Nutzer in Kontakt zu treten.
9. Das System wird die Möglichkeit bieten Trainingspartner in eine Freundesliste hinzuzufügen.
10. Das System wird den Nutzer die Möglichkeit bieten anderen Nutzern oder hinzugefügte Freunde herauszufordern.
11. Das System wird fehlerfrei bei der Sportlichen Aktivität die Werte ermitteln können.
12. Das System soll dem Nutzer die Möglichkeit bieten, verschiedenen Trainingspartner auszusuchen. Die Punkte müssen fehlerfrei auf das Konto des Benutzers hinzugefügt werden

Anforderungen

► Organisationale Anforderungen

1. Das System wird zu einem späteren Zeitpunkt in Android, IOS Implementiert.
2. Das System muss fehlerfrei sein, damit es nutzbar ist.

Anforderungen

► Qualitative Anforderungen.

1. Das System muss vertrauenswürdig mit Umgang der personenbezogenen Informationen Umgehen.
2. Das System muss eine schnelle Ermittlung der gemessenen Daten vergleichen können, um den Gewinner zu ermitteln.
3. Das System muss die Werte alle speichern können.
4. Das System soll die Möglichkeit bieten gewünschte Informationen anzupassen.
5. Das System wird eine angemessene Gestaltung und einen fehlerfreien Abruf beinhalten.
6. Das System wird eine fehlerfreie GPS Ortung durchführen können.
7. Das System wird die Werte beim Sport Live messen können.

Anforderungen

► Technische Anforderungen

1. Das System muss eine Datenbank besitzen
2. Das System muss eine sichere Kommunikation zwischen den einzelnen Bestandteilen haben.
3. Das System muss eine sichere Kommunikation zwischen Dienstanutzer und Dienstgeber anbieten.
4. Das System soll anpassbar sein (modular aufgebaut sein), um den Anforderungen in der Zukunft gerecht zu werden.

Alleinstellungsmerkmal

- ▶ Das Alleinstellungsmerkmal des Systems „Fitness with Friends“ gegenüber der Konkurrenz ist, dass ich als „Fitness with Friends“ den Nutzer ein System bieten möchte, welches Benutzerfreundlich ist.
- ▶ Mit der Funktion Trainingspartner suchen, möchte ich unserer Zielgruppe dazu verleiten mehr Sport mit unterschiedlichen Nutzern in der Umgebung zu machen. Die Funktion soll beeinflussen, dass man sich mit neuen Leuten beim Sport messen kann um somit an seine eigenen Persönlichen Grenzen stoßen kann.
- ▶ Mit der Wettbewerbsfunktionalität möchte ich bezwecken, dass sich die Nutzer nicht nur jedes Mal zum Sport verabreden um neue Leute kennen zu lernen, sondern das soll sie dazu verleiten sich mit der Konkurrenz zu Messen um sich gegenseitig Herauszufordern um somit nach der Zeit sich immer selbst verbessern

Proof of Concept

- ▶ 1. Standortermittlung
 - ▶ Um nicht allein zu trainieren bietet mein System die Möglichkeit sich nach Trainingspartner in der Umgebung zu erkundigen.
Es muss getestet werden, ob der aktuelle Standort des Endgerätes abgefragt werden kann.
- ▶ Exit Kriterien (Erfolg):
 - ▶ Aktueller Standort wird angezeigt
 - ▶ Trainings Partnern in der Umgebung werden übermittelt
- ▶ Fail Kriterien:
 - ▶ Koordinaten können nicht übermittelt werden.
- ▶ Fallback:
 - ▶ Wenn keine Standortbestimmung möglich ist, wird der Nutzer nach seinem Standort gefragt.

Proof of Concept

- ▶ 2.Partner auswählen
 - ▶ Wenn in der Umgebung des Nutzers weitere Nutzer ermittelt werden, hat man dann die Auswahl sich mit den Nutzern in der Umgebung in Kontakt zu setzen um gemeinsam trainieren gehen zu können.
- ▶ Exit Kriterien (Erfolg):
 - ▶ Nutzer in der Umgebung vorhanden
 - ▶ Trainingspartner gefunden
 - ▶ Kommunikation findet Statt
- ▶ Fail Kriterien: (Misserfolg):
 - ▶ Keine Nutzer in der Umgebung vorhanden
 - ▶ Trainingspartner nicht gefunden
 - ▶ Keine Kommunikation möglich
- ▶ Fallback(Alternative)
 - ▶ Der Nutzer kann allein Trainieren gehen und trotzdem bestehende Freunde herausfordern.

Proof of Concept

- ▶ 3. Herausforderung Definieren
 - ▶ Bevor man anfängt jemanden herauszufordern muss man als Nutzer die Herausforderung erstellen und definieren.
Man wählt die Sportart (Joggen) aus, setzt ein Maßstab und den Zeitraum, in dem man den Maßstab erreichen möchte, dann kann es schon losgehen.
- ▶ Exit Kriterien (Erfolg):
 - ▶ Sportart (Joggen) auswählen
 - ▶ Maßstab (Km) setzen
 - ▶ Zeitraum definieren
 - ▶ Ergebnis speichern
- ▶ Fail Kriterien (Misserfolg):
 - ▶ Server stürze bei berechnungen ab
 - ▶ Ergebnis konnte nicht gespeichert werden
- ▶ Fallback (Alternative):
 - ▶ Herausforderung neu starten nachdem server neugestartet wurde.

Proof of Concept

▶ 4. Statistik Messen

- ▶ Für den Prototypen werden die Werte der Aktivitäten Simuliert im späteren Verlauf bei fortführen des Projektes werden die Statistiken Live gemessen um die genauen Zahlen der Leistungen zu speichern.
- ▶ Exit Kriterien (Erfolg):
 - ▶ Statistik wird Simuliert
 - ▶ Statistik kann verglichen werden
- ▶ Fail Kriterien (Misserfolg):
 - ▶ Statistik kann nicht Simuliert werden.
 - ▶ Statistik kann nicht verglichen werden.
- ▶ Fallback (Alternative):
 - herausforderung neustarten

Proof of Concept

- ▶ 6.0 Gewinner ermitteln
 - ▶ Wenn die Nutzer die Herausforderung abgeschlossen haben, wird dann die Statistik der beiden verglichen und der Gewinner wird ermittelt.
 - ▶ Wenn der Gewinner ermittelt wurde ist werden die Punkte auf das Gewinner Konto gut geschrieben.
- ▶ Exit Kriterien (Erfolg):
 - ▶ Gewinner wird ermittelt.
 - ▶ Punkte werden gutgeschrieben
 - ▶ Statistik von beiden werden ausgegeben für den Vergleich
- ▶ Fail Kriterien (Misserfolg):
 - ▶ Gewinner konnte nicht eindeutig ermittelt werden
- ▶ Fallback (Alternative):
 - ▶ Beide Teilnehmer der Herausforderung werden als Gewinner angegeben und bekommen die Punkte gut geschrieben

Proof of Concept

- ▶ 7.0 Android Programmierung
 - ▶ Umsetzung der Alleinstellungsmerkmale während des Zeitraums des Projektes bestimmen.
- ▶ Exit Kriterien (Erfolg):
 - ▶ Einfache Funktionen möglich.
 - ▶ Gestaltung möglich.
- ▶ Fail Kriterien (Misserfolg):
 - ▶ Anwendungslogik in Android umzusetzen wird problematisch.
 - ▶ Zu wenig Kenntnisse und zu wenig Zeit für mich für die Umsetzung
- ▶ Fallback (Alternative):
 - ▶ Als Alternative kann man den Prototypen in Javascript Veranschaulichen.
 - ▶ Der Vorteil wäre, dass diese auf Eigenschaften des jeweils benutzten Endgeräts, vor allem Smartphones und Tabletcomputer, reagieren können.

Risiken

► Konzeptionelle Risiken

- Für das Nutzen des Systems ist es zunächst wichtig, dass sich ein Nutzer dazu entscheidet sich über das System zu Registrieren um sich sein eigenes Profil anzulegen.
- Somit besteht die Möglichkeit mit anderen Nutzern zu Kommunizieren und sich zum Sport verabreden.
- Aus der Marktrecherche kann man anhand der Konkurrenten entnehmen, dass eventuell Nutzer schon eine ähnliches System Nutzen und nicht ein weiteres benutzen möchten.

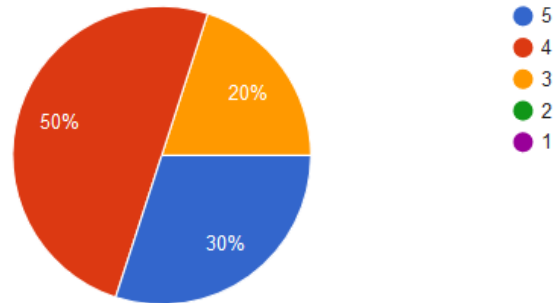
► Technische Risiken

- Ein Technisches Risiko ist, wenn die Funktion Standort ermitteln nicht funktionieren sollte wie gewohnt oder Probleme aufweist, weil somit der Nutzer nicht die Möglichkeit besitzt sich einen Trainingspartner in seiner Umgebung zu suchen.
- Somit könnte der Nutzer mit der Zeit das Interesse an das System verlieren, wenn es zu häufig auftreten sollte.
- Ein weiteres Risiko wäre, das im Team derzeit kein Erfahrener Android Entwickler vorhanden ist und im Projektzeitraum das Umsetzen eines Android Prototypen nicht machbar wäre.
Bei weiterführen des Projektes außerhalb der Zeitspanne würde man sich erfahrene Entwickler suchen, um den vollen Umfang der App zu erfüllen.
- Daher möchte ich den Prototypen als Webanwendung umsetzen, um zu veranschaulichen, dass es umsetzbar ist. Das System wird Response gestaltet, da es den Vorteil beinhaltet, dass es auf allen Endgeräten sich anpasst.
- Es ist sinnvoll aus der Sicht des Systems, dass als erstes die Trainingspartner Suchfunktion implementiert wird, um daraufhin die Wettbewerbsfunktionalität darauf aufzubauen, da sonst das Risiko bestehen könnte, das im System Fehler entstehen können.

Evaluation

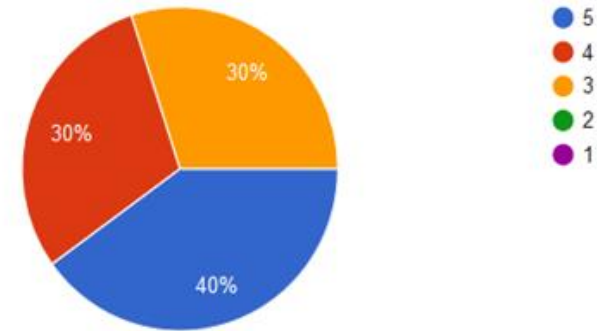
Wie ist Ihr Eindruck von dem jetzigen Prototypen?

10 Antworten



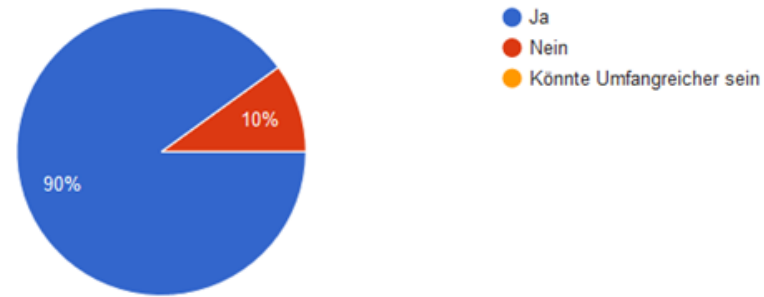
Wie ansprechend ist das Design im Bezug auf das Thema "Sport"?

10 Antworten



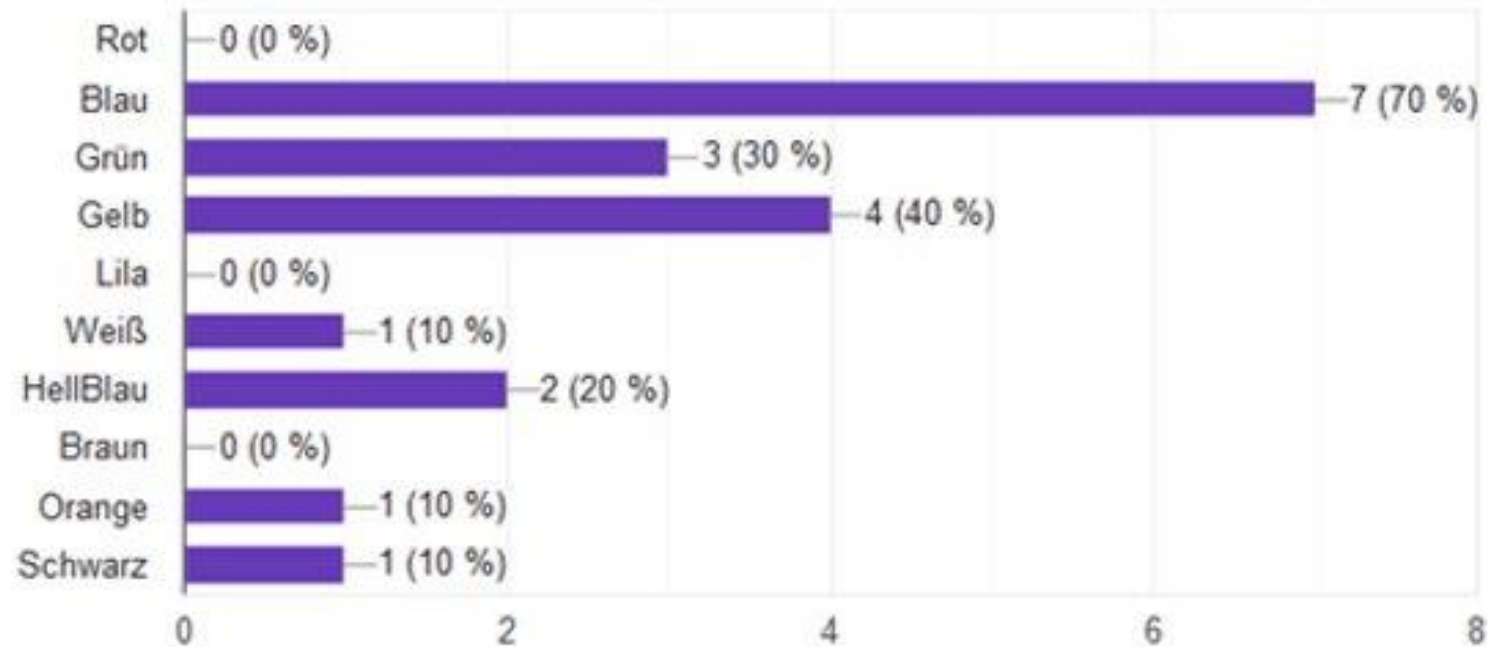
Ist Ihnen die Auswahl der Sportarten genug?

10 Antworten

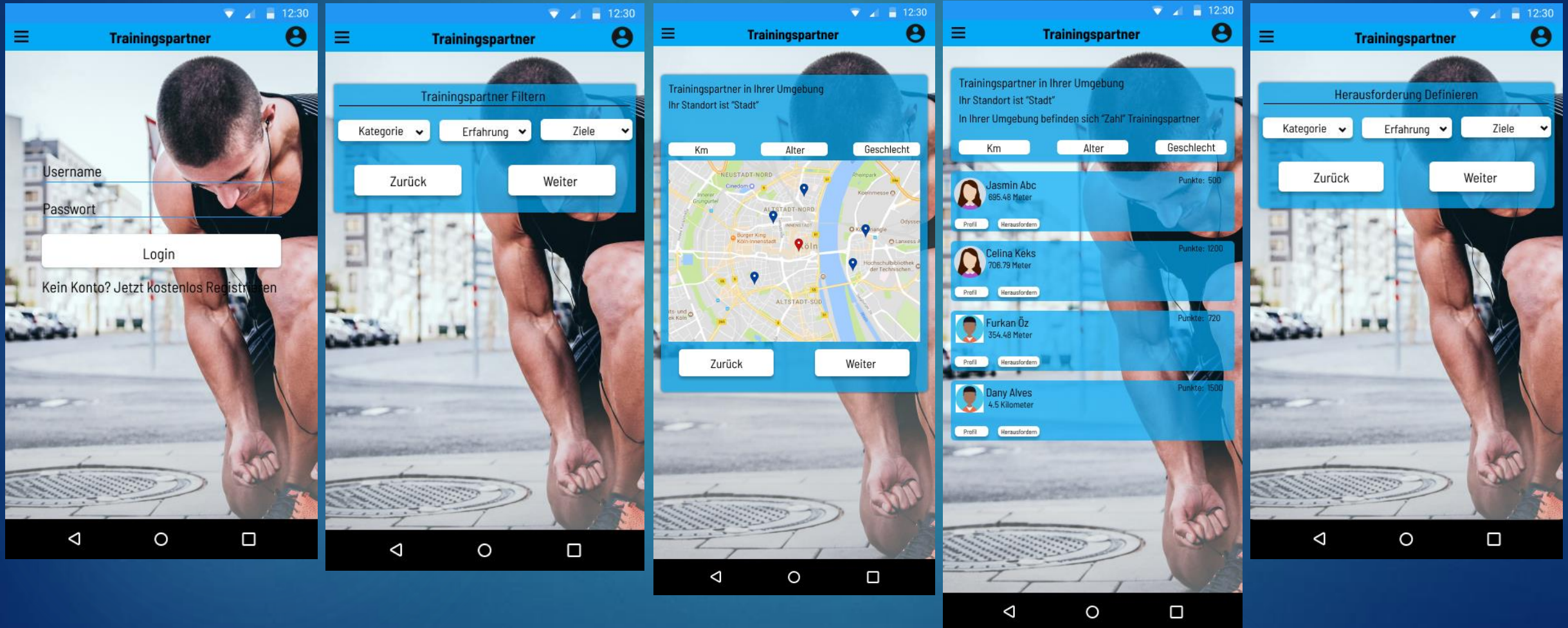


Welche Farbe Assoziieren sie mit der Thematik Sport?

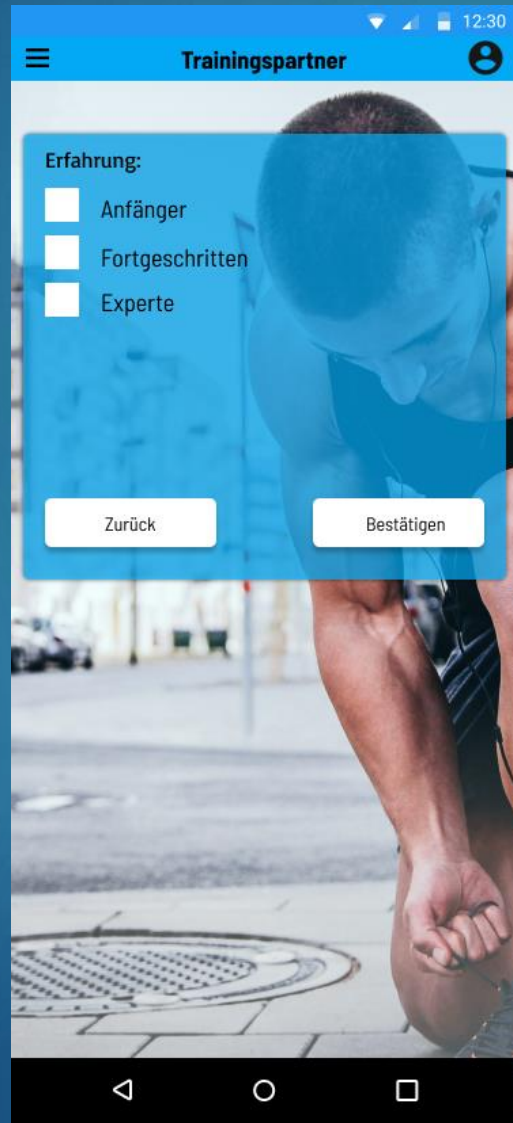
10 Antworten



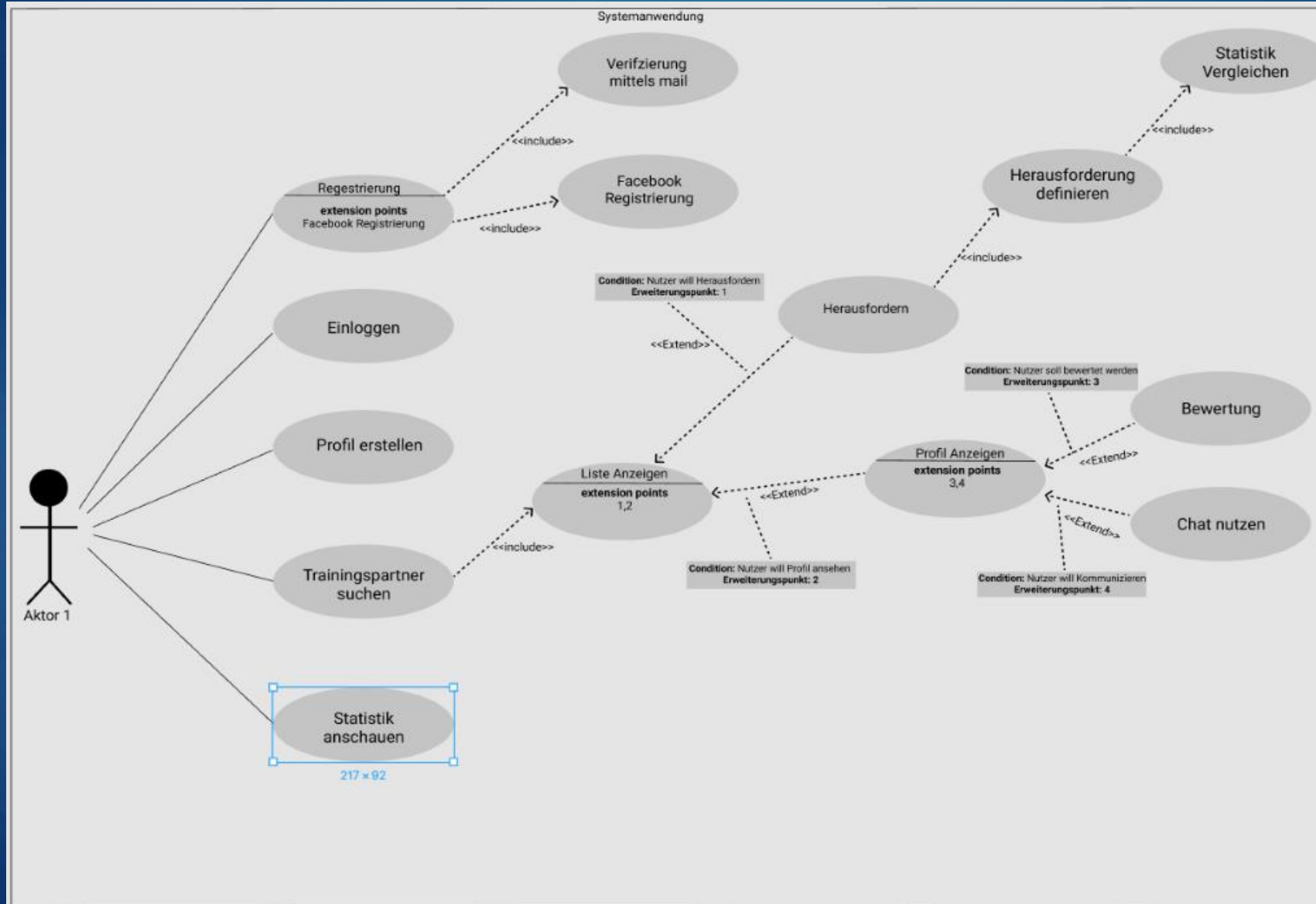
User interface Prototype



User interface Prototype



Verhaltensmodellierung



Rest-tabelle

Ressource	Methode	Semantik	Req	Res	Statuscodes
/compete	POST	Speichert alle Statistiken der User in die Datenbank, sobald ein User eine neue Statistik abspeichern will.	JSON	JSON	200: Erfolg 500: Serverfehler
/compete/points	POST	Speichert die vom Client errechneten Punkte in das jeweilige Nutzerprofil in der Datenbank	JSON	STRING	200: Erfolg 500: Serverfehler
/find/:name	GET	Sucht den aktuellen Benutzer aus der Datenbank und basierend auf den zuletzt gespeicherten Koordinaten Nutzer, die sich in seiner Nähe befinden (bis zu 5Km)	STRING		200: Erfolg 500: Serverfehler
/find/user/:id	GET	Gibt die zuletzt berechnete Statistik eines Nutzers zurück	STRING	JSON	200: Erfolg 500: Serverfehler

<code>/:user/:challenged</code>	GET	Rendert die HTML Seite um eine Herausforderung zu Definieren	STRING	HTML	200: Erfolg
<code>/compete</code>	POST	Berechnet Gewinner und Verlierer der Herausforderung basierend auf den zuvor vom Server abgefragten Statistiken. Wenn beide die Herausforderung erfolgreich abgeschlossen haben, gibt es keinen Verliere und Beide werden als Gewinner beachtet	JSON	STRING	200: Erfolg 500: Serverfehler
<code>/compete/display? {QUERYSTRING}</code>	GET	Fragt die Statistiken vom Server ab und rendert eine HTML seite für den Gewinner und Verliere samt der Statistiken die angezeigt werden.	STRING:	HTML:	200: Erfolg 500: Serverfehler
<code>/find</code>	GET	Rendert eine HTML Seite mit dem aktuellen Benutzer und Trainingspartner in der Umgebung	JSON	HTML	200: Erfolg 500: Serverfehler
<code>/</code>	GET	Rendert eine temporäre Login Seite ohne Passwort Eingabe, um dem System zu signalisieren, welcher User grade das System nutzt	-	-	-

Fazit

► Was wurde erreicht?

- Durch das Eis-Projekt habe ich sehr viel an Erfahrung gewonnen. Besonders in der Planung und der Arbeitseinstellung für die Entwicklung eines Systems konnte ich noch einiges lernen, sowie Ideen entwickeln und diese anschließend umsetzen.
- Durch die jeweiligen Meilensteine habe ich mich durch die Kriterien zu Verbesserung Vorschläge immer weiter gesteigert womit ich einen laufenden Prototyp erstellen konnte.

Fazit

- ▶ Was waren die Schwierigkeiten?
 - ▶ Anfangs war es schwer eine Struktur aufzubauen, wie man sich am besten durcharbeitet.
 - ▶ Nach einer Einarbeitungszeit wurde es aber verständlicher und es war für mich kein Problem mehr.
 - ▶ Eine weitere Schwierigkeit war, dass die Umsetzung des Projektes für mich eine schwere Wahl war, da ich zuerst den Prototypen in Android vorstellen wollte.
 - ▶ Durch zu wenig Kenntnisse und den Zeitmangel, ergab sich das Risiko, dass das Projekt nicht zum Abgabetermin fertig werden würde. Deshalb entschied ich mich dazu meine Wba2 Kenntnisse anzuwenden und ein System und die Idee dahinter in JavaScript als Prototypen darzustellen.
 - ▶ Da das System im finalen zustand für Android Windows und IOS angeboten wird ist dies dementsprechend vorteilhaft den Prototypen als eine Webanwendung zu Präsentieren

Fazit

- ▶ Was wurde verfehlt?
 - ▶ Die Android App wurde verfehlt, weil mir die Kenntnisse in Android fehlen für das Projekt.
 - ▶ Ich habe in der Zeit mir versucht es so gut wie möglich Kenntnisse anzueignen, da es aber durch Zeitmangel nicht funktioniert hätte wäre das Projekt nicht rechtzeitig fertig geworden.

Prozessassessment

► Reflexion des Projektes

- In meinem Arbeitsprozess konnte ich die Aufgaben in Meilensteinen zu Teilaufgaben aufteilen, um sie dementsprechend zu bearbeiten. Ich habe dabei ein iteratives Vorgehen verwendet, um die einzelnen Ergebnisse ständig zu überprüfen und zu optimieren.
- Das Feedback, welches ich im 1. Meilenstein erhielt, habe ich für den 2. Meilenstein genutzt und umgesetzt, indem ich durch das iterative Vorgehen meine Artefakte immer auf den neusten Stand gebracht habe.
- Zusätzlich konnte ich durch dieses Vorgehen einen laufenden Prototyp erstellen der meine Alleinstellungsmerkmale widerspiegelt und somit zur Veranschaulichung meiner bisherigen Arbeit dient.
- Für den 3. Meilenstein habe ich den Fokus auf die Implementierung des Vertikalen Prototypen gelegt. Zu Beginn war die Umsetzung auf Android geplant, aufgrund meiner nicht ausreichenden Erfahrungen in der Android Programmierung, habe ich die Programmiersprache JavaScript verwendet.

Prozessassessment

► Einhaltung des Projektplans

- Den Projektplan konnte ich in den ersten beiden Meilensteinen weitestgehend einhalten.
Wegen der in Meilenstein 1 und 2 abgegebenen Artefakte, die aus meiner Sicht zu theoretisch im Modul MCI waren, traten Probleme bei der Implementierung für den 3. Meilenstein auf.
- Da ich mich im 2. Meilenstein zu sehr auf theoretische Grundlage im Modul MCI fokussiert habe, sind mir in der Implementierung der Funktionalitäten Herausforderungen entgegengekommen. Auf Grund dessen, wurde deutlich mehr Aufwand für den 3. Meilenstein benötigt.
- Bei dem Meilenstein 1 gab es wenig Schwierigkeiten, die für mich als Herausforderung galten. Für mich war es wichtig eine Struktur aufzubauen und viel zu Brainstormen, um herauszufinden, wie die Umsetzung der Artefakte am besten durchgeführt werden können. Diesbezüglich habe ich versucht dies so gut wie möglich umzusetzen, um den Anforderungen des Modules gerecht werden.

Prozessassessment

- ▶ Bei dem Meilenstein 2 war zunächst die größte Herausforderung die Kriterien des Meilenstein 1 einzuhalten und diese dementsprechend zu verbessern. Zusätzlich hatte ich Probleme mit der Entscheidungsfindung für die Vorgehensweise in der MCI Modellierung des Usability Engineering.
- ▶ Durch die fehlenden Lektüren in der Hochschulbibliothek, musste ich mir Wissen durch das Internet aneignen, welches sehr viel Zeit kostete. Bei Problemen in der Bearbeitung von Artefakten habe ich meiste Zeit mit Hilfe des Internets oder Kommilitonen gelöst.
- ▶ Im 3. Meilenstein war das Problem, dass ich mich am Anfang nicht entscheiden konnte ob ich den Prototypen in Android Implementieren oder als eine Web Anwendung mit JavaScript entwerfen sollte. Ich habe mich zuerst für Android entschieden und habe mir durch viele YouTube Tutorials versucht die Sprache anzueignen. Frühzeitig habe ich gemerkt, dass es für mich sehr zeitaufwendig ist und ich bis zur Abgabe nicht fertig geworden wäre.
- ▶ Dementsprechend habe ich mich dazu entschieden eine Web Anwendung zu implementieren, da ich noch die vorhandene Erfahrung aus WBA2 nutzen konnte.
- ▶ Bei Problemen oder Fragen habe ich mich oft an meine Kommilitonen, welche das Projekt auch machen gewendet, da man sich untereinander am besten austauschen kann, da man leider immer nur ein Beratungstermin pro Iteration wahrnehmen darf.

Prozessassessment

▶ Individuelle Reflexion

- ▶ Abschließend möchte ich meine Reflexion zu dem Projekt abgeben, was ich während der Bearbeitungsphasen des Projektes gelernt habe und in welcher Weise mein Interesse sich geändert hat.

▶ Projektstrukturierung

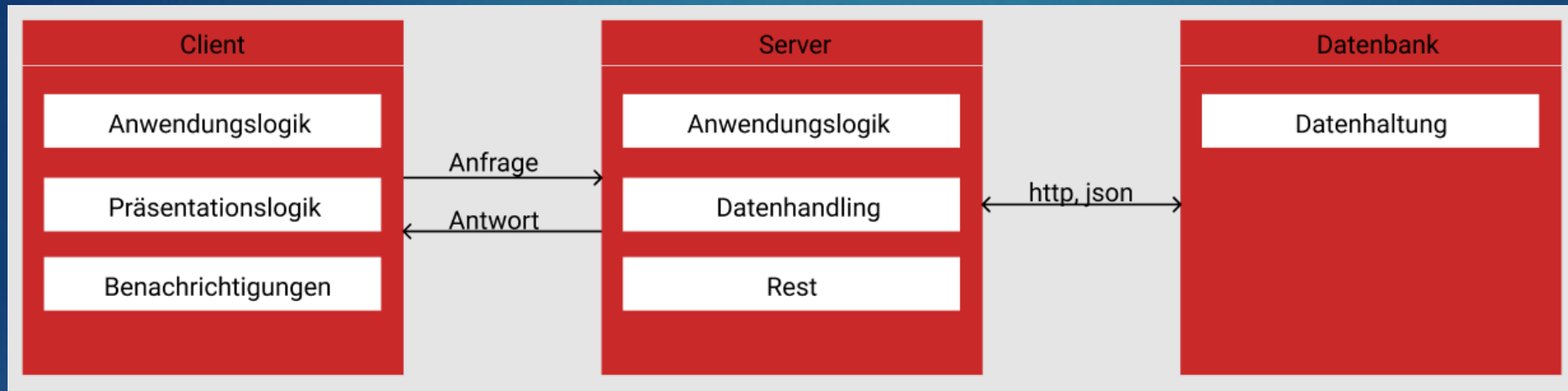
- ▶ Zunächst war es für mich kein Problem die Artefakte zu bearbeiten für Meilenstein 1 und 2, doch während der Bearbeitungsphase habe ich mit der Zeit gemerkt, dass man etwas Komplexer denken sollte um den Fokus auf wichtige Aufgaben zu legen um die bisherige Herangehensweise an ein Projekt zu ändern. Man sollte sich jedoch oft Gedanken machen und oft seine Entscheidungen durchdenken.
- ▶ Aus diesem Projekt konnten ich lernen, dass eine gut strukturierte Planung und Vorarbeit wichtig ist für so ein Projekt.

Prozessassessment

► Programmierung

- Bei der Programmierung hatte ich viele Probleme, da ich mich am Anfang nicht entscheiden konnte ob ich ein Prototyp in Android oder als eine Web Anwendung in JavaScript umsetzen sollte.
- Dies hat auch dementsprechend Zeit gekostet hat. Ich habe angefangen mir Android Tutorials anzuschauen um mir die Kenntnisse anzueignen. Als ich versucht habe die Funktion Wettbewerbsfunktionalität aus meinem Alleinstellungsmerkmal zu implementieren, habe ich gemerkt, dass ich es nicht zeitlich schaffen würde.
- Deshalb habe ich die Veranschaulichung des Prototypens in eine Webanwendung in JavaScript umgesetzt und dabei meine Kenntnisse von Wba2 verwendet.

Architekturdiagramm



► Anwendungslogik

- Client: Der Client vergleicht die generierten Statistiken und ermittelt den Gewinner.
- Server: Berechnet Trainingspartner in der Nähe und generiert die Statistiken.