

## **Moderationsnotizen**

### **Anforderungen:**

Meine Anforderungen aus dem 2. Meilenstein wurden durch das iterative Vorgehen etwas angepasst, um prüfen zu können, ob das System den Anforderungen gerecht geworden ist.

### **Alleinstellungsmerkmale**

Meine Alleinstellungsmerkmale sind spezielle Merkmale, die mein Produkt von ähnlichen Produkten auf dem Markt unterscheiden.

Unternehmen wollen durch ihre Alleinstellungsmerkmale, Kunden gewinnen und den Umsatz, sowie ihre Bekanntheit zu steigern.

Dies geschieht z.B. durch ein einzigartiges Produkt oder Dienstleistung.

### **Proof of Concept**

Das Proof of Concept ist ein Meilenstein, an dem die Durchführbarkeit eines Vorhabens belegt wird. In der Regel ist mit dem Proof of Concept meist die Entwicklung eines Prototyps verbunden, der die benötigte Kernfunktionalität aufweist.

### **Poc Android:**

Anhand des Poc zur Umsetzung eines Android Prototypen kann man entnehmen, dass es für mich aus zeitlichen Gründen und durch zu wenig Kenntnisse nicht möglich war, für den 3. Meilenstein einen funktionierenden Android Prototypen zu erstellen.

Aus diesem Grund habe ich mich dafür entschlossen, die noch vorhandene Zeit auszunutzen und einen Prototyp in JavaScript zur Veranschaulichung der Alleinstellungsmerkmale zu erstellen.

### **Poc IOS:**

Dieses Poc kam von Anfang an nicht in Frage, da ich genauso wie in Android bis dato unerfahren in Swift bin und dies zeitlich nicht möglich wäre, fertig zu werden für das Projekt.

### **Poc JavaScript:**

In JavaScript war es mir möglich, im gegebenen Zeitrahmen etwas sinnvolles basierend auf meinem Poc zu implementieren.

Hierbei wurde sich auf das Backend konzentriert, speziell das Finden von Trainingspartnern in der Umgebung sowie das Simulieren, Berechnen und Vergleichen der Statistiken.

Das Alleinstellungsmerkmal des Herausforderers wurde hierbei auf Client-Server verteilt:

1. Server
  - a. Simuliert Statistiken beider Nutzer (Herausforderer und Herausgeforderter). Außerdem werden die Statistiken in die jeweiligen Nutzerprofile der Datenbank abgesichert.
2. Client
  - a. Die vom Server errechneten Statistiken werden an den Client gesendet (nach entsprechendem Request). Hier werden diese in Bezug auf Distanz (aus zeitlichen Gründen konnte der Vergleich der Zeit nicht implementiert werden) verglichen und entsprechend auf einer HTML-Seite ausgegeben.

**Evaluation:**

Eine Evaluation ist eine systematische Untersuchung und Bewertung eines Design Layouts für einen Prototypen auf der Basis Erfahrens basierter Informationen.

Es soll dazu beitragen die Qualität und das Nutzen eines Produktes weiterzuentwickeln.

Nach Evaluation der Ergebnisse, kann man schon festlegen in welche Richtung es mit dem Design geht. Es werden Verbesserungen mit der Zeit vorgenommen die sich an dem System anpassen aber für den Prototypen genügt es bislang.

**User Interface Prototype:**

Ich habe den Prototypen mit Hilfe meiner Kenntnisse aus dem Fach Gdvk erstellt und habe dementsprechend das Programm Figma benutzt.

Für die Gestaltung der einzelnen Screens, habe ich mir zuvor Gedanken gemacht über die Designentscheidungen und habe dementsprechend eine Evaluation gemacht und dies dementsprechend nach dem 2 Audit angepasst.

Daraus ist der jetzige Prototyp entstanden.

Die Farbgebung hat sich dementsprechend auch durch die Evaluation entschieden, sodass ich mich für die Farbe Blau entschieden habe.

Bei Weiterführen des Projektes können durch Erweiterungen der Evaluierung neue Erkenntnisse gewonnen werden und diese, aus der Sicht des Nutzers übernommen werden können.

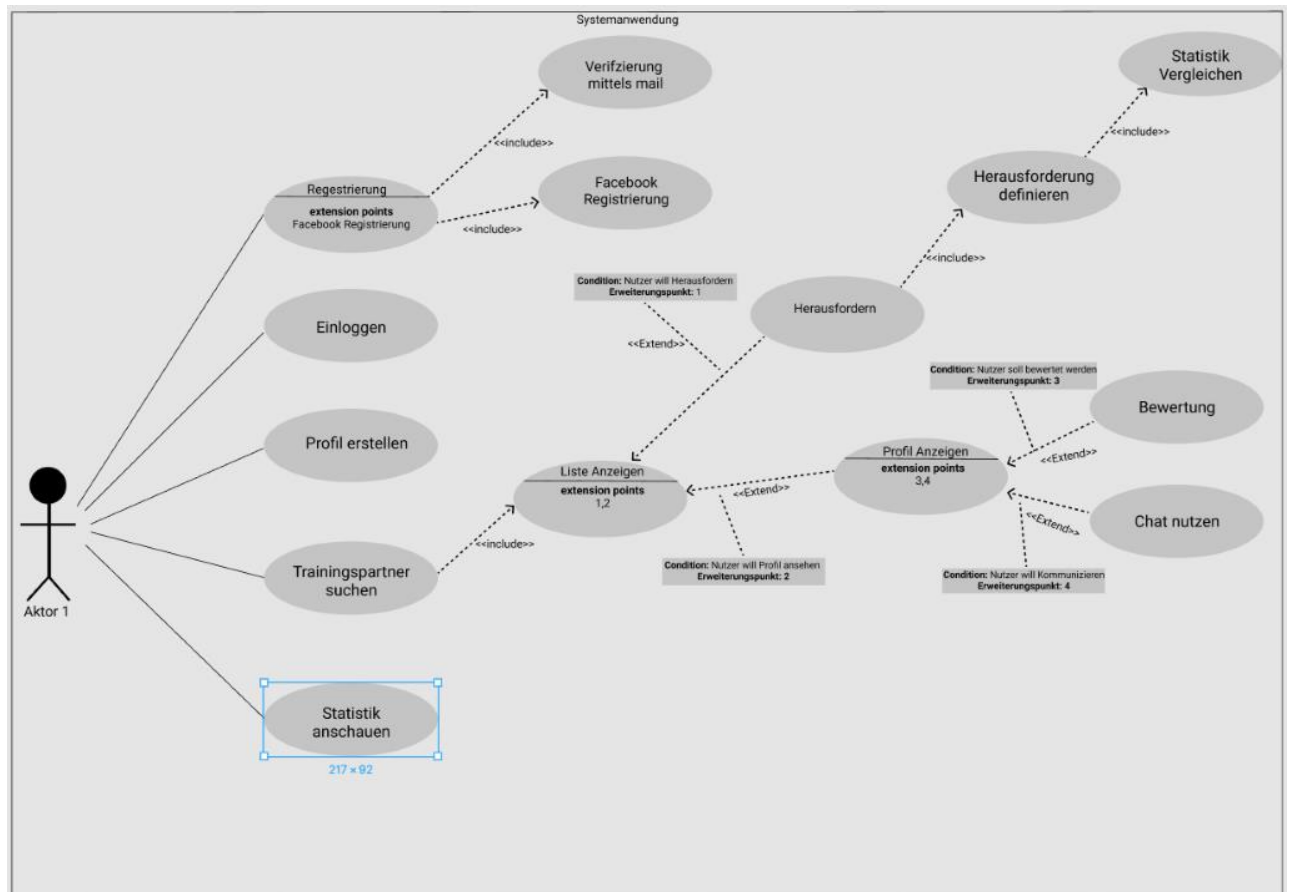
Für die Implementation des Designs fehlte es mir an Zeit, da ich mich überwiegend an die Implementierung gehalten hatte und versucht habe so gut wie es geht alles zu Implementieren.

### Verhaltensmodellierung:

Durch das Use Case Diagramm aus dem 1. Meilenstein, was ich iterativ verändert habe, ist es nun möglich, einen Überblick über die jeweiligen Anwendungsfälle zu erlangen.

Das Use Case Diagramm zeigt uns, welche Aufgaben ein System unterstützen soll für einen Benutzer; zusätzlich zeigt es uns, wie eine Benutzeroberfläche dargestellt ist.

Durch das Modul Softwaretechnik konnte ich mein gelerntes Wissen bezüglich der Anwendungsfalldiagramme nutzen und habe dementsprechend mein Use Case Diagramm iteriert.



## Architekturdiagramm

Bei meinem Architekturdiagramm habe ich auf eine Client-Server-Datenbank Architektur gesetzt die zu meinem Aktuellen Prototypen passt.

Hierbei war es mir wichtig das eine gewisse Eigenschaft gegeben ist, also das eine Lose Kopplung zwischen den einzelnen Elementen herrscht um in Zukunft einfacher und angenehmer weiterentwickeln zu können.



### Anwendungslogik

**Client:** Der Client vergleicht die errechneten Statistiken und ermittelt den Gewinner und errechnet die zu vergebenden Punkte.

**Server:** Ermittelt basierend auf Koordinaten der Nutzer in der DB Trainingspartner in der Nähe des Aktuellen Nutzers und errechnet die Statistiken und Speichert die neuen Punktezahl in das Benutzerprofil.

Vom Server zur Datenbank ist die Verbindung Asynchron.

Vom Client zum Server ist es Asynchron, wenn der Server zur Erfüllung der anfrage mit der Datenbank kommunizieren muss.

Und wenn es um Berechnungen geht ohne Anfragen zur Datenbank ist die Verbindung Synchron.