



ProFi

Projektfindungs-App

Projekt B | HAW Hamburg | WS 20/21

Tobias Braun 2365827, Nicolas Kolbeck 2365259, Melina Zanon 2364383

Inhalt

Einleitung	2
Projektziel	2
Teamaufteilung	2
ProFi Function	3
Vorbereitung: Fähigkeiten Werte berechnen	3
1 Stable Matching	3
2 Weitere Zuordnung	4
3 Ideen Löschen	4
3.a Ideen Löschen, die nicht zur Mindest-Teilnehmerzahl gekommen sind	4
3.b Ideen Löschen, deren Fähigkeiten nicht abgedeckt sind	5
4 Finale Zuordnung	5
4.a (optional) Leere Ideen hinzufügen	5
App	6
Allgemeiner Aufbau	6
Profil-Screen	7
Login-Screen	8
Home-Screen	8
Kurs-Screen	8
Ideen-Screen	9
Freie Projekte	9
Firebase Functions	9
Website	10
Entwicklung	10
Unterschiede zur App	11
Bekannte Fehler	11
Ausblick	11
ProFi Function	11
Datenbank	11
App	12
Website	12
Quellen	12

Einleitung

Dieses Projekt wird durchgeführt um das Modul Projekt B/Projekt 2 zu erlangen. Dabei bauen wir auf dem Projekt aus dem Kurs Mobile Systeme auf. Ziel des Projekts ist es eine funktionsfähige und überarbeitete Version der App zu entwickeln. Im Kursprojekt nicht verwirklichte Funktionen sollen eingebaut werden. Das Ergebnis wird in einer Präsentation vorgestellt.

Die App soll das Finden und Zuteilen von Interessenten in Teams unterstützen. Sie ist auf die Studiengänge Media Systems und Medientechnik der HAW Hamburg zugeschnitten. Als Grundlage für die Teamzuteilung dienen von den Usern erstellte Ideen. Passend zu den erforderlichen Fähigkeiten für die Umsetzung der Ideen werden die User anhand ihrer eigenen Fähigkeiten zugeteilt.

Projektziel

Die Teamfindungs-Funktion (ProFi Function) soll ausgebaut werden, sodass sie die bestmöglichen Ergebnisse erreicht. Hierfür sollen neben den Fähigkeiten auch die Interessen der User berücksichtigt werden. Ein ideales Team soll mit der vorher festgelegten Teilnehmerzahl möglichst alle erforderlichen Fähigkeiten abdecken und eine große Interessenüberschneidung haben.

An der Usability sollen kleine Verbesserungen vorgenommen werden wie z.B. eine erweiterte Kommentarfunktion, Lade-Animationen, Änderung von E-Mail/Passwort. Gegen die Umsetzung eines Kameramoduls haben wir uns schon am Anfang des Projektes entschieden.

Als Alternative zu Kursen soll es die Möglichkeit geben, freie Projekte zu erstellen, bei denen Ideen ohne Deadlines und Gruppengrößen ausgetauscht werden können.

Zusätzlich soll eine Web-Anwendung erstellt werden, die die Funktionalität der App im Browser zur Verfügung stellt.

Teamaufteilung

Tobias Braun	Website
Nicolas Kolbeck	Datenbank, App
Melina Zanon	ProFi Function

ProFi Function

Der Algorithmus soll Teams bilden, die möglichst gut die für die Umsetzung der Projektidee benötigten Fähigkeiten abdecken.

Um das zu erreichen müssen die Fähigkeiten gewichtet werden. Kommt eine Fähigkeit selten vor, ist es wichtig, dass die Teilnehmer, die sie besitzen priorisiert zu den Ideen, die sie benötigen zugeordnet werden. Dafür wird ein Punktesystem verwendet, das den Fähigkeiten unterschiedliche Werte zuschreibt.

Vorbereitung: Fähigkeiten Werte berechnen

Für jede Fähigkeit, die bei den Kursteilnehmern und in den geposteten Ideen vorkommt, wird ein Wert berechnet. Dieser Wert ist abhängig davon, wie oft die Fähigkeit vorkommt und wie oft sie benötigt wird.

Dafür werden alle Teilnehmer-Fähigkeiten und benötigten Fähigkeiten gezählt und die Anzahlen durcheinander geteilt. Man bekommt den Prozentsatz wie viele Teilnehmer, die die Fähigkeiten besitzen, gebraucht werden, um sie in allen Ideen abzudecken.

Bsp.: Haben 3 Teilnehmer eine Fähigkeit und sie wird in 3 Ideen benötigt, ist der Wert 100% bzw. 1:

- Genaue Abdeckung = 1
- Seltene Fähigkeit (häufiger benötigt als vorhanden) > 1
- Häufige Fähigkeit < 1

Dieser Wert wird dann mal 5 genommen und aufgerundet, damit die Werte ganzzahlige Punkte werden. Es entsteht folgende Werte Tabelle:

- 0%-20% = 1
- 21%-40% = 2
- 41%-60% = 3
- 61%-80% = 4
- 81%-100% = 5
- usw.

1 Stable Matching

Anhand der Wertetabelle wird für jeden Teilnehmer eine Rangliste erstellt. Dafür wird abgeglichen, in welchen Ideen er eine benötigte Fähigkeit besitzt und bekommt entsprechend Punkte. Für eine favorisierte Idee gibt es zusätzliche 5,5 Punkte. No-Gos werden nicht in die Rangliste aufgenommen.

Mithilfe dieser Ranglisten wird ein sogenannter Stable-Matching Algorithmus angewendet. Es wird eine Version des Gale-Shapley Algorithmus implementiert, bei der mehrere Leute zugeordnet werden können. Der Algorithmus liefert stabile Gruppen, in denen kein Teilnehmer im Kurs besser zu der Idee passt als die Teammitglieder.

Dabei werden die Leute anhand der Ranglisten bis zur Mindestgruppengröße zu den Ideen zugeordnet, die ganz oben in ihrer Rangliste sind. Hat eine Idee bereits die Mindestgruppengröße erreicht, wird verglichen, ob der Teilnehmer nicht doch besser passt als ein schon zugeordneter. Dafür wird eine neue Rangliste aus den Zugeordneten und dem Bewerber für die Idee erstellt. Der schlechteste in dieser Liste wird entsprechend wieder entfernt.

Der Algorithmus läuft solange, bis alle Kursteilnehmer zugeordnet wurden, oder am Ende ihrer Rangliste angekommen sind und somit nicht besser als die bereits zugeordneten Teilnehmer sind.

2 Weitere Zuordnung

Sind nach dem Stable-Matching noch Teilnehmer unsortiert, werden diese weiter zugeordnet.

Zunächst werden die Fähigkeitswerte basierend auf den Fähigkeiten der unsortierten Teilnehmer und den noch nicht abgedeckten, benötigten Fähigkeiten neu berechnet. Den Teilnehmern werden wieder Punkte für die Ideen zugeschrieben. In diesem Schritt zählt ein Favorit 0,5 Punkte. Außerdem bekommt der Teilnehmer auch Punkte für gemeinsame Interessen mit den schon zugeordneten Teammitgliedern.

Derjenige, der die meisten Punkte hat, wird der entsprechenden Idee zugeordnet, wenn diese noch nicht voll ist. Dann fängt die Schleife von vorne an und es werden wieder neue Fähigkeitswerte berechnet, bis es keine Übereinstimmung zwischen Fähigkeiten oder Interessen mehr gibt.

3 Ideen Löschen

3.a Ideen Löschen, die nicht zur Mindest-Teilnehmerzahl gekommen sind

In diesem Schritt sind die meisten Teilnehmer bereits zugeordnet und wenn eine Idee noch unter der Mindestgruppengröße liegt, wird diese höchstwahrscheinlich auch nicht mehr gefüllt werden können. Daher wird die Idee gelöscht. Gibt es mehrere unvollständige Ideen, werden diese nach Anzahl der fehlenden Fähigkeiten und dann nach Gruppenanzahl sortiert und in dieser Reihenfolge gelöscht. Die Idee mit den meisten fehlenden Fähigkeiten und den wenigsten Teammitgliedern wird als erstes gelöscht.

Bei jeder Löschung wird zunächst geprüft, ob es noch genug freie Plätze gibt, um die Teammitglieder der zu Löschenden Idee auf andere Ideen aufzuteilen.

Die Teilnehmer, die jetzt wieder unsortiert sind, werden wie in Schritt 2 neu zugeordnet.

3.b Ideen Löschen, deren Fähigkeiten nicht abgedeckt sind

Da das Ziel ist, möglichst alle benötigten Fähigkeiten abzudecken, werden vor der finalen Zuordnung noch einmal alle Ideen auf Ihre Abdeckung geprüft.

Fehlen zu viele Fähigkeiten und sind noch genug Plätze frei, werden diese Ideen gelöscht. Die mit den meisten fehlenden Fähigkeiten zuerst. Dabei wird mit einer Toleranzgrenze von 20% gerechnet. Fehlen also weniger als 20% der benötigten Fähigkeiten, zählt die Idee als abgedeckt und wird nicht gelöscht.

4 Finale Zuordnung

Wenn es immer noch unsortierte Teilnehmer gibt, werden diese ein letztes Mal in einer Rangliste sortiert und entsprechend auf nicht volle Ideen aufgeteilt. Dabei zählen alle passenden Fähigkeiten gleich viel und ein Favorit 5 Punkte. Auch gemeinsame Interessen werden hier zugezählt.

Die letzten Teilnehmer werden priorisiert auf kleinere Gruppen aufgeteilt, um die Gruppengrößen etwas auszugleichen.

In diesem Schritt werden auch Kursteilnehmer, die eine Punktzahl von 0 erreichen, auf eine Idee aufgeteilt.

Gibt es nicht genug Ideen, geht die Aufteilung nicht ganz auf. In diesem Fall wird nach der finalen Zuordnung geprüft wie viele Leute übrigbleiben. Sind es mehr als die Mindestgruppengröße oder es fehlen nur bis zu 25%, um die Mindestgruppengröße zu erreichen, werden neue leere Ideen angelegt. Andernfalls werden die übrigen Leute auf die bestehenden Ideen aufgeteilt, auch wenn die Teams dann über die maximale Gruppengröße gehen.

4.a (optional) Leere Ideen hinzufügen

Gibt es zu wenig Ideen, müssen Teams ohne Idee gebildet werden. Diese werden im Folgenden leere Ideen genannt. Das heißt es wird ein Team zusammengestellt, basierend auf gemeinsamen Interessen, und die Teammitglieder müssen noch eine Projektidee entwickeln.

Ein zufälliger unsortierter Kursteilnehmer wird als erstes Mitglied in eine leere Idee gepackt. Alle anderen bekommen Punkte, wenn sie ein gemeinsames Interesse mit dem ersten Teilnehmer haben. Diejenigen mit den höchsten Punktzahlen kommen in die Idee.

Ist ein Platz weniger als die maximale Gruppengröße erreicht, wird eine weitere leere Idee erstellt und wieder ein zufälliger Teilnehmer als erstes Mitglied gewählt. Dies wird gemacht, bis keiner mehr übrig ist.

Wenn in der letzten leeren Idee weniger Leute als die Mindestgruppengröße sind (mit einer Toleranzgrenze von 25%), wird diese wieder aufgelöst. Die Teilnehmer werden entsprechend auf die anderen leeren Ideen, in denen immer noch ein Platz übrig ist, aufgeteilt.

Gibt es nicht genug leere Ideen, werden die Teilnehmer in die nächste nicht leere Idee mit den meisten fehlenden Fähigkeiten sortiert, auch wenn sie dann über die maximale Gruppengröße geht. Ist die nächste Idee das No-Go des Teilnehmers wird sie übersprungen.

App

Allgemeiner Aufbau

Die App wurde mit React Native und Expo erstellt. Per Tab-Navigator am unteren Bildschirmrand erfolgt die Navigation zwischen Profil- und Kurs-Bereich. Die einzelnen Screens dieser Bereiche sind in Stack-Navigatoren angeordnet. Fast alle UI-Elemente sind selbst erstellte Komponenten, die an verschiedenen Stellen in der App verwendet werden, um ein einheitliches Aussehen zu erzeugen. Eine „ThemeManager“-Komponente stellt außerdem die Farben zur Verfügung und ändert diese, je nachdem ob sich das Smartphone im Light- oder Dark-Mode befindet. Text-Stile und Icon-Grafiken sind in einer „Style.js“ Datei hinterlegt. Die Interaktion mit der Firebase-Datenbank erfolgt in der Schnittstelle „DB_API.js“. Diese enthält alle benötigten Funktionen zum Schreiben und Abrufen von Informationen in (meist asynchronen) Funktionen und wird überall in der App angewandt. Einige Beispiele für solche Funktionen sind:

- Angelegen, Löschen und bearbeiten von Userdaten
- Suchen nach Kursen, an denen der User interessiert ist, und Zurückgeben dieser Kurse mit allen Informationen als aufbereitete Liste
- Erhalten aller Informationen über einen User per User-ID

Gibt der User die Erlaubnis dafür, wird ein Expo Push-Token für das Gerät in der Datenbank gespeichert, mit dem er Push-Mitteilungen erhalten kann.

Profil-Screen

Im Profil kann mithilfe des nativen Image-Pickers das Profilbild ausgewählt werden. Der Ausschnitt des Bildes lässt sich individuell festlegen. Dieses wird dann komprimiert und in die Datenbank hochgeladen. Hatte der User vorher ein anderes Profilbild, wird dieses aus der Datenbank gelöscht.

Username und Kurzbeschreibung können in je einem Modal geändert werden, beim Drücken des „Bestätigen“ Buttons werden die Änderungen in die Datenbank übernommen. Auch E-Mail-Adresse und Passwort lassen sich ändern, allerdings hier mit zusätzlicher Eingabe des Passworts, da Firebase diesen Schritt zur Absicherung erfordert.

In einem weiteren Modal kann der User verwalten, für welche Ereignisse er Push-Mitteilungen erhalten möchte. Dabei gibt es folgende Optionen:

- Einteilung von Ideen – Sobald ein Kurs oder eine freie Idee eingeteilt wurde, in der der User Mitglied war, bekommt er eine Mitteilung mit den entsprechenden Informationen
- Neue Kommentare – Wenn in einer vom User erstellten Idee ein Kommentar gepostet wird, oder auf einen Kommentar des Users geantwortet wird, erhält er diesen als Nachricht
- Änderung an meinen Kursen – Wenn der Ersteller eines Kurses, in dem der User Mitglied ist, Änderungen vornimmt, wie das Hinzufügen neuer Skills oder Änderung des Datums, wird der User über die Art der Änderung informiert
- Löschen von Ideen – Bevor eine alte Idee automatisch gelöscht wird (siehe Firebase Functions) wird der Ersteller benachrichtigt
- Neu verfügbare Fähigkeiten – falls durch die Admins eine neue Kategorie (oder neue Fähigkeit innerhalb einer bestehenden Kategorie) hinzugefügt wird, wird der User darüber informiert

Seine Fähigkeiten und Interessen (im Code zu „Attribute“ zusammengefasst) kann der User auf einem weiteren Screen auswählen. Beim ersten Aufrufen dieser Seiten wird er darüber informiert, welche Bedeutung die Auswahl hat. Ausgewählte Attribute werden sofort in der Datenbank aktualisiert.

Mit dem Abmelden-Button wird der User sofort ausgeloggt und das Push-Token wird aus der Datenbank entfernt, sodass dieses Gerät nicht mehr die Mitteilungen des Users erhält.

Die Funktion User löschen entfernt alle eingegebenen Daten sowie die Authentifizierungsmöglichkeit aus der Datenbank. Der User wird aus allen Kursen entfernt, die noch nicht eingeteilt sind. Bei bereits eingeteilten Kursen und Kommentaren wird er weiterhin als „gelöschter User“ angezeigt.

Login-Screen

Ist der User nicht eingeloggt, öffnet sich dieser Screen beim Starten der App. Durch Eingabe von E-Mail-Adresse und Passwort wird der User bei Firebase authentifiziert und zum Home-Screen weitergeleitet. Eine Funktion zum Zurücksetzen des Passwortes per E-Mail steht ebenfalls zur Verfügung. Das Passwort kann bei Bedarf ein- und ausgeblendet werden.

Ein neues Profil kann im Registrieren-Modal hinzugefügt werden. Username, E-Mail und Passwort werden eingegeben und bei Firebase gespeichert. Der User wird sofort eingeloggt.

Home-Screen

Dies ist der Start-Screen nach dem Einloggen. In einer Section-List werden alle Kurse nach Semestern geordnet dargestellt, bei denen der User zu den „prospects“ (Interessenten) gehört. Kurse, bei denen er auch „member“ ist, also bei der Einteilung berücksichtigt wird, sind mit einem Haken vor dem Kursnamen markiert. Bereits eingeteilte Kurse, in denen der User Mitglied ist, sind blau hinterlegt. Eingeteilte Kurse, bei denen der User kein Mitglied ist, sind ausgegraut. Wenn der User bestimmte Kurse aus der Ansicht entfernen möchte, wischt er nach rechts und benutzt den Austreten-Button. Dies entfernt ihn sowohl aus den „prospects“ als auch aus den „members“ des Kurses.

Über das Modal Kurs finden kann der User eine Kurs-ID eingeben, die in der Datenbank gesucht und zur Liste hinzugefügt wird.

Im Modal Neuer Kurs kann er einen neuen Kurs erstellen, sofern es diesen noch nicht gibt. Die eindeutige Kurs-ID wird aus dem vom User eingegebenen Kürzel und dem Semester des gewählten Datums automatisch erstellt. Während das Datum also nachträglich noch angepasst werden kann, ändert sich das Semester dann nicht mehr. Neben dem Kursnamen kann ein Emil-Link angegeben werden, der automatisch als solcher formatiert wird. Mithilfe des nativen Datums- und Uhrzeit-Pickers wird der Termin für die Gruppeneinteilung ausgewählt.

Kurs-Screen

Hierhin gelangt der User nach Auswahl eines bestimmten Kurses. Ist er der Ersteller des Kurses, hat er die Möglichkeit, den Kurs nachträglich zu bearbeiten. Oben werden alle Profilbilder der Kursmitglieder angezeigt. Beim Klick auf ein Profilbild öffnet sich die „ProfileView“ mit den Eigenschaften des entsprechenden Users. Bei nicht eingeteilten Kursen kann jeder User beitreten bzw. austreten, um bei der Team-Erstellung berücksichtigt bzw. nicht berücksichtigt zu werden. Jeder User kann Ideen hinzufügen. Neben einem kurzen Titel und einem Beschreibungstext muss er hierzu die benötigten Fähigkeiten aus einer Liste auswählen.

Ideen können durch Wischen nach rechts als Favorit oder No-Go markiert werden. Favoriten erhöhen die Chance, dass der User dieser Idee zugeteilt wird, während No-Gos die Einteilung in diese Idee verhindern. Pro Kurs kann ein Favorit und ein No-Go gesetzt werden.

Die Idee, in der der User eingeteilt wurde, wird blau hinterlegt. Ideen, die bei der Einteilung gelöscht wurden und danach kein Team haben, werden ausgegraut.

Ideen-Screen

Hier sieht der User die Eigenschaften der Idee inklusive der benötigten Fähigkeiten auf einem eigenen Screen. Bei eingeteilten Ideen wird dort auch die Abdeckung dieser Fähigkeiten durch kleine Profilbilder dargestellt sowie eine Liste des Teams angezeigt.

Bei automatisch erstellten Teams ohne Idee werden hier auch die gemeinsamen Interessen angezeigt.

Im Kommentarbereich kann jeder User Kommentare verfassen, liken und beantworten. Eigene Kommentare können nicht geliked, aber gelöscht werden. Neuere Kommentare werden weiter oben angezeigt, wobei Antworten auf andere Kommentare allerdings chronologisch von oben nach unten dargestellt werden.

Der Ersteller einer Idee wird unter dem Kommentarbereich angezeigt.

Freie Projekte

In dieser Kategorie finden sich Projekte ohne Deadline-Datum. Diese können nicht von Usern erstellt werden, sondern nur von Admins. User können wie bei regulären Kursen Ideen posten. Hier erfolgt die Einteilung allerdings nicht automatisch, sondern durch den Ideen-Ersteller manuell. Der User kann den Ideen beitreten, die er interessant findet. Die aktuelle Abdeckung der benötigten Fähigkeiten wird durch Profilbilder aller beigetretenen User dargestellt. Ist der Ersteller mit der Abdeckung zufrieden, kann er die Einteilung vornehmen. Die Idee wird mit allen aktuellen Mitgliedern „eingefroren“ und das Team steht fest.

Firebase Functions

Einige Funktionen der App können nicht auf dem Gerät des Users ausgeführt werden, da sie user-unabhängig sind. Hierfür dient die „index.js“ im Funktions-Ordner. Diese liegt auf dem Firebase-Server und enthält Funktionen, die teilweise zu festgelegten Zeitpunkten, teilweise bei Interaktion mit der Datenbank automatisch ausgeführt werden.

Zu den Firebase Functions gehören:

- Löschen von Kursen, die seit mehr als 180 Tagen eingeteilt sind
- Löschen von Ideen in freien Kursen, die vor über 3 Wochen eingeteilt wurden
- Löschen von Ideen in freien Kursen, die seit mehr als 180 Tagen inaktiv sind
- Aktualisierung aller Attribut-Listen, wenn vom Admin ein neues Attribut hinzugefügt wird
- Teams einteilen in Kursen, deren Deadline eingetreten ist (dies wird jede Minute geprüft), für genaue Beschreibung siehe ProFi Function
- Push-Mitteilungen versenden eine Woche vor Löschung von Kursen/Ideen, bei neuen Kommentaren, bei Team-Einteilung

Website

Entwicklung

Am Anfang der Projektlaufzeit sollte die Website eine browserbasierte Kopie der App werden. Um möglichst viele fertige Module bei der Programmierung der Website wieder verwenden zu können, wurde React Native Web gewählt. Zur Erstellung des Projekts wurde der Guide „Create a native and web app with React Native + React Native Web“ (1) von Aurelio Merenda befolgt. Als Package Manager wurde Yarn verwendet.

Bei der Implementierung der React Native-Module mussten die exportierten Arrow Functions zu Klassen umgeschrieben werden. Es stellte sich heraus, dass einige React Native-Module nicht mit React Native Web funktionieren. Daher wurden die Module schrittweise implementiert, um die Lauffähigkeit des Moduls zu prüfen und es gegebenenfalls wieder zu entfernen. Diese Module mussten nachprogrammiert werden, wie zum Beispiel die Section-List. Swipe-Buttons konnten durch reguläre Buttons ersetzt werden. Bei der Datumsauswahl wurde bei der Nachprogrammierung das Auswahlverfahren geändert und an Web-Standards angepasst: Zur Auswahl eines Datums wird ein Monatsblatt angezeigt. Dieses Modul wurde erweitert, indem nur Daten auswählbar sind, die in der Zukunft liegen, um Falscheingaben der User zu verhindern.

Einige Module wurden nicht neu programmiert, stattdessen wurde entschieden die betroffenen Funktionen nicht auf der Website anzubieten. Zu diesen Funktionen gehört das Hochladen eines Profilbildes. Bei der Weiterentwicklung der App wurde der Funktionsumfang der App erweitert. Einige dieser neuen Funktionen wurden ebenfalls nicht für die Web-Version adaptiert. Dazu gehören Benachrichtigungen und der Dark-Mode.

Die Datenbank-Funktionen konnten von der App übernommen werden.

Unterschiede zur App

Darüber hinaus besitzt die Website den vollen Funktionsumfang der App. Für die Browser-Version mussten die Styles an einen Desktop-Bildschirm angepasst werden. Um die vom User wahrgenommenen Seiten zu reduzieren, wurden einige Module und Funktionen umgeschrieben: Die Kurssuche wird mit auf der Kurslisten-Seite angeboten, Änderungen des Profils mittels Tastatureingaben der User werden direkt auf der Profilseite angezeigt und die Texteingabe bei der Kommentarfunktion wird in einem dynamisch ein- und ausgeblendeten Footer ermöglicht.

Bekannte Fehler

Beim Anzeigen der Website wird die Warnung „measureLayoutRelativeToContainingList“ ausgegeben. Die Ursache für diesen Fehler wurde trotz Web-Recherche nicht erkannt. Da dieser Fehler nicht zum Absturz der Seite führt wurde er weiterhin ignoriert.

Beim Laden der Seiteninhalte aus der Firebase kann es dazu führen, dass die Daten nicht rechtzeitig geliefert werden und die entsprechenden Inhalte nicht angezeigt werden können. Bei wichtigen Inhalten wie der Kursliste wird deshalb ein Hinweis zum Neuladen der Seite angezeigt.

Einige State-Hooks werden nicht korrekt gesetzt. Dies wirkt sich darin aus, dass Seiteninhalte, bzw. Komponenten nach einer Aktion der User nicht aktualisiert werden: Bei dem Festlegen eines Favoriten oder eines No-Gos werden die Icons nicht geändert. Die gleiche Funktionsweise wird bei dem Mitglieds-Button genutzt, bei dem bisher kein Fehler aufgetreten ist. Eine Reduzierung der Anzahl von State-Hooks durch das Zusammenfassen von primitiven Datentypen zu Arrays oder JSON-Objekten hat das Problem nicht gelöst.

Ausblick

Mögliche Erweiterungen oder Verbesserungen des Projektes außerhalb des Rahmens von Projekt B:

ProFi Function

Weiteres Testen der Funktion mit realen Benutzern, um weitere Ausnahmefälle zu erkennen.

Datenbank

Für ein besseres Error-Handling bieten sich andere Datenbank-Systeme wie MongoDB oder Oracle an. Ein Datenbank-Wechsel ist jedoch mit großem Aufwand verbunden.

App

Die App muss, um in den jeweiligen App Stores veröffentlicht zu werden, als Deployment-Version für Android und iOS gespeichert werden.

Website

Zur Veröffentlichung der Website muss noch ein Impressum und eine Informationsseite zum Datenschutz erstellt werden.

Um die Backspace-Taste der Tastatur zum Zurücknavigieren auf der ProFi-Website nutzen zu können, kann eventuell ein Key Listener für die Backspace-Taste erstellt werden.

Quellen

1 (Seite 6): „Create a native and web app with React Native + React Native Web“, Aurelio Merenda, 29.01.2020

<https://medium.com/@aureliomerenda/create-a-native-web-app-with-react-native-web-419acac86b82>

Zuletzt aufgerufen am 15.01.2021