

README

der Weitblick-App (iOS-Version)

Aufbau:

Die App hat als Navigationelement einen TabBarController. Dieser ist in fünf Unterpunkte aufgeteilt.

Zuerst gibt es den Tab Neuigkeiten und Events. Dieser besteht aus drei Controllern. Jeweils einen Controller für News und Events und einen Controller für den Switch zwischen diesen beiden Controllern. Sowohl der News- als auch der EventController haben eine Tabelle, um die verschiedenen Einträge anzuzeigen. Von den einzelnen Tabelleneinträgen gelangt man zu einer Detailansicht der Einträge. Diese DetailController haben jeweils eine Scrollview um dynamisch langen Text anzuzeigen auch wenn dieser größer als das Handydisplay ist.

Der zweite Tab besteht wieder aus einem Switch. Hier gibt es jeweils den Projekt- und Blogcontroller sowie einen Controller, um zwischen diesen beiden zu wechseln. Der BlogController sowie der BlogDetailController haben den gleichen Aufbau wie News bzw. Events. Der ProjektController hat in den Spalten, wenn das Projekt fahrradfahrbar ist, in der Tabelle zusätzlich einen Button um dieses zum radeln auszuwählen. Außerdem gibt es oberhalb der Tabelle eine Karte, auf der alle Projekte markiert sind. Wenn man auf eine Projektspalte klickt scrollt diese Karte zu dem Standort des Projektes. Der ProjectDetailController hat zusätzlich eine Tabelle. In dieser Tabelle sind für die verschiedenen Einträge wie News, Meilensteine etc. jeweils Zellen, die erneut Tabellen beinhalten, um die verschiedenen Unterpunkte darzustellen.

Der dritte Tab zeigt den MapViewController. Hier gibt es einen Button, in dem der Name, mit dem zum Fahrrad fahren ausgewählten Projektnamen steht. Wenn der Button geklickt wird, öffnet sich ein neuer Controller mit einer Liste, in der die Projekte stehen, für die man Fahrrad fahren kann. Außerdem kann das Projekt hier geändert werden. Im Hintergrund wird eine Karte vom aktuellen Standort angezeigt. Wenn man auf den Playbutton klickt kommt man zu einem neuen Controller, dem RouteViewController. Dieser treckt die gefahrene Strecke und zeigt die Daten sowie eine Map an. Wenn man auf den Zentrierenbutton klickt zentriert sich die Karte automatisch wieder auf den Standort des Nutzers, wenn diese zuvor bewegt wurde. Mit dem Pausebutton kann das „Tracken“ der Route angehalten werden. Der Infobutton öffnet ein Popover der Projektdetailseite des Projekts, für das man gerade Fahrrad fährt. Wenn man mit dem Stopbutton das Radeln beendet kommt man zum ResultMapController. Dieser zeigt die Daten vom radeln an und zudem ein Bild von dem Projekt, für das man gerade gefahren ist. Außerdem ist auch hier eine Tabelle integriert worden, um die Sponsoren des Projektes aufzulisten.

Der vierte Tab besteht erneut aus einem Switch mit drei Controllern. Dem RankingViewController sowie dem RouteViewController. Zum Wechseln zwischen den beiden Controllern gibt es den StatsViewController. Sowohl der RankingViewController als auch der RouteViewController enthalten Tabellen, um den jeweiligen Inhalt anzuzeigen. Oberhalb der Tabelle des RouteViewControllers stehen die Beschriftungen für die einzelnen Spalten der Tabelle. Oberhalb der Tabelle des RankingViewController gibt es einen Switch, um die Sortierung von Euro zu KM und anders herum zu ändern.

Der letzte Tab ist der MoreViewController. Dieser besteht aus 4 Buttons die wieder zu neuen Controllern führen. Der oberste Button öffnet den AGBController. In diesem wurde wieder eine Scrollview eingebaut, um dynamisch viel Text anzeigen zu können. Dieser Text wird zusammen mit dem Bild oberhalb des Textes vom Server geladen. Der KontaktViewController hat die gleiche Struktur wie der AGB Controller. Des Weiteren kommt man von dem MoreViewController zum FAQViewController. Dieser besteht aus einer Tabelle mit verschiedenen Unterpunkten. Der CreditViewController hat erneut eine Scrollview. Außerdem wurde unter dem Text eine Tabelle eingefügt, um die Entwickler in einer Liste anzeigen zu können.

Schrift:

Um die vom Weitblick CD festgelegte Schrift einzubinden wurden die jeweiligen „.oft“ und „.ttf“ Dateien eingebunden und in der info.plist hinzugefügt.

Models:

Die einzelnen Entity-Models wurden den REST-Schnittstellen angepasst.

Services:

DataService.swift: Laden der Newsdaten und der Projektdaten aus der REST-Schnittstelle

LoginService.swift: Einloggen des Users mit Email und Passwort

UserService.swift: Passwort zurücksetzen, Passwort ändern, Userdaten per Token anfragen, Profilbild hochladen

LogoutService.swift: Ausloggen des Users

RegisterService.swift: Registrieren des Users

FAQService.swift: FAQ laden, Contact laden, Credits laden

SegmentService.swift: Fahrradsegmente an Server schicken, Tour ID von Server abfragen

RouteService.swift: Gefahrene Routen des Users laden

RankingService.swift: Globale Rangliste aller User laden

EventService.swift: Events laden anhand einer ID, alle Events laden

BlogService.swift: Blogs laden anhand einer ID, alle Blogs laden