

imin – Kurzdokumentation



Sinn und Zweck der Applikation

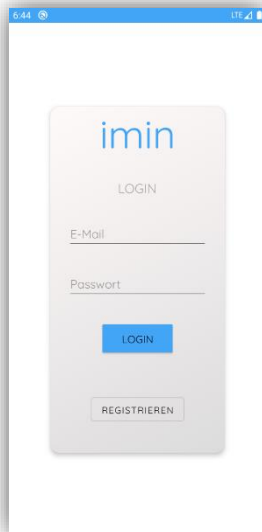
Diese Android Applikation soll als eine Art Event-Assistent dienen. Aus eigener Erfahrung weiß ich, dass es nicht allzu einfach ist eine Veranstaltung mit mehreren Teilnehmern übersichtlich zu planen. **imin** (ausgesprochen „I am in“ in Kurzform „I'm in“) soll dazu dienen, **Veranstaltungsinformationen** übersichtlich **darzustellen**, jegliche **Informationen** über **Teilnehmer** einzuholen und diese **einheitlich** und **eindeutig** hinterlegen. Gerade das ist wichtig, da sehr wohl Menschen nicht immer so einfach eindeutige Antworten abgeben können. Zudem wären damit auch Fragen wie „Du, wann war das nochmal geplant?“ oder „Wegen was ist das am Freitag?“, „Was muss ich mitbringen?“, „Welche Gerichte kann ich dort dann essen?“ immer auch für den Teilnehmer beantwortet und der Veranstalter kann sich **in Ruhe um die Planung** anhand der eingesammelten Information kümmern.

Zielgruppe

Die Zielgruppe ist breitgefächert. **imin** ist sowohl für Menschen, die gerne **Umfragen** machen würden, für **Businessleute**, die gerne ein Business-Event planen und bspw. Teilnehmer nach ihren Bedürfnissen (Essen, Trinken, Inhalte) für ein bestmögliches Erlebnis auszufragen, oder auch einfach nur **Freunde**, die ihren nächsten Grillausflug organisieren möchten. Gerade aber auch für Menschen, die Schwierigkeiten mit Technologie haben, bietet die App eine **simple Lösung**, Einladungen zu erwidern, sodass diese vom Veranstalter gut nachvollziehbar sind.

Technische Besonderheiten

- Als Datenquelle wurde hier **Firestore Realtime Database** verwendet. Dies ermöglicht auch die Online-Funktion, Veranstaltungen an bestimmte Nutzer zu versenden und auf welche zu Antworten.
- Auch die Authentifizierung wurde mit **Firestore Authentication** umgesetzt. Dabei werden alle benötigten Daten der User und den Veranstaltungen mit ihren Fragen und Antworten abgespeichert.
- Die Veranstaltungsansicht wird je nachdem ob man Eingeladener oder Ersteller ist, **mit verschiedenen Aktionen** dargestellt.
- Je nach Art der Frage, wurde beim Beantworten der Teilnehmer auch die Logik so umgesetzt, dass nur eine **einzelne Option, oder mehrere** möglich sind.
- **Designelemente** wurden erstellt, um der App eine Distanz vom üblichem Android Design zu schaffen.



Login und Registrierungs-View

Selbsterklärend, man kann sich hier anmelden, und mit einem klick auf den „Registrieren“-Knopf kommt man zu einem ähnlichen Screen, bei dem man sich Registrieren kann.

Vorbereiteter Test-Account

E-Mail: Paul@abc.de
Passwort: asd123

Mit dem Account kann man alle Funktionen der App testen und ansehen. In diesem Account befinden sich eine erstellte Veranstaltung und zwei Einladungen zu Veranstaltungen.



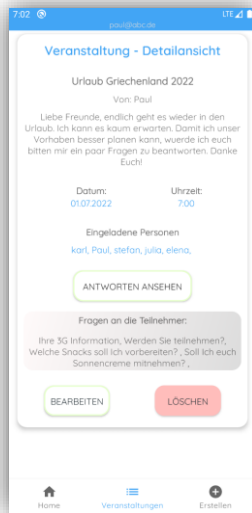
„Welcome“-View

Erster Screen, den man sieht, wenn man sich einloggt. Hier kann man sich entweder mit der Bottom-Navigation navigieren lassen oder sich ausloggen.



„Veranstaltungen“-View

Hier befinden sich die erstellten Veranstaltungen und Veranstaltungen, bei denen man eingeladen ist. Auch bemerkbar durch den minimal gehaltenen farblichen Unterschied am Rand der Listeneinträge



Veranstaltung – Detailsansicht (*Ersteller*)

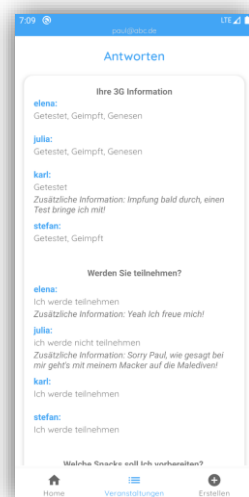
Alle wichtigsten Details einer Veranstaltung auf einem Blick. Je nachdem ob man auf eine erstellte Veranstaltung oder eine Einladung geklickt hat, hat man verschiedene Optionen von hier aus weiter.

Ersteller:

- Antworten ansehen
- Veranstaltung bearbeiten (navigiert zu „Veranstaltung erstellen“-View, mit den Eventdetails der angeklickten Veranstaltung)
- Veranstaltung löschen

Eingeladener:

- Veranstaltungsfragen beantworten



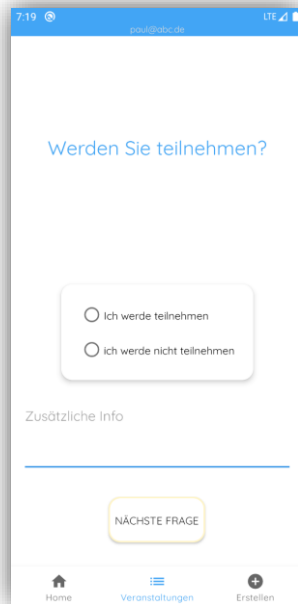
Antworten Liste-View (*Ersteller*)

Hier werden alle Antworten der Teilnehmer aufgelistet. Auf diese View kommt man, wenn man von der Veranstaltungsdetailsansicht als Ersteller auf „Antworten ansehen“ gedrückt hat.



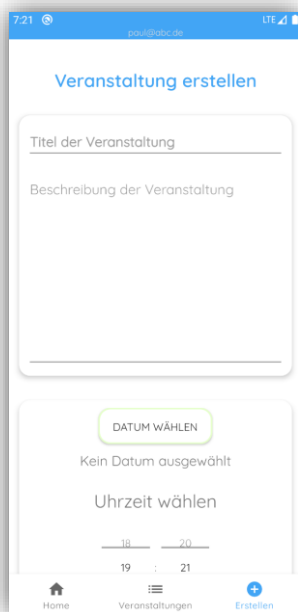
Veranstaltung – Detailsansicht (*Eingeladener*)

Von hier aus kann man die Optionen des Erstellers nicht sehen. Das heißt, es ist hier nur möglich auf eine Veranstaltung zu antworten.



Antworten-View (Eingeladener)

Hier antwortet man auf die Fragen der Veranstaltung. Je nachdem wie eine Frage erstellt wurde (Einzelne Antwortoption oder mehrere), kann man hier eine oder mehrere Optionen auswählen und bei Bedarf, eine zusätzliche Information zur Antwort hinterlassen. Am Ende bekommt man nochmal eine schöne Übersicht der abgegebenen Antworten und kann diese dann abgeben.



Veranstaltung erstellen – View

Zuletzt hier ist es möglich eine Veranstaltung zu erstellen. Hier können Titel, Beschreibung, Datum, Uhrzeit, Fragen (3G Information und Anwesenheit über Switch ganz einfach einzufügen) wie auch selbsterstellte Fragen und Teilnehmer festgelegt werden. Wenn man sich selbst in die Personenliste schreibt, kann man auch auf die Veranstaltung antworten. Sie wird dann nicht nur als erstellte, sondern auch als eingeladene Veranstaltung angezeigt. Dies ist ebenso die gleiche View, um eine Veranstaltung zu bearbeiten mit dem Unterschied, dass alle Event-details bereits in diesen Screen voreingestellt sind.

Projektstruktur der App

Die Applikation besteht aus drei Activities. Register- und LoginActivity kümmern sich um die Registrierung und die Anmeldung des Nutzers. MainActivity ist die Activity bei der alle Inhalte in Fragments dargestellt werden. Die Fragments sind in dem dafür erstellten Ordner jeweils zu finden. Zu Lernzwecken wurde ein geschachteltes Fragment verwendet, welches sich in dementsprechend benannten Ordner „nested Fragments“ befindet. Für die Navigation wurden zwei Graphen für den Authentifizierungsprozess und die Applikation erstellt. Alle wichtigen Daten die in die Firebase Realtime Database geladen und abgerufen werden sollen, wurden dementsprechend als Klasse im Ordner „model“ erstellt, um die Strukturierung der Daten zu vereinfachen. Es befinden sich drei Recyclerviews innerhalb der Applikation, deren Adapter sich im Ordner „adapters“ befinden. Unter dem Ordner „model“ befindet sich die Klasse „FirebaseHandler“. Diese Klasse ist die Schnittstelle zwischen der Applikation und der Firebase Realtime Database. Hier werden die CRUD-Operationen durchgeführt.

Technische Kurzbeschreibung der Activities und Fragments

RegisterActivity

Verwaltet die Frontend-Registrierung, Erstellt das „myUser“-Objekt und überreicht es der FirebaseHandler-Klasse („registerateUser()“) um den Nutzer in der Firebase zu registrieren.

LoginActivity

Verwaltet die Frontend-Anmeldung, Erstellt das „myUser“-Objekt“ und überreicht es der FirebaseHandler-Klasse (loginUser()) um den Nutzer bei der Firebase anzumelden.

MainActivity

Initialisiert statische Listen für die Veranstaltungen, Einladungen, Fragen und Antworten. Initialisiert die Navigation und das „Bottom Naviagtion Menu“

WelcomeFragment

Setzt den Willkommenstext, und hat einen Logout Button, welcher den Nutzer ausloggt und zurück zur LoginActivity navigiert.

myEventsFragment

Zeigt eine Übersicht mit erstellten Veranstaltungen und Veranstaltungen, bei denen man eingeladen ist, an. Man kann die Veranstaltungen anklicken, um in eine Detailansicht mit weiteren Optionen zu gelangen.

eventDetailFragment

Bietet eine Detailansicht zu einer Veranstaltung. Je nachdem ob man die Veranstaltung erstellt hat, hat man hier die Option sie zu bearbeiten, löschen oder zu beantworten.

eventAnswersFragment

Bietet eine Übersichtsseite, um alle Antworten der eingeladenen Nutzer anzuschauen

answerFragment

Hier kann ein Nutzer Frage für Frage seine Antwort zur Veranstaltung abgeben. Am Ende wird eine Übersicht der Antworten nagezeigt. Die Antworten werden dann der Veranstaltung hinzugefügt oder überschrieben.

createEventFragment

Hier kann eine Veranstaltung samt seinen Komponenten erstellt oder bearbeitet werden. Alle Eingaben werden gesammelt und eine Veranstaltung wird erzeugt, welche dann in die Datenbank geschrieben werden kann.

QuestionFragment

Erstellt eine Frage zu einer Veranstaltung, hier kann man Die Frage definieren, den Frage typ (ausschließliche eine Wahl, oder mehrere Selektionen) und die jeweiligen Antwortoptionen definieren.

personAdderFragment

Ein Nested Fragment im createEventFragment, welches eine Eingabe der zu einladenden Nutzer bietet.