

Unsere App

INVIST

Version 1.0.0, 28.Oktober 2020 | Dennis Miceli, Nikolai Schunk

Inhalt

[1 Abstract (Kurzbeschreibung) 2](#_Toc510016994)

[2 Mockups 3](#_Toc510016995)

[3 Technische Realisierung 4](#_Toc510016996)

[4 Testing 5](#_Toc510016997)

[4.1 Manuelle UI-Tests 5](#_Toc510016998)

[4.2 Testauswertung 6](#_Toc510016999)

[5 Fazit 7](#_Toc510017000)

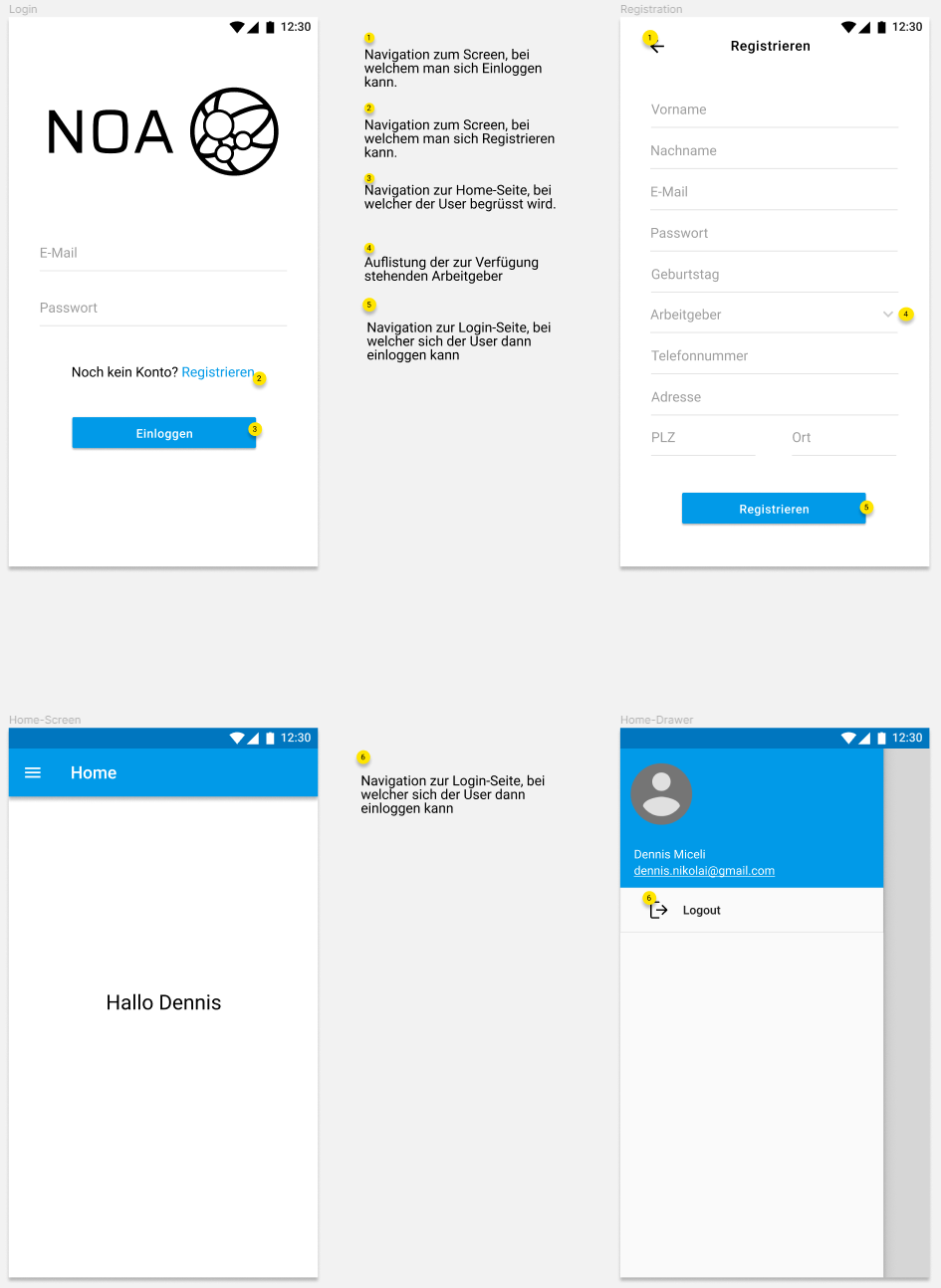
# Abstract (Kurzbeschreibung)

Wir, Nikolai und Dennis, entwickeln für den ÜK335 ein Projekt, welchem wir den Namen INVIST gegeben haben. Ziel ist es, eine deutschsprachige Applikation für Android Geräte herzustellen, über welche sich eine Person registrieren und danach einloggen kann. Für die Registrierung gibt er seine Daten ein, welche anschliessend im Front- wie auch im Backend validiert werden. Diese Daten werden in einer SQLite Datenbank lokal auf dem Endgerät gespeichert. Mit der E-Mail-Adresse und einem zuvor gesetzten Passwort, welches mit einem Verschlüsslungs-Algorithmus verschlüsselt wurde, kann sich der Benutzer einloggen. Bei einer fehlgeschlagenen Anmeldung wird eine Fehlermeldung wiedergegeben. Bei einer erfolgreichen Anmeldung wird der Benutzer bei seinem Namen begrüsst. Der Nutzer hat dann die Möglichkeit sich wieder abzumelden, woraufhin er wieder den Login-Screen sieht.

GitHub: <https://github.com/Th3Dennis/uek335NikolaiDennis>

Figma: <https://www.figma.com/file/i88N3kNy8UKmVCOUPFPtkp/Android-GUI-Design-Files?node-id=15566%3A153>

## Mockups



1. **Startactivity / Login**Innerhalb der Startactivity / Login wird dem Benutzer zentral ein Login-Formular präsentiert. Oberhalb des Formulars wird das Logo der App, welches von NOA ist, platziert. Die Eingabefelder sind mit einem «hint» versehen, welcher dem Benutzer zeigt, welche Daten er eingeben muss.

Unter den Eingabefelder ist ein Text mit dem Inhalt «Noch kein Konto? Registrieren» versehen, wobei die blaue Farbe des Wortes «Registrieren» den Benutzer zeigt, dass es sich um eine Art Link handelt. Unter diesem Text befindet sich ein Button, auf welchem «Einloggen» steht. Dabei soll der Login-Button farblich hervorgehoben werden da dieser öfters benutzt wird und die Hauptfunktion der Activity darstellt.

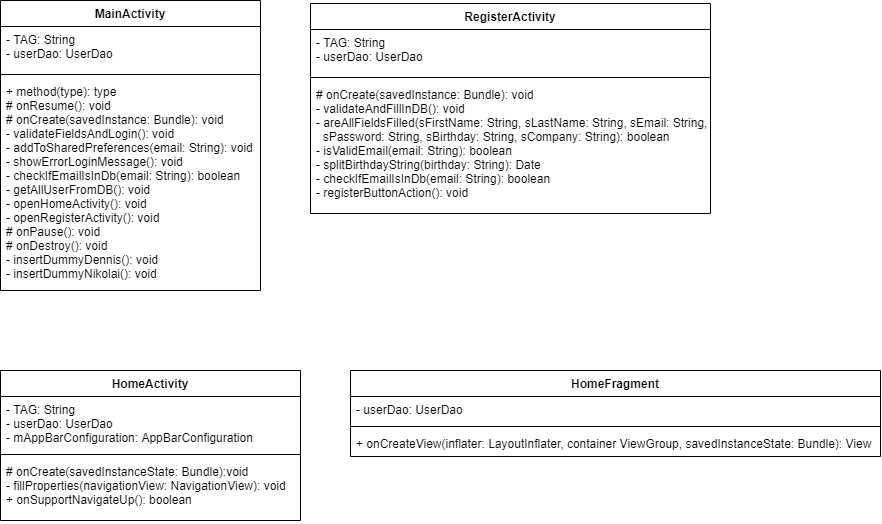
1. **Registration**

Innerhalb der Registration-Activity wird dem Benutzer ein Formular vorgewiesen. Die Eingabefelder sind jeweils mit einem «hint» versehen, welcher dem Benutzer indiziert, um welche Daten es sich handelt. Oberhalb des Formulars wird mit dem Schriftzug «Registrieren» dem Benutzer gezeigt, auf welcher Seite er sich befindet. Falls er zur Login-Seite zurück möchte, kann er den Pfeil links oben drücken. Auch ist es dem Benutzer möglich, die Navigationstaste «zurück» auf dem Android Gerät zu drücken. Beim Feld «Arbeitgeber» ist ein «Dropdown» vorhanden, bei welchem die verfügbaren Arbeitgeber angezeigt werden und ein Element ausgewählt werden kann. Unter dem Formular ist ein Button, welcher den Text «Registrieren» beinhaltet. Wird dieser Knopf gedrückt, weist es den Benutzer auf die Login Seite weiter, auf welcher er sich dann mit den soeben eingegebenen Daten einloggen kann. Der Login-Button wurde farblich hervorgehebt, da dieser die Hauptfunktion der Seite darstellt. Ausserdem ist er in einer dunkleren Farbe, wenn nicht das komplette Formular ausgefüllt wurde. Sobald alle Felder Werte beinhalten, wird der Knopf in die Standartfarbe gewechselt.

1. **Home**

Innerhalb der Home-Activity wird dem Benutzer in der Mitte des Bildschirms ein «Hallo <Name>» gezeigt. Oberhalb dieses Willkommenstextes befindet sich ein Schriftzug, auf welchem «Home» geschrieben steht. Dieser Text vermittelt dem User, dass er sich nun auf der Hauptseite, welche nach «Home» benannt ist, befindet. Auf der linken Seite des Textes, befindet sich ein «Hamburger Menu» Icon, welcher die Funktionalität hat, den Drawer zu öffnen. In diesem Home-Drawer befindet sich ein Foto des Benutzers. Falls der Benutzer kein Foto besitzt, wird ein Standartlogo benutzt. Unterhalb befindet sich in einer ansprechenden Farbe der Vor- und Nachname auf der ersten Zeile und auf der Zweiten Zeile die E-Mail-Adresse. Unterhalb befindet sich ein Kästchen, welches einen, falls gedrückt, zur Login-Activity führt. In diesem Kästchen befindet sich ein Text mit dem Inhalt «Logout», wie auch links davon ein Logo, welches dem Benutzer visuell denselben Nutzen zeigt.

# Technische Realisierung



**MainActivity**

In dieser Klasse / Aktivität befindet sich die Logik, welche es dem User ermöglicht sich einzuloggen. In der onCreate() Methode wird der userDao Variable das DAO der Datenbank zugewiesen. Auch überprüft es in dieser Methode, ob sich bereits die Variable «activeUser» in der Shared Preference «invistPrefs» befindet. Falls dies der Fall sein sollte, wird der Benutzer automatisch auf die HomeActivity weitergeleitet. Hier befinden sich ebenfalls zwei OnClickListener. Der erste Reagiert, falls man auf den blau geschriebenen «Registrieren» Text klickt. Somit wird man zur RegisterActivity weitergeleitet. Der zweite Listener reagiert, sobald man auf den «Einloggen» Button klickt. Dieser ruft dann die Methode validateFieldsAndLogic auf.

Methode validateFieldsAndLogic()

In dieser Methode wird die eingegebene E-Mail und das Passwort in eine String variable gespeichert. Nach der Prüfung der Verfügbarkeit der E-Mail, wird das verschlüsselte Passwort mit der E-Mail als Primärschlüssel von der Datenbank geholt. Dieses wird dann mit dem eingegebenen Passwort überprüft. Bei erfolgreicher Authentifizierung wird die E-Mail in die bereits erwähnte Shared Preference gespeichert und anschliessend wird der Benutzer auf die Home Activity weitergeleitet.

**HomeActivity**

Erneut wird hier in der onCreate Methode, wie bei der MainActivity, das UserDao instanziiert. Ebenfalls in dieser Methode wird ein Menu Item für den Drawer erstellt. Dieses Menu Item besitzt einen Listener, welcher den User ausloggt, falls dieser reagiert. Ebenfalls wird die «activeUser» Variable aus dem Shared Preference «invistPrefs» entfernt.

Methode fillProperties()

In dieser Methode wird aus der Shared Preference die E-Mail des Benutzers entnommen. Anhand von dieser E-Mail-Adresse wird dann der User aus der Datenbank bezogen. Dies wird gemacht, dass man damit den Vornamen, den Nachnamen und die E-Mail-Adresse im Drawer angezeigt bekommt.

**RegisterActivity**

In dieser Klasse / Aktivität befindet sich die Logik, welche es dem User ermöglicht ein Konto zu erstellen, mit welchem er sich anschliessend anmelden kann. In der OnCreate() Methode wird der userDao Variable das DAO der Datenbank gegeben. Anschliessend wird der Back-Button in der ActionBar hinzugefügt. Auch wird der Titel der ActionBar zu «Registrieren»(mit der String Ressource) gewechselt. Anschliessend wird dem Spinner die Werte zugewiesen. Diese Werte werden der String Ressource aus einem Array entnommen. Ebenfalls befindet sich in der onCreate() Methode ein OnClickListener, welcher reagiert, sobald der Registrier-Button geklickt wird. Der Listener ruft dann die validateAndFillInDb() Methode auf.

Methode validateAndFillInDb()

In dieser Methode werden die eingegebenen Daten in die jeweiligen String Variablen abgefüllt. Zuerst wird überprüft, ob alle Pflichtfelder befüllt sind. Wenn alle Pflichtfelder befüllt sind, überprüft die Methode, ob es die E-Mail bereits in der Datenbank gibt. Da die E-Mail den Primärschlüssel des Users darstellt, kann dieser jeweils nur einmal existieren. Falls es diese E-Mail noch nicht gibt, werden alle eingegebenen Daten auf ihre jeweilige Richtigkeit überprüft. Wenn alle Felder richtig sind, wird das Passwort in einen Hash umgewandelt und alle Daten werden in ein neues User Objekt gefüllt. Dieses Objekt wird anschliessend in die Datenbank gefüllt und der User wird zur MainActivity weitergeleitet.

**User**

In dieser Klasse befinden sich die Felder, welcher der Benutzer in der RegisterActivity eingegeben hat. Der Benutzer wird als Objekt dieser Klasse gespeichert. Die String Variable E-Mail wird in der Datenbank als Primärschlüssel verwendet, da diese für jeden Benutzer einzigartig sein muss.

**HomeFragment**

Dieses Fragment befindet sich in der HomeActivity. In dieser Klasse wird die Hallo-Nachricht definiert.

**AppDatabase**

In dieser abstrakten Klasse wird in der Methode getAppDb() eine Datenbank auf dem Endgerät erstellt. Falls der Benutzer bereits eine Datenbank auf dem Android Gerät besitzt, wird diese zurückgegeben.

**Converters**

Diese Java Klasse wird vom benutzten ORM namens «Room» benutzt. Diese beiden beinhalteten Methoden werden gebraucht, um den eingegebenen Geburtstag des Users in einen Long zu speichern, wie auch andersrum.

Beschreibt hier, wie ihr eure komplexe Komponente technisch umgesetzt habt. Zur Darstellung der technischen Umsetzung wird ein UML-Diagramm empfohlen, welches zusätzlich in Textform beschrieben wird. Erklärt kurz die wichtigsten Klassen und Methoden und deren Zusammenspiel. Ein Entwickler, der dieses Kapitel liest, sollte schnell an diesem Projekt weiterentwickeln können.

# Testing

## Manuelle UI-Tests

|  |  |
| --- | --- |
| Abschnitt | Innhalt |
| ID | Testfallnummer (ST = Systemtest) |
| Anforderungen | Welche Anforderungen werden durch diesen Testfall abgedeckt. (User Stories) |
| Vorbedingungen | Was muss gegeben sein, damit dieser Test durchgeführt werden kann? |
| Ablauf | Welche Schritte werden bei der Durchführung des Tests durchlaufen? |
| Erwartetes Resultat | Was sollte nun passiert sein? |

|  |  |
| --- | --- |
| Abschnitt | Innhalt |
| ID | ST-01 |
| Anforderungen | US-01; US-03 |
| Vorbedingungen | In der Datenbank existiert ein Benutzer, welcher gesperrt ist. |
| Ablauf | 1. Die App wird gestartet damit das Login-Formular erscheint 2. Der korrekte Benutzername sowie das korrekte Passwort werden eingegeben. 3. Der Button mit dem Label „Login“ wird geklickt |
| Erwartetes Resultat | Ein Toast mit dem Text «Login erfolgreich» wird angezeigt. Die App wechselt zu der Ansicht mit den favorisierten Schwimmbäder |

## Testauswertung

Zusammenfassung aller durchgeführten Tests. Nur fehlgeschlagene Tests und Tests mit Bemerkungen müssen in der folgenden Tabelle aufgelistet werden.

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Bemerkungen |
| ST-01 | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| … | … | … |
|  |  |  |

# Fazit

Hier kommt eure Reflexion zum Projekt.