Documentation uK 335

Julian Werner & Felix Schneider
8th of May 2020

Abstract

The goal was to create an Android app for the uK 335. In our case we had to create an app where you could register a user and login. After logging in you come to a new window, where you get greeted. In that window you don't have a lot of functionality. The only thing that you can do is logout where you get sent back to the login screen.

Functional requirements

- Register as a user

Register:		
Must-have	First name	Max. 50 Characters
Must-have	Last name	Max. 50 Characters
Must-have	E-Mail address	Max. 100 Characters & correct email format
Must-have	Password	Input in password field
Optional	Photograph	Make profile picture with camera app

As pretty much everywhere else all the inputs have to be validated. The user data then gets stored in the SQLite Database on the smartphone / emulator. If the inputs are incorrect a Toast gets shown describing what appears to be wrong

- Login as a user

Login:	
E-Mail address	
Password	Input in password field

Again here as well, if the input fields are incorrect a Toast gets shown. After the login a new activity should be shown where the user gets greeted. E.g. (Welcome Firstname Lastname!)

Non functional requirements

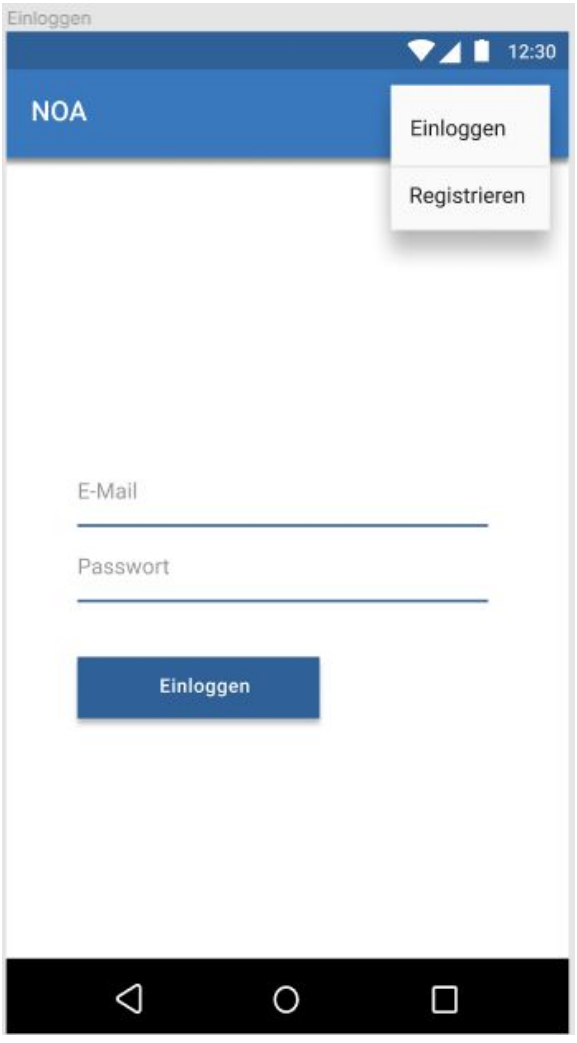
- Users get saved in the local SQLite database read from there as well
- The password gets encrypted and saved in the database
- Photographs are made with the camera app
- The app is only available in german
- The profile pictures only need to be displayed correctly in the portrait mode

Mockups

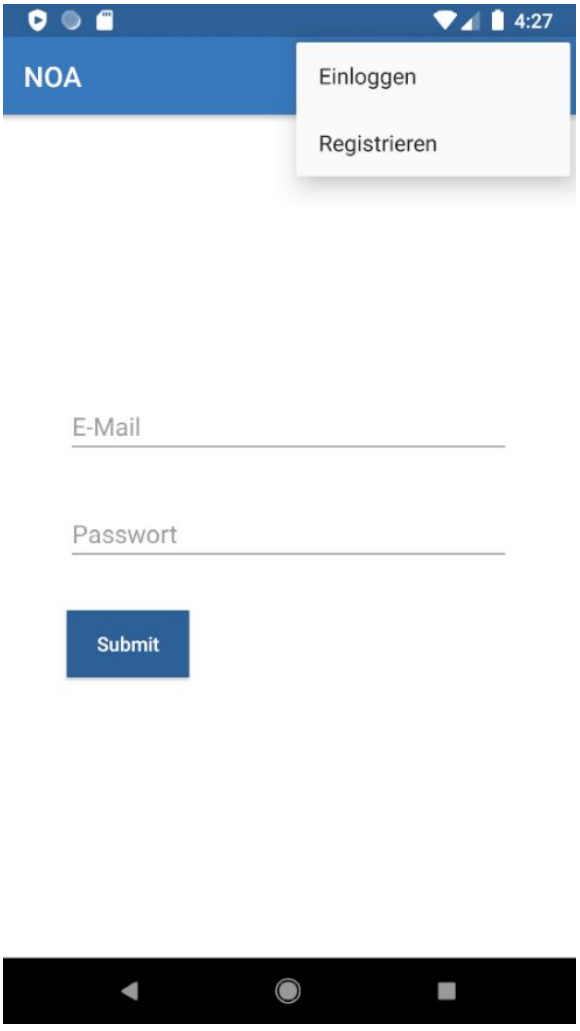
We decided to go with a sleek and modern design. The blues were specifically chosen to match the Noser Young logo. We had a couple of different ideas but in the end we went with this one.

Login

Mockup

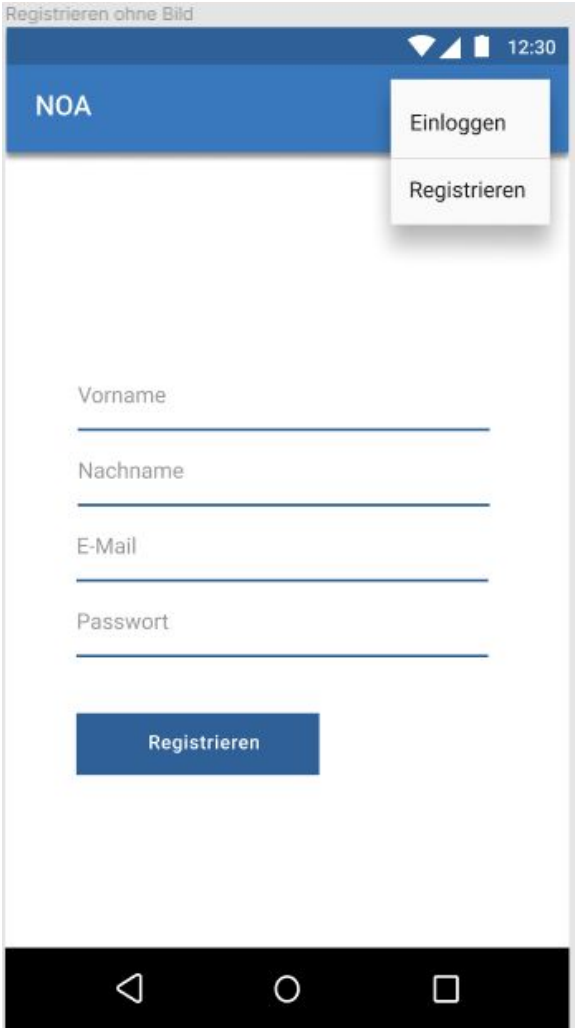


App

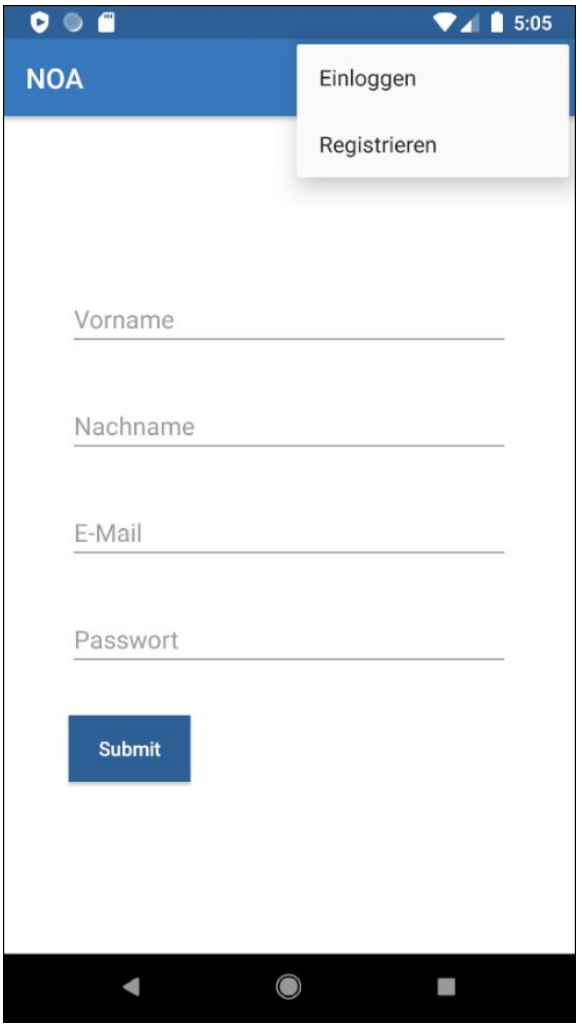


Register

Mockup



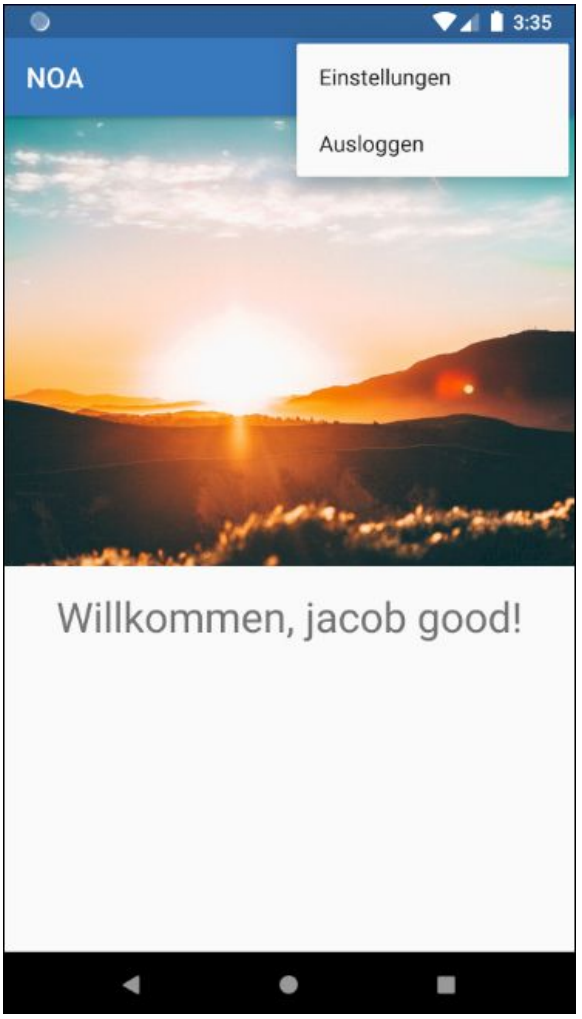
App



Dashboard

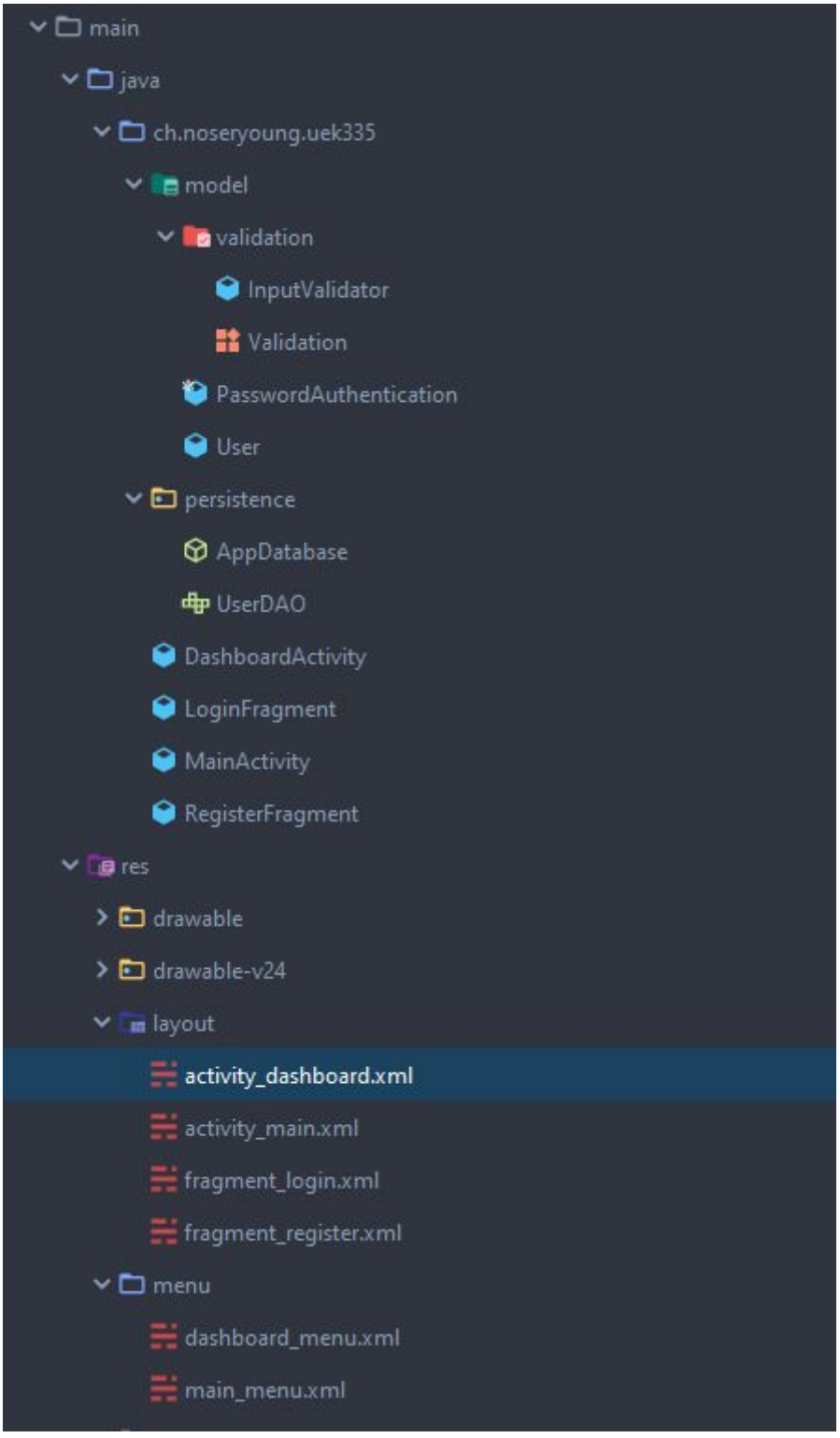
Mockup

App



Technical Realisation

File tree



We Completed this application with a Main Activity that has a Fragment container. The Fragment container gets filled with the respective fragment, depending on which the user currently wishes to view. The two fragments are fragment_login and fragment_register. When the user registers an account, the input is validated and inserted to the SQLite database locally on the users phone. This is done through a connection with the DAO.
The Dashboard activity gets opened once the user logs themselves in to the application. The Login Fragment adds the User as a serializable to the bundle, which is handed over with an Intent. The Dashboard can then deserialize this Object and get the User Object. From there we can display the users name.

Database

User entity / table

```
@Entity
public class User {

    @PrimaryKey(autoGenerate = true)
    private int id;

    @ColumnInfo(name = "first_name")
    private String firstName;

    @ColumnInfo(name = "last_name")
    private String lastName;

    @ColumnInfo(name = "email")
    private String email;

    @ColumnInfo(name = "password")
    private String password;
```

The Android SDK can generate an SQLite database from annotations added to the Object that we want to persist. With the @Entity annotation we mark it as such. @PrimaryKey is the primary key of the table, for which we said they should be auto generated.

Testing

Manual UI Tests

Test Case	1				
Test Case Description	Signup				
Test Environment	Noa App 1.0 Pixel 2 Virtual Device				
Functionality to be Tested	- Input validation for register - Change to login and register fragment				
Date	08.05.2020				
Tester	Julian Werner & Felix Schneider				
Test Steps					
Nr.	Action	Expected Result	Actual Result	Success	Comment
1	Open App	Login page becomes visible.	“		
2	Prompt menu top right	Menu appears with tabs “Einloggen” and “Registrieren”	“		
3	Select “Registrieren” in menu	Login page fragment gets displayed	“		
4	Input fields are empty & **Prompt submit**	Displays Toast: “Bitte fülle alle Felder aus”	“		
5	**Following inputs:** Vorname: valid Nachname: valid Email: not valid Password: valid & **Prompt submit**	Displays Toast: “Bitte benutze eine gültige Email ”			
6	**Following inputs:**	Displays Toast:			

	Vorname: valid Nachname: valid Email: valid Password: not valid & **Prompt submit**	“Dein Passwort muss mind. 8 Zeichen lang sein, gross- und kleinbuchstaben und eine Zahl sowie ein Spezialzeichen beinhalten”			
7	**Following inputs:** Vorname: more than 50 characters Nachname: valid Email: valid Password: not & **Prompt submit**	Displays Toast: “Vorname darf nicht länger wie 50 Zeichen sein”			
8	**Following inputs:** Vorname: valid Nachname: more than 50 characters Email: valid Password: not & **Prompt submit**	Displays Toast: Nachname darf nicht länger wie 50 Zeichen sein			
9	User already exists	Display Toast: “Benutzer mit dieser Email existiert bereits”			
10	Inputs are correct & **Prompt submit**	User gets created and saved in the database	“		

Test Case	2
Test Case Description	Login
Test Environment	Noa App 1.0 Pixel 2 Virtual Device
Functionality to be Tested	- Input validation for login - Change to login and register fragment

Date		08.05.2020			
Tester		Julian Werner & Felix Schneider			
Test Steps					
Nr.	Action	Expected Result	Actual Result	Success	Comment
1	Open App	Login page becomes visible.	“		
2	Prompt menu top right	Menu appears with tabs “Einloggen” and “Registrieren”	“		
3	Select “Einloggen” in menu	Login page fragment gets displayed	“		
4	Select “Einloggen” in menu	Login page fragment gets displayed	“		
5	Input fields are empty & **Prompt submit**	Displays Toast: “Bitte fülle alle Felder aus“	“		
6	Wrong email and or password	Displays Toast: “Falsche Logindaten”	“		
7	Credentials are correct & **Prompt submit** (After restarting app)	You get forwarded to the dashboard	“		Generally works, except the App needs to be restarted → Problems with Fragments

Test Case	3
Test Case Description	Dashboard
Test Environment	Noa App 1.0 Pixel 2 Virtual Device
Functionality to be Tested	- Greetings

	- Change to login fragment				
Date	08.05.2020				
Tester	Julian Werner & Felix Schneider				
Test Steps					
Nr.	Action	Expected Result	Actual Result	Success	Comment
1	User logs in	Dashboard activity gets opened	“		
2	Greetings get displayed	Following gets displayed: “Welcome, first name last name!”	“		
3	Prompt menu top right	Menu appears with tabs “Einstellungen” and “Ausloggen”	“		
4	Select “Einstellungen” in menu	Nothing happens as this tab is only for demonstration purposes	“		
5	Select “Ausloggen” in menu	You get “logged out” -> forwarded back to login page	“		

Testevaluation

The App generally works as we expected it to do. All Test cases succeeded, except for the Problem that the application has to be restarted in order to login after registering an account. We looked at this Problem with Gianni and Carlo, but in our short discussion they couldn’t help us either.

Review

Problems

Github

Throughout the project we had some problems with Github. One these were weird errors while un/shelving and un/patching which we had never seen before. As Felix hadn't used Github in quite a while it took him some time to get into it again in the beginning. On the other hand Julian had worked with it during the last couple months and was sort of a pro which was very useful at times.

Fragments

In the end we had some problems with the fragments. If the Fragment was changed during runtime, the fragment that was first displayed at startup wasn't able to execute a listener. This means the user can't consecutively register and log in.

What went well

Communication

The communication went pretty well as we were in a call together at all times which made the exchange between each other pretty easy. The Pull Requests on Github were also a good option for discussing code. We also had a project backlog, where we had all of our ToDos listed.

What we learned

Android Studio

Although we couldn't go into much depth and mostly looked at the basics of Android software development I think that we can say for ourselves that we learned a lot. Although we had some hiccups throughout the project with all the new Information we learned, we think that due to those Problems, we learned even more than we expected.

Android Studio is very similar to Intellij, as it is based off of it, which is why we could navigate around fairly easily, because we already knew how the IDE worked.