

# Arbeitsprotokolle der Diplomarbeit

Jakob M. Holzinger

17.06.2022

## 1 15.06.2022 - Besprechung mit BizzNetIT

Heute sind ich und Jonas zur Firma gefahren, um zu besprechen, welchen Umfang die App haben soll und wie dies technisch realisiert werden soll. Anschließend habe ich ein Dokument für Haslinger Peter aufgesetzt, da wir diesen gerne als Diplomarbeitsbetreuer hätten. Wir werden in den kommenden Wochen eine Besprechung mit der Firma, zwecks Datenbankaufbau, haben. Wir bekommen von Ihnen einen Zugang auf ihren Datenbankserver, damit wir dort eine Datenbank aufbauen und befüllen können. Ich habe ebenfalls ein GitHub-Repository erstellt. Ich war in der Firma von 12:45 bis 16:15, die Besprechung hat von 14:00 - 15:45 gedauert.

## 2 16.06.2022 - Ordnerstruktur des Repositories und der App

Heute habe ich begonnen die Ordnerstruktur in unserem Repository, sowie der App, zu erstellen. Dazu habe ich ein doc/ und ein src/ Verzeichnis erstellt. In src/ habe ich ein neues Flutter Projekt erstellt und unsere Ordnerstruktur implementiert. Da es ein großes Projekt wird und wir es von anfang an übersichtlich und gut strukturieren wollen, habe ich im Internet recherchiert, wie das andere Developer, in großen Flutter Projekten machen. Ich bin dort auf ein gutes Video [1] gestoßen, in welchem eine gute Ordnerstruktur aufgezeigt wird. Ich habe diese etwas an unsere Anforderungen angepasst. Dauer: 1h

## 3 17.06.2022 - Authentication

Heute habe ich begonnen den Sign Up Screen zu erstellen. Ich habe mir zuerst ein Design für das Textfeld überlegt. Wir wollen etwas schlichtes, jedoch auch modernes Design für die App. Deshalb habe ich einfach die Ecken abgerundet, und die Textfarbe auf weiß gestellt, da wir in der App aktuell ein dunkles Design haben. Ich habe ebenfalls noch eine Beschriftung hinzugefügt. Man muss beim verwenden des Widgets, ein label (Was soll eingegeben werden?) und einen TextInputType (E-Mail, Passwort, Text, etc.) übergeben. Diese Werte werden über

den Konstruktor instanziert. Es gibt ebenfalls eine `getInput()` Funktion, mit der man den Text des Textfeldes erhält. Dies habe ich mit einem `TextEditingController` [2] erreicht. Ich habe um zu testen, ob das Textfeld und die `get`-Methode funktionieren, eine `FloatingActionButton` hinzugefügt, welcher mich auf den Splash-Screen navigiert [3], welcher erscheint, wenn die Registrierung erfolgreich war. Dort zeige ich aktuell nur die Eingaben des Users an. Es funktioniert einwandfrei, das heißt ich kann bereits die eingegebenen Daten auslesen und mit ihnen arbeiten. Das wird wichtig für meine nächsten Schritte. Dauer: 2h

## **4 18.06.2022 - Überprüfung auf Richtigkeit der LogIn Daten**

Heute habe ich die authentication domain geschrieben. Diese besteht aus 3 Klassen: `CheckPassword`, `CheckEmail` und `AuthenticationManager`. `CheckPassword` bzw `CheckEmail` bekommen die Daten per Konstruktor übergeben und überprüfen ob diese zur gegebenen E-Mail passen. Dies wird in einer Funktion überprüft, welche einen `bool` zurückgibt. Der `AuthenticationManager` bekommt Objekte von `CheckEmail` bzw `CheckPassword` per Konstruktor übergeben und hat eine Methode, welche ebenfalls einen `bool` zurückgibt. Hier werden einfach die beiden Methoden der Check-Klassen aufgerufen, falls beide `true` zurückgeben, sind die Daten korrekt. Ansonsten wird eine `Exception` geworfen, ob nun das Passwort oder die E-Mail falsch ist. Dauer: 1h

## **5 19.06.2022 - Authentifizierung mit Datenbank**

Heute habe ich eine Testdatenbank mit xampp erstellt. Diese hat die gleiche/eine ähnliche Struktur wie die folgende. Dann habe ich zwei php files geschrieben, wie hier beschrieben[4]. Anschließend habe ich eine `RegistrationManager` Klasse erstellt mit der `startRegister` Methode, welche die Daten aus den Textboxen per Konstruktor erhält. Diese Funktion ruft dann das php file auf, welches für den Datenbankzugriff dient. Dasselbe habe ich mit dem `LoginManager` gemacht, um Log-Ins zu handeln. Dauer: 3h

## **6 20.06.2022 - Registrieren und LogIn mit PHP und MySQL**

Heute wollte ich die Fehler im PHP Script beheben. Dazu habe ich im Internet recherchiert wie ich meine Version von PHP checken kann und, je nach dem ob ich eine Version 7 bzw. 5 habe, die syntax anpassen muss. Denn ich erhielt eine Menge `Syntax-Errors`. Ich habe lange probiert und in den error logs von xampp gelesen, bis ich alle Fehler eliminieren konnte. Nun kann ich die Daten der Registrierung auch in der Datenbank speichern. Ich habe zusätzlich noch die Splash Screens vollständig implementiert und die Fehlerbehandlung im PHP

Script verbessern. Es wird mit RegEx überprüft ob die Eingaben auch im richtigen Format sind und ob die Email bereits registriert ist. Wenn die Registrierung fehlschlägt wird man auf einen SplashScreen umgeleitet, auf dem steht welche Eingaben falsch sind. Es befindet sich ebenfalls ein Knopf auf der Seite, welcher einen wieder zur SignUp Page leitet. Es gibt ebenfalls einen SplashScreen falls die Eingaben erfolgreich waren, dieser ist aber evtl. nur temporär, da man bei erfolgreicher Registrierung auch gleich auf den Home-Screen weitergeleitet werden könnte. Das ganze hat 5h gedauert.

## 7 21.06.2022 - Log-In fertiggestellt

Heute habe ich die LoginPage erstellt. Dazu habe ich wie bei der Registrationsseite meine benötigten TextBox Widgets eingebunden. Ich habe ebenfalls das PHP Script angepasst. Es ist eine Fehlerbehandlung implementiert, genau so wie beim register script. Es gibt 3 Fälle: Die Mail ist nicht registriert, das Passwort zur Mail ist falsch, oder beide Eingaben sind korrekt. Je nach dem welcher Fall eintritt, wird ein String zurückgegeben. Dieser wird dann wie beim Registrierungsvorgang in der Flutter App ausgelesen und dann werden die weiteren Schritte eingeleitet. Es gibt noch ein kleines Problem und zwar speichere ich die Passwörter als Hash in der Datenbank ab. Deshalb muss ich zur Validierung den Hash aus der Datenbank auslesen und vergleichen. Dort gibt es noch irgendeinen Conversion Error, diesem bin ich bisher noch nicht auf die Schliche gekommen. Dauer: 2h

## References

- [1] Tadas Petra. *Flutter File Structure for Big Projects*. URL: <https://www.youtube.com/watch?v=Mt41FpSS-Vo>. (accessed: 16.06.2022).
- [2] Flutter. *Retrieve the input of a text field*. URL: <https://docs.flutter.dev/cookbook/forms/retrieve-input>. (accessed: 17.06.2022).
- [3] Flutter. *Navigation in Flutter*. URL: <https://docs.flutter.dev/cookbook/navigation/navigation-basics>. (accessed: 17.06.2022).
- [4] W3 Schools. *Select Data with MySQLi*. URL: [https://www.w3schools.com/php/php\\_mysql\\_select.asp](https://www.w3schools.com/php/php_mysql_select.asp). (accessed: 19.06.2022).