

Lernende Verwaltung

Abschluss Projekt für ÜK 335

Version 2.0.0, 05.Juni.2020 | Salome Treichler und Nate Widmer

Inhaltsverzeichnis

Kurzbeschreibung	2
Mockups.....	3
Übersicht	3
Beschreibung	4
Use Cases	5
Use Case Diagramm	5
Brief Use Cases	5
Technische Realisierung.....	7
Testing.....	11
Manuelle UI-Tests	11
Testauswertung.....	14
Fazit	15

Kurzbeschreibung

Ausgangslage

Im Rahmen vom ÜK 335 soll als Projekt eine App für Android Smartphones entwickelt werden. Die Grundidee ist eine Mobile Applikation mit den Funktionen von NOA zu erstellen, eine Software der Firma Noser Young zur internen Administration und der Kommunikation mit externen Berufsbildner von Lernenden. Verschiedene Gruppen übernehmen Teile der Funktionen, um die Arbeit aufzuteilen. Uns wurde die Lernenden Verwaltung als Aufgabenbereich zugeteilt.

Ziel

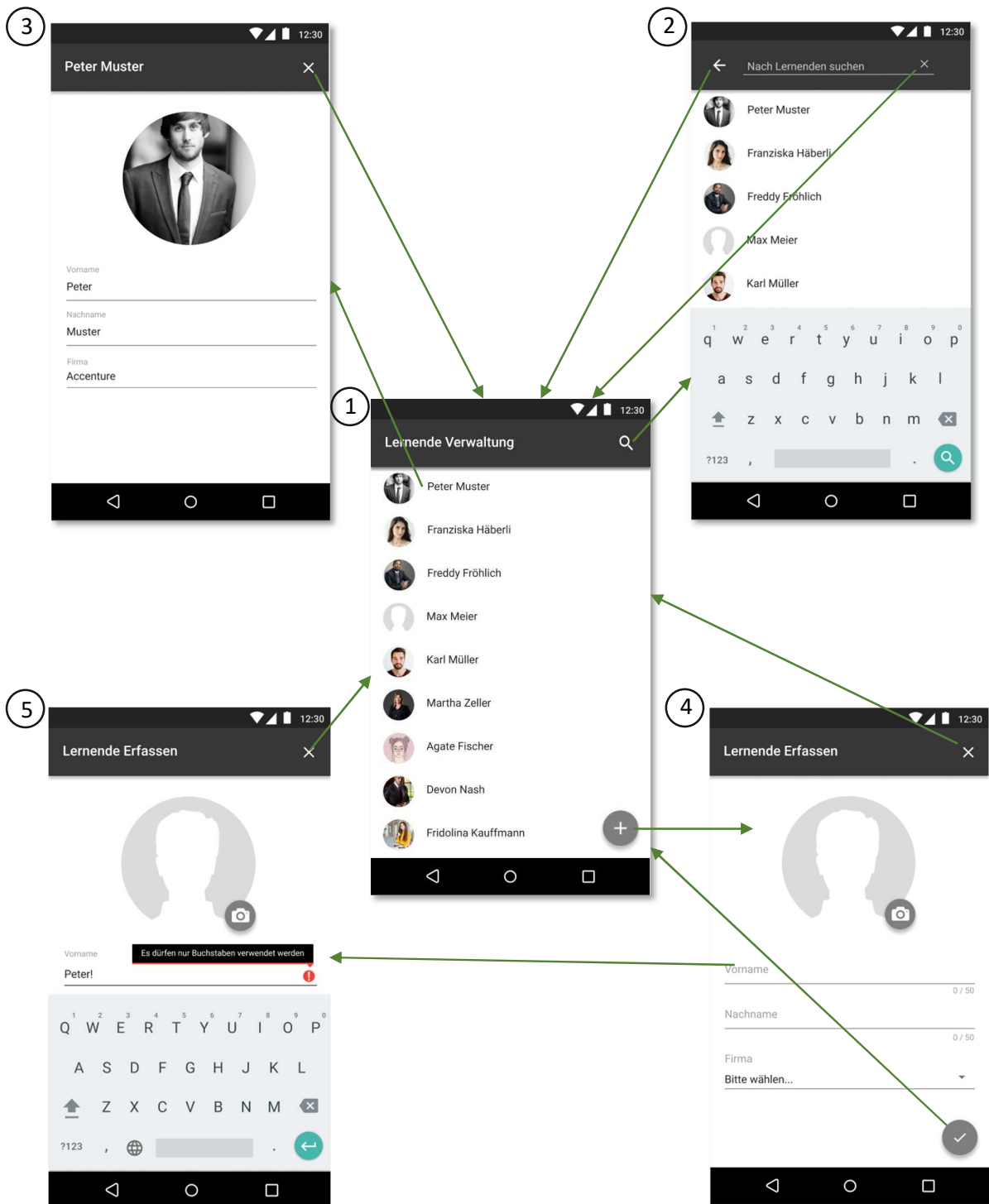
Dieses Projekt deckt nur ein Teil der grossen NOA Applikation ab. Das Ziel ist es eine Simple Verwaltung von Lernenden bereitzustellen. Es soll auf der Startseite eine Übersicht über alle Lernenden vorhanden sein. Ausserdem soll man einen neuen Lernenden erstellen und jeden einzelnen Lernenden als Detailansicht aufrufen können. Für die Personalisierung soll es möglich sein ein Profilbild beim Erstellen eines Lernenden zu definieren. Das Design soll schlicht sein und zum Design der Web Application NOA passen.

Ausblick

Da es in der kurzen Zeit nur möglich ist eine simple Verwaltung umzusetzen gibt es viele Möglichkeiten diese Smartphone App zu erweitern. Es würde definitiv noch einen Modus für das nachträgliche Bearbeiten eines Lernenden Profils brauchen da dies bis jetzt noch nicht möglich ist. Ausserdem könnte man auch wahlweise noch mehr Attribute zum Lernenden hinzufügen, wie zum Beispiel eine Adresse. Ein weiteres Feature wäre das Auswählen eines Profilfotos aus der Galerie. Da auch noch andere Teile der Funktionen von anderen Teams umgesetzt wurden könnte man diese zusammenfügen und das Design anpassen für ein einheitliches und stimmiges Gesamtbild.

Mockups

Übersicht



Beschreibung

Das Design und die Farbpalette wurden von NOA übernommen, um ein Einheitliches Bild zu schaffen.

1 - Lernende Übersicht

Diese Ansicht ist die Startseite der App. Hier werden alle Lernenden mit Vor- und Nachnamen und ihrem Profilbild aufgelistet. Beim Klicken auf ein Item der Liste wird die Detailansicht des geklickten Lernenden angezeigt. Wenn man einen neuen Lernenden erfassen möchte dann klickt man auf den Plus Button in der unteren rechten Ecke und wird dann zur Create Ansicht weitergeleitet. Die Liste kann auch durchsucht werden, indem man das Lupen Icon oben rechts drückt.

2 - Lernende Übersicht durchsuchen

Diese Ansicht ist in der Startseite. Sie wird aufgerufen, wenn das Lupen Icon geklickt wird. Es geht automatisch die Tastatur auf und man kann die Liste durchsuchen. Über das "x" Icon kann man die Suche Abbrechen und über den Pfeil nach Links kommt man wieder zurück auf die ursprüngliche Ansicht der Startseite.

3 - Detail Ansicht Lernende

In der Detail Ansicht sieht der User das Profil eines Lernenden. Das Profilbild wird gross oben in der Mitte angezeigt und darunter die gespeicherten Infos des Lernenden. Das Layout ist genau das gleiche wie in der Erstell Ansicht. Über das "x" in der oberen Rechten Ecke kehrt man wieder zur Startseite zurück.

4 - Neue Lernende erfassen

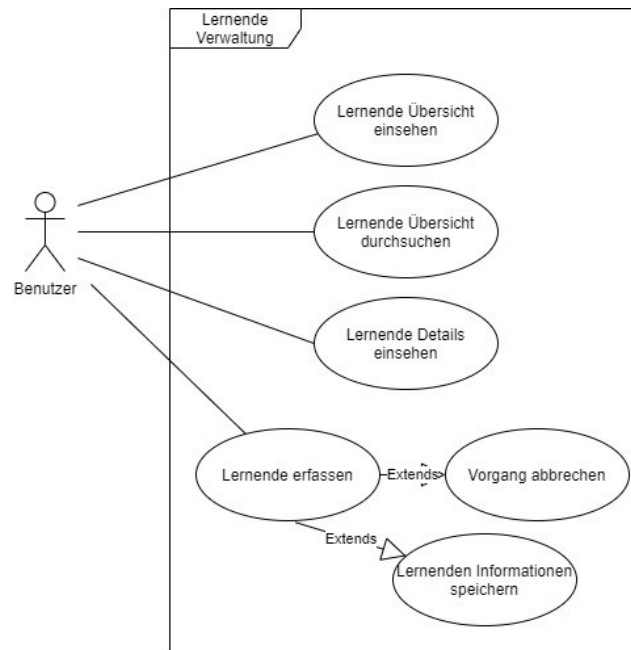
Diese Ansicht ist ein Formular, um einen neuen Lernenden zu erfassen. Oben in der Mitte ist ein Platzhalter für das Profilbild und rechts darunter befindet sich ein Kamera Button und wenn man auf diesen klickt, öffnet sich die Kamera App. Wenn man ein Foto gemacht hat, wird das im Platzhalter angezeigt. Für den Vor- und Nachnamen gibt es ein Textfeld und für die Firma ein Dropdown. Das Dropdown wurde gewählt, damit es bei der Firma keine Fehler bei der Eingabe gibt. Wenn man auf den Button unten rechts klickt und alle Angaben richtig sind, wird der Lernende erstellt und man kehrt zurück auf die Lernende Übersicht. Über das "x" in der oberen rechten Ecke kehrt man wieder zur Startseite zurück und bricht dabei den Erstellprozess ab.

5 - Validierung der Textfelder

Diese Ansicht zeigt wie das Formular reagiert, wenn die Validierung fehlschlägt. Wenn eine falsche Eingabe im Textfeld stattfindet, erscheint ein roter Kreis mit einem Ausrufezeichen rechts am Textfeld. Darüber oder darunter erscheint die Fehlermeldung in einem schwarzen Quadrat mit weissem Text, welche meldet, was bei der Eingabe noch nicht stimmt. Die Fehlermeldung verschwindet wieder sobald der Fehler behoben wurde.

Use Cases

Use Case Diagramm



Brief Use Cases

Abschnitt	Inhalt
ID	US-01
Akteure	App Benutzer
Beschreibung	Der Benutzer hat beim Starten der App eine Übersicht über alle Lernenden in einer Liste. In der Liste werden Vor- und Nachname angezeigt sowie das Profilbild des Lernenden.

Abschnitt	Inhalt
ID	US-02
Akteure	App Benutzer
Beschreibung	Der Benutzer kann auf der Startseite die Liste der Lernenden mit einer Suchfunktion durchsuchen.

Abschnitt	Inhalt
ID	US-03
Akteure	App Benutzer
Beschreibung	Wenn der Benutzer auf der Startseite auf einen Lernenden in der Liste klickt öffnet sich eine Detailansicht. In der Detailansicht werden das Profilbild sowie Vor- und Nachname und die Firma des Lernenden angezeigt.

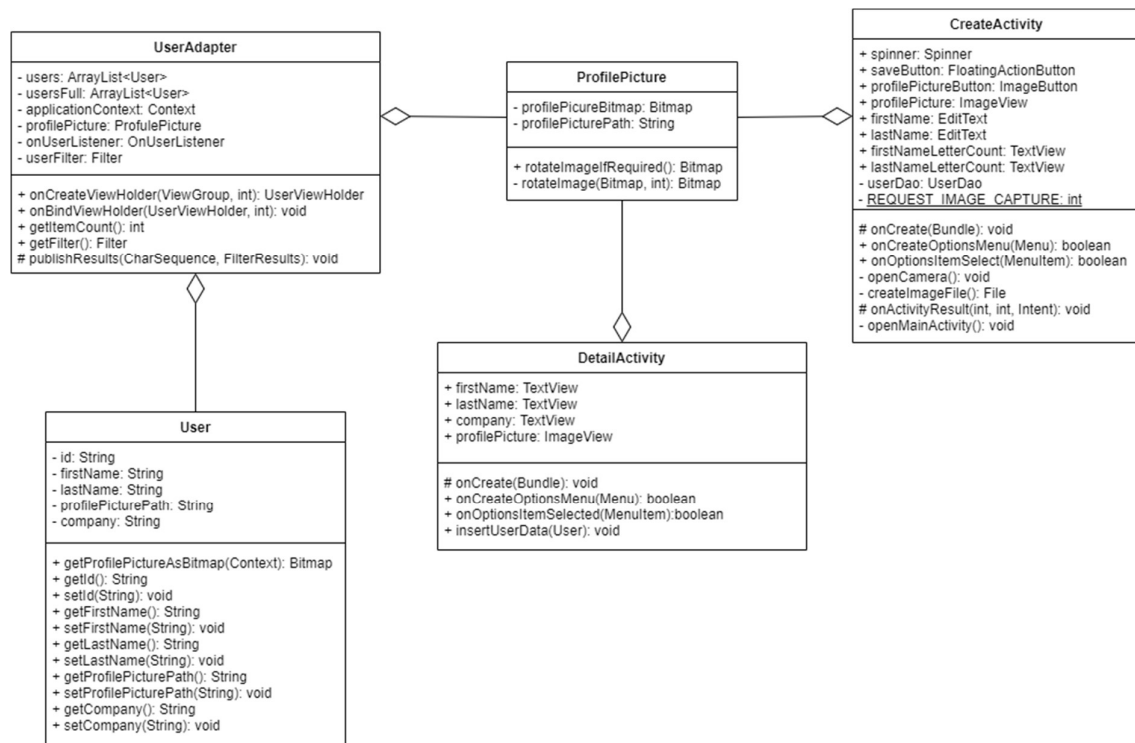
Abschnitt	Inhalt
ID	US-04
Akteure	App Benutzer
Beschreibung	Der Benutzer kann über ein Formular einen neuen Lernenden erfassen. In dem Formular kann er den Vor- und Nachnamen eingeben, aus einer Auswahl die Firma des Lernenden auswählen sowie mit der Kamera ein Bild aufnehmen und dieses als Profilbild setzen.

Abschnitt	Inhalt
ID	US-05
Akteure	App Benutzer
Beschreibung	Der Benutzer kann den Erstell-Vorgang eines Lernenden Abbrechen.

Abschnitt	Inhalt
ID	US-06
Akteure	App Benutzer
Beschreibung	Der Benutzer kann nachdem korrekten Ausfüllen des Formulars einen Lernenden speichern. Der Benutzer wird informiert, wenn eine Eingabe falsch ist und kann den Lernenden ohne korrekte Eingaben nicht speichern

Technische Realisierung

In unserer technischen Realisierung erklären wir das Zusammenspiel unserer Komponenten im Zusammenhang mit dem Profilbild. Das Klassendiagramm (siehe Bild unten) enthält alle wichtigen Komponenten, welche etwas mit dem Ablauf des Erstellens und Einsetzens des Profilbildes zu tun haben.



User

In unserer User Klasse definieren wir die Entität "User" mit all seinen Attributen. Darunter befindet sich auch der *profilePicturePath*, ein String, in welchem der Dateipfad des Profilbildes gespeichert wird. Die User Klasse enthält eine Funktion *getProfilePictureAsBitmap* welche den Kontext der Applikation benötigt und eine Bitmap des Profilbildes zurückgibt.

ProfilePicture

Unsere ProfilePicture Klasse ist eine Support Klasse und enthält das Profilbild als Bitmap, sowie den Pfad des Bildes. Wir haben hier die *rotateImageIfRequired* und die *rotateImage* Funktion, welche beim Aufrufen unser Profilbild korrekt ausrichtet und es als Bitmap zurückgibt. Dies benötigen wir bei den Ansichten, zum Beispiel im RecyclerView, der CreateActivity und der DetailActivity. Um herauszufinden, welche Ausrichtung das Bild vor dem Drehen hat, verwenden wir hier ein ExifInterface welches uns Daten über das Bild liefert.


```

public Bitmap rotateImageIfRequired() throws IOException {
    if (profilePicturePath != null) {
        ExifInterface ei = new ExifInterface(profilePicturePath);
        int orientation = ei.getAttributeInt(ExifInterface.TAG_ORIENTATION,
            ExifInterface.ORIENTATION_NORMAL);

        switch (orientation) {
            case ExifInterface.ORIENTATION_ROTATE_90:
                return rotateImage(profilePictureBitmap, 90);
            case ExifInterface.ORIENTATION_ROTATE_180:
                return rotateImage(profilePictureBitmap, 180);
            case ExifInterface.ORIENTATION_ROTATE_270:
                return rotateImage(profilePictureBitmap, 270);
            default:
                return profilePictureBitmap;
        }
    }
    return profilePictureBitmap;
}

```

UserAdapter

Unsere UserAdapter Klasse erhält, wenn sie in der MainActivity Klasse erstellt wird eine Liste der User, welche in der Datenbank erfasst sind und ein Objekt der ProfilePicture Klasse. In der Funktion *onBindViewHolder* initialisieren wir das *profilePicture* Objekt mit dem Profilbild des momentanen Users aus der Liste. Dieses Profilbild setzen wir später in der gleichen Funktion in der View ein und um sicherzustellen, dass das Profilbild richtig herum ist, rufen wir die Funktion *rotateImageIfRequired* aus der ProfilePicture Klasse auf.

CreateActivity

In der CreateActivity Klasse befinden sich unsere Kamera Funktionen. Zunächst fragen wir unseren Benutzer, ob wir auf die Kamera zugreifen und Files in seinen Dateien erstellen dürfen. Klick der Benutzer dann auf den Kamera Button in unserer View, startet die Funktion *openCamera*.

```

private void openCamera() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        File photoFile = createImageFile();

        if (photoFile != null) {
            profilePicturePath = photoFile.getAbsolutePath();
            Uri photoURI = FileProvider.getUriForFile(CreateActivity.this,
                "ch.noseryoung.lernendeverwaltung", photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
        }
    }
}

```

In dieser Funktion erstellen wir zunächst einen impliziten Intent für das Öffnen der Kamera, mithilfe des `MediaStore.ACTION_IMAGE_CAPTURE`, welcher den Benutzer ein Bild mit der Kamera machen lässt und dieses dann an uns zurücksendet. Wir überprüfen dann, ob es überhaupt möglich ist mit diesem Gerät diese Activity zu starten und sollte dies der Fall sein, kreieren wir ein File für unser Foto, mithilfe der Funktion `createImageFile`. Nach dem Aufrufen dieser Funktion prüfen wir, ob das File erfolgreich erstellt werden konnte und weisen dann unserer Variable `profilePicturePath` den Dateipfad dieses Files zu, damit wir es später in der Datenbank abspeichern können.

Nun verwenden wir den FileProvider, welchen wir in unserem Manifest File definiert haben, um den Content URI für das File zu erstellen. Mit dem Content URI können wir das Foto mit anderen Apps teilen. Wir erstellen unseren Content URI mit unserem momentanen Kontext, unserer Authority und dem erstellten File.

Nun geben wir unserem Intent für das Starten der Kamera das `photoURI` als Extra mit, damit der Intent weiss, wo er das Bild nach dem Erstellen abspeichern muss. Die letzte Zeile der Funktion startet dann die Activity mit dem `takePictureIntent` Intent.

Sobald das Foto gemacht wurde startet die `onActivityResult` Funktion.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    ProfilePicture profilePicture;
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bitmap imageBitmap = BitmapFactory.decodeFile(profilePicturePath);
        profilePicture = new ProfilePicture(imageBitmap, profilePicturePath);

        try {
            imageBitmap = profilePicture.rotateImageIfRequired();
        } catch (IOException e) {
            e.printStackTrace();
        }

        this.profilePicture = findViewById(R.id.avatarPicture);
        this.profilePicture.setImageBitmap(imageBitmap);
    }
}
```

In dieser Funktion überprüfen wir ob beim Erstellen des Fotos alles funktioniert hat. Dann erstellen wir eine Bitmap mit dem Dateipfad, mit welcher wir wie schon beim UserAdapter das `profilePicture` initialisieren und das Bild falls nötig rotieren. Schlussendlich setzen wir die Bitmap in der View ein.

DetailActivity

In der DetailActivity zeigen wir das Profilbild so wie beim UserAdapter im RecyclerView an.

Manifest.xml und file_paths.xml

Um mit unserer App ein Foto machen zu können müssen wir noch einige Dinge im Manifest.xml File hinzufügen. Zum einen geben wir hier an, dass unsere App die Kamera des Gerätes verwendet, das wir die Erlaubnis haben diese zu verwenden und Schreiberlaubnis für die Dateien des Gerätes haben.

```
<uses-feature android:name="android.hardware.camera"
             android:required="true" />

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.CAMERA" />
```

Weiter unten definieren wir unseren FileProvider. Hierbei geben wir unsere Authority an (Wir sehen hier, dass diese Authority dieselbe ist, welche wir dem Content URI mitgegeben haben). Wir sagen dem FileProvider ausserdem, wie unser Dateipfad heissen soll.

```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="ch.noseryoung.lernendeverwaltung"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths"></meta-data>
</provider>
```

Im file_paths.xml definieren wir den Pfad in welchem wir unsere Files abspeichern wollen.

```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
    <external-files-path name="noa_profile_pictures" path="Pictures"/>
</paths>
```

Testing

Manuelle UI-Tests

Abschnitt	Inhalt
ID	ST-01
Anforderungen	US-01
Vorbedingungen	Die App muss installiert sein
Ablauf	1. App über Startbildschirm des Android Handys öffnen
Erwartetes Resultat	Die App soll starten und die Startseite anzeigen mit einem Floating Button, um zur Erstellt Ansicht zu gelangen und einem Such Icon in der App Bar. Der Titel der Seite ist "Lernende Verwaltung"

Abschnitt	Inhalt
ID	ST-02
Anforderungen	US-05
Vorbedingungen	Der Benutzer befindet sich auf der Startseite. Es wurden noch keine User erstellt.
Ablauf	1. Runder Plus Button in der unteren rechten Ecke klicken 2. Kreuz Icon in App Bar drücken
Erwartetes Resultat	Nach dem Klicken des Plus Buttons wird die Formular Ansicht für das Erstellen eines Lernenden geöffnet. Nach dem Klicken des Kreuz Icons wird der Benutzer auf die Startseite weitergeleitet. Die Mitte der Startseite ist leer, es wurde kein User erstellt.

Abschnitt	Inhalt
ID	ST-03
Anforderungen	US-04, US-06
Vorbedingungen	Der Benutzer befindet sich auf der Seite zum Erstellen eines neuen Lernenden. Das Formular ist leer.
Ablauf	1. Vorname "Max" eingeben 2. Nachname "Muster" eingeben 3. Firma "Six" wählen 4. Save (Hacken) Button Klicken
Erwartetes Resultat	Die Felder konnten ausgefüllt werden. Nach dem Klicken auf den Save Button wurde der Benutzer auf die Startseite weitergeleitet und der neu erstellte User Max Muster wurde angezeigt mit einem Avatar Platzhalter für das Profilbild.

Abschnitt	Inhalt
ID	ST-04
Anforderungen	US-04, US-06
Vorbedingungen	Der Benutzer befindet sich auf der Seite zum Erstellen eines neuen Lernenden. Das Formular ist leer.
Ablauf	<ol style="list-style-type: none"> 1. Vorname "Frida", Nachname "Fröhlich", Firma "Google" eingeben 2. Kamera Button neben dem Profilbild betätigen 3. Foto aufnehmen 4. Save Button Klicken
Erwartetes Resultat	Die Felder konnten ausgefüllt werden. Beim Klicken auf den Kamera Button wurde die Kamera App geöffnet und man konnte ein Foto schießen. Nach dem Bestätigen des Fotos wurde der Benutzer zum Formular zurückgebracht. Nun wird oben das gerade entstandene Foto mit der richtigen Ausrichtung angezeigt. Nach dem Speichern wird der Benutzer zur Startseite weitergeleitet und der Neu erstellte User wird mit Profilbild und Name angezeigt.

Abschnitt	Inhalt
ID	ST-05
Anforderungen	US-06
Vorbedingungen	Der Benutzer befindet sich auf der Seite zum Erstellen eines Lernenden. Das Formular ist leer.
Ablauf	<ol style="list-style-type: none"> 1. Beim Feld Vorname mehr als 50 Zeichen eingeben 2. Feld Nachname leer Lassen 3. Bei Firma den Default Wert lassen 4. Save Button Klicken
Erwartetes Resultat	Während man beim Vornamen zu viele Zeichen eingibt, wird sobald das Limit von 50 Zeichen überschritten ist ein Fehler angezeigt. Nach dem Verlassen des Textfeldes sieht man die Anzahl Buchstaben unter dem Textfeld. Beim Versuch den User zu speichern wird beim Vornamen der Error für zu viele Zeichen angezeigt und beim Nachnamen, dass das Feld nicht leer sein darf.

Abschnitt	Inhalt
ID	ST-06
Anforderungen	US-06
Vorbedingungen	Der Benutzer befindet sich auf der Seite zum Erstellen eines Lernenden. Das Formular ist leer.
Ablauf	<ol style="list-style-type: none"> 1. Beim Feld Vorname Zahlen eingeben 2. Beim Feld Nachname Sonderzeichen eingeben 3. Bei Firma den Default Wert lassen 4. Save Button Klicken
Erwartetes Resultat	Nach dem Klicken des Save Buttons wird bei beiden Textfeldern ein Error angezeigt, dass man nur Buchstaben verwenden darf.

Abschnitt	Inhalt
ID	ST-07
Anforderungen	US-02
Vorbedingungen	Der Benutzer befindet sich auf der Startseite. Es wurden bereits 2 User erstellt.
Ablauf	<ol style="list-style-type: none"> 1. Such Icon in der App Bar klicken 2. "M" als Suche eingeben 3. Zurück Pfeil Icon klicken.
Erwartetes Resultat	Nach dem Klicken des Such Icons wird die Suchansicht mit einem Textfeld in der App Bar geöffnet. Nach der Eingabe des Buchstabens wird nur noch Max Muster anstatt der vollständigen Liste angezeigt. Nach dem Klicken des Pfeil Icons ist wieder die Start App Bar zu sehen. Es wird noch immer nur Max Muster angezeigt, da wir das "M" aus der Suchleiste noch nicht entfernt haben.

Abschnitt	Inhalt
ID	ST-08
Anforderungen	US-03
Vorbedingungen	Der Benutzer befindet sich auf der Startseite. Es wurden bereits 2 User erstellt.
Ablauf	<ol style="list-style-type: none"> 1. Auf Listen Eintrag mit Frida Fröhlichs Namen klicken. 2. Kreuz Icon in App Bar klicken
Erwartetes Resultat	Nach dem Klicken auf den Listen Eintrag mit Frida Fröhlichs Namen wird eine Detailansicht ihres Profils angezeigt. Der Benutzer sieht Fridas Profilbild, ihren Vor- und Nachnamen und die Firma, in welcher sie arbeitet. Nach dem Klicken des Kreuz Icons wird der Benutzer auf die Startseite weitergeleitet.

Testauswertung

ID	Erfolgreich	Bemerkungen
ST-01	Ja	-
ST-02	Ja	-
ST-03	Ja	-
ST-04	Ja	-
ST-05	Ja	-
ST-06	Ja	-
ST-07	Ja	Beim Öffnen der Suche wird nicht sofort das Textfeld angezeigt, man muss noch einmal auf einen Lupe Icon klicken, um diese Ansicht zu erhalten.
ST-08	Ja	-

Fazit

Erfreuliches

Das ganze Projekt hat uns sehr viel Spass gemacht. Wir fanden es spannend etwas komplett Neues lernen zu dürfen. Uns liegt das Frontend und auch das Design von GUIs, deshalb konnten wir effizient ein schönes Mockup mit Figma erstellen und haben dort keine Zeit verloren. Wir sind beide schnell mit der neuen Programmierumgebung klargekommen und konnte so ohne grosse Hindernisse mit dem Projekt starten.

Wir haben uns die Arbeit während des Projektes gut aufgeteilt und haben uns in vielen Hinsichten immer wieder ergänzen können.

Dass wir ein gutes Team sind war uns von Anfang an schon bewusst, aber auch während dieses Projekts wurde uns wieder klar, dass wir als ein eingespieltes Team eine gute Arbeit abliefern können.

Schwierigkeiten

Natürlich gab es währen dem Prozess auch immer wieder Schwierigkeiten. Bei manchen haben wir die Lösung schnell gefunden bei anderen hatten wir ein bisschen länger. Was uns grosse Schwierigkeiten gemacht hatte, war die Search Funktion. Wir hatten lange bis wir eine Lösung finden konnten und unser Berufsbildner hatte uns dabei auch noch ein wenig unterstützt. Mit der jetzigen Lösung sind wir mehr oder weniger zufrieden, hätten aber sicher, wenn wir Mehr Zeit gehabt hätten auch diese Komponente noch perfektioniert.

Zufriedenheit mit dem Endergebnis

Wir sind sehr zufrieden mit unserem Ergebnis und sind stolz auf unsere Leistung. Wir konnten alle gewünschten Features umsetzen und haben sogar noch weitere hinzugefügt, wie zum Beispiel die Suchfunktion. Wir sind auch mit dem Design sehr zufrieden, weil es modern ist und von den Farben und Formen zur bereits bestehenden NOA Webapp passt.

Neue Erkenntnisse und Erlerntes

Wir konnten mit diesem Projekt sehr viel neues lernen. Wir haben Android Studio kennengelernt und alles rundum die Android App Entwicklung. Wir haben auch unser Zeitmanagement verbessern können, weil wir managen mussten, wie wir Projekt und ausführliche Dokumentation in der gegebenen Zeit fertigstellen können. Auch dass wir guten Code schreiben müssen, welcher den Regeln von Clean Code entspricht, ist uns in diesem Projekt noch einmal klar geworden. Denn jeder musste den Code verstehen können und dies ist nur gegeben, wenn man gewisse Regeln einhält. Alles in Allem waren die zwei Wochen eine lehrreiche Zeit.