

Softwarepraktikum 2022

Einführungsaufgaben

Abgabefrist: 30.4.2022

Allgemeine Hinweise:

- Die Meilensteinabnahme erfolgt unter aktiver Beteiligung aller Gruppenmitglieder. Im Fall der Abwesenheit eines Gruppenmitglieds während der Meilensteinabnahme muss ein ärztliches Attest vorgelegt werden. Selbiges gilt auch für die mit dem jeweiligen Tutor/der jeweiligen Tutorin vereinbarten Besprechungstermine in den Tutorien.
- Die Abnahme dieser Einführungsaufgabe erfolgt durch den jeweiligen Tutor.
- Die Abgabe aller Artefakte erfolgt über das teameigene Git-Repository.
- Für Fragen bzw. Anregungen außerhalb der betreuten Rechnerzeiten nutzen Sie bitte auch das Forum in Ilias.

1 GitHub

Alle Artefakte, die bei der Bearbeitung des Praktikums entstehen, sollen in einem Git-Repository gespeichert und damit unter Versionskontrolle gestellt werden. Hierzu empfehlen wir <https://github.com/> an. Legen Sie dort für Ihr Team ein privates Repository an. Benennen Sie dieses nach dem Schema:

SP21_Gruppe[A-E]

Fügen Sie alle Teammitglieder, den/die Tutor/in Ihrer Gruppe und die Nutzer *lucieflek* und *caisa-admin* als *Collaborator* hinzu. Die Nutzernamen der Tutoren finden Sie in der Gruppenbeschreibung in Ilias.

Nutzen Sie Git aktiv! Verwenden Sie insbesondere Branches um parallel zu arbeiten und Ihre Ergebnisse später zusammenzuführen. Führen Sie insbesondere **vor jeder Meilensteinabgabe** Ihre Ergebnisse zusammen!

2 Erste Schritte

Die offizielle Dokumentation zur Entwicklung mit Android (<https://developer.android.com/docs>) enthält neben der Android- API, Tutorials und Code-Beispielen wichtige Informationen zur Architektur und dem Lebenszyklus von Android-Apps und wird vermutlich die wichtigste Informationsquelle während Ihres Praktikums sein.

Als Einstieg bieten sich einige Android Tutorials an:

<https://developer.android.com/courses>

<https://github.com/wnyao/android-development-course>

<https://github.com/AlexeyBuzdin/android-development-course>

<https://web.stanford.edu/class/cs193a/lectures.shtml>

<https://www.geeksforgeeks.org/android-projects-from-basic-to-advanced-level/>

<https://github.com/androiddevnotes/awesome-android-learning-resources>

Wer lieber ein Video gucken möchte, findet zum Beispiel mit dem Android Java Tutorial 2020 einen schnellen Einstieg in die grundlegenden Konzepte der App-Erstellung: <https://www.youtube.com/watch?v=bwXGY8cNLGY>

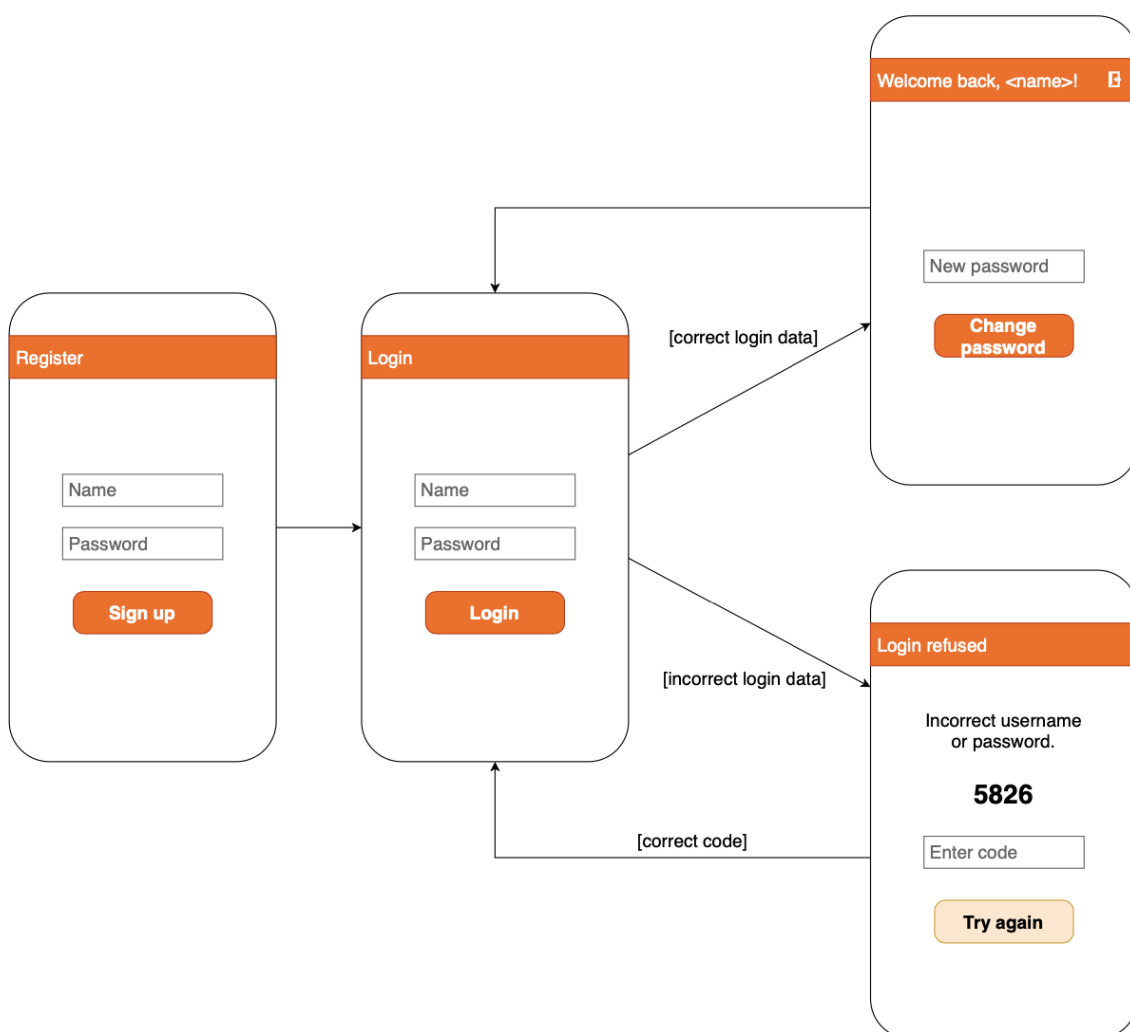
Im Zuge des Praktikums wird jedes Team eine mittelmäßig komplexe Android-App erstellen. Da viele vermutlich keine Vorkenntnisse in der Android-Programmierung besitzen, sollen Sie die ersten Wochen dazu nutzen, individuell Grundkenntnisse zu erwerben, bevor Sie als Team ein größeres Projekt in Angriff nehmen. Die folgenden Übungsaufgaben sollten daher unter den Teammitgliedern aufgeteilt werden. Sie werden sich zur Bearbeitung absprechen und auf Schnittstellen und Datenstrukturen einigen müssen, darüber hinaus sollte jede Teilaufgabe aber möglichst selbständig von der verantwortlichen Person erledigt werden!

Folgende Lernziele sollen erreicht werden. Kenntnisse über

1. Das Erstellen einer Android-App und ihre Ausführung (im Emulator oder auf einem Mobilgerät),
2. Das Erstellen einer Aktivität und Gestalten der Benutzeroberfläche,
3. Die Umsetzung von Interaktionen mit der Benutzeroberfläche,
4. Die Implementierung von Anwendungslogik,
5. Die Verwendung von Klassen zur Datenhaltung,
6. Die Interaktion zwischen Aktivitäten.

3 Übung

Erstellen Sie eine kleine App, die eine minimalistische Login-Funktion umsetzt. Diese soll aus den folgenden vier Aktivitäten bestehen. Eine grobe Skizze des erwarteten Ablaufs und der Nutzeroberflächen ist in Abbildung 1 dargestellt.



Legen Sie für diese daher am besten einen eigenen Ordner in Ihrem Git-Repository an und entwickeln Sie die Haupt-App später in einem anderen Ordner. Die Nutzerdaten müssen *nicht* über den Lebenszyklus der App hinaus persistiert werden. Zur Umsetzung von Knöpfen werden Sie in dieser ersten App vermutlich onClick-Events verwenden.

Eine hilfreiche Diskussion zum Thema *onClickListener* vs. *onClick-XML-Attribut*, zwei Möglichkeiten, solche Events zu behandeln, findet sich auf stackoverflow : <https://stackoverflow.com/questions/21319996/android-onclick-in-xml-vs-onclicklistener>

Sie können hier gerne beide Konzepte ausprobieren.

3.1 Registrierungs-Bildschirm

Beim Starten der App soll der Nutzer mit einem Registrierungs-Bildschirm begrüßt werden. Auf diesem soll er nach einem Nutzernamen und einem Passwort gefragt werden. Über einen Knopf kann der Nutzer die Eingaben bestätigen und sich bei der App registrieren. Die eingegebenen Nutzerdaten werden in der App gespeichert und der Nutzer wird zum Login-Bildschirm weitergeleitet.

Hinweis: Üblicherweise würde eine App vermutlich mit dem Login-Bildschirm starten und von diesem aus eine Registrierung ermöglichen. Um die Komplexität der Aufgabe gering zu halten, beginnt diese App jedoch immer mit einer einmaligen Registrierung.

3.2 Login-Bildschirm

Hat sich der Nutzer registriert, muss er sich bei der App einloggen. Es werden der Nutzernamen und das Passwort des Nutzers abgefragt. Über einen Knopf kann der Nutzer seine Eingaben bestätigen und den Login durchführen. Daraufhin werden die eingegebenen Daten mit den in der App hinterlegten Nutzerdaten verglichen. Stimmen Nutzernamen und Passwort überein, wird der Nutzer zu einem Willkommens-Bildschirm weitergeleitet. Stimmen diese nicht überein, wird der Loginfehler-Bildschirm geöffnet.

3.3 Willkommens-Bildschirm

Der Nutzer wird mit seinem Namen begrüßt. In der Aktionsleiste befindet sich ein Logout-Knopf, über den der Nutzer wieder zum Login-Bildschirm gelangt. Außerdem hat er die Möglichkeit, sein Passwort zu ändern. Er muss hierzu ein neues Passwort eingeben und dieses über einen Knopf bestätigen.

3.4 Loginfehler-Bildschirm

Eine Fehlermeldung zeigt an, dass der Login fehlgeschlagen ist. Zudem wird dem Nutzer ein 4-stelliger Code angezeigt. Wird dieser korrekt vom Nutzer eingegeben, wird ein Knopf aktiviert, der den Nutzer wieder zum Login-Bildschirm führt, sodass er einen neuen Login-Versuch unternehmen kann.

Hinweis: Wer möchte, kann den angezeigten Code von der App generieren lassen. Für den Zweck dieser Übung genügt es jedoch, einen festgelegten Code in Form einer Konstante im Quellcode zu hinterlegen.

4 Testen

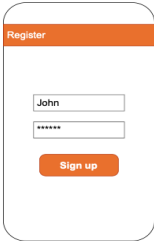
Testen Sie die Korrektheit Ihrer Umsetzung, indem Sie Systemtests für die beiden möglichen Pfade, die in der App durchlaufen werden können, durchführen. Dokumentieren Sie Ihre Systemtests entweder tabellarisch:

Erfolgreicher Login nach Registrierung


Vorbedingung: Der Nutzer startet die App.	
Ablauf: Der Registrierungs-Bildschirm wird angezeigt. Der Nutzer gibt seine Zugangsdaten ein und bestätigt diese. Der Login-Bildschirm öffnet sich. Der Nutzer gibt erneut die gewählten Zugangsdaten ein und bestätigt diese. ...	
Erwartetes Verhalten:	Tatsächliches Verhalten:

oder durch Screenshots mit Beschriftung:

Erfolgreicher Login nach Registrierung



Der Registrierungs-Bildschirm wird angezeigt. Der Nutzer gibt seine Zugangsdaten ein und bestätigt diese.



Der Login-Bildschirm öffnet sich. Der Nutzer gibt erneut die vorher gewählten Zugangsdaten ein und bestätigt diese.

⋮

In jedem Fall müssen aus der Dokumentation eines Testfalls die Vorbedingung (was war der Ausgangszustand der App), der Ablauf des Testfalls (welche Eingaben sind erfolgt; welche Rückmeldungen/Änderungen der App waren im Verlauf zu beobachten) das erwartete Verhalten (was hätte gemäß der Anforderungen passieren müssen) und natürlich das tatsächliche Verhalten hervorgehen. Zudem sollte jeder Testfall einen sprechenden Titel besitzen, der auf einen Blick beschreibt, welche Funktionalität in ihm getestet wird.