
Kleiderforum

Modul	335
Datum	16.08.2024
Durchführungsort	ICT LearnFactory
Lerncoach	Mateusz
Projektmitarbeitende	Prashanth Lavan

1 Ausgangslage

Es gibt bislang keine bekannte App, die sich speziell auf den Austausch von Kleidung fokussiert. Eine solche App, welche es ermöglicht, dass Nutzer ihre besten Kleidungsstücke posten und sich darüber unterhalten können.

2 Zielgruppenanalyse

2.1 Ziele

Unser Projekt zielt darauf ab, ein umfassendes und interaktives Kleiderforum zu schaffen, das Modebegeisterte aus aller Welt zusammenbringt. Das Hauptziel dieses Forums ist es, eine Plattform zu bieten, auf der Nutzer ihre Leidenschaft für Mode teilen, sich inspirieren lassen und voneinander lernen können.

2.2 Personas

Zielgruppe

Die Zielgruppe umfasst hauptsächlich modebewusste Menschen, die an Nachhaltigkeit und alternativen Konsum-Konzepten interessiert sind, dies umfasst bereits einen grossen Teil der Menschheit. Besonders angesprochen werden sollen:

1. **Millennials (25-40 Jahre):** Diese Altersgruppe ist in der Regel experimentierfreudig und legt grossen Wert auf Individualität und Selbstentfaltung.
2. **Gen Z (16-24 Jahre):** Die jüngere Generation, die stark auf soziale Medien und Trends fokussiert ist und eine Vorliebe für Kleider-Mode entwickelt hat.
3. **Fashion-Influencer und -Blogger:** Personen, die regelmässig ihre Outfits teilen und an neuen Möglichkeiten interessiert sind, ihren Stil zu präsentieren und ihre Garderobe zu erweitern.

Personas

1. Anna, 28 Jahre, Marketing-Managerin

- **Hintergrund:** Anna lebt in einer Grossstadt, arbeitet in der Marketingbranche und hat ein grosses Interesse an Mode. Sie nutzt regelmässig soziale Medien, um sich über aktuelle Modetrends zu informieren, und ist immer auf der Suche nach neuen, einzigartigen Stücken für ihren Kleiderschrank.

- **Ziele:** Anna möchte ihre Garderobe regelmässig auffrischen, ohne ständig zufällige Kleidung kaufen zu müssen. Sie ist an einer Community interessiert, in der sie Gleichgesinnte findet.

- **Herausforderungen:** Anna hat wenig Zeit, in Läden zu stöbern, und ist daher auf digitale Plattformen angewiesen, die ihr einen schnellen und einfachen Zugang zu neuen Outfit-Inspirationen ermöglichen.

2. Max, 22 Jahre, Student

- **Hintergrund:** Max studiert Umweltwissenschaften und legt Wert auf Nachhaltigkeit. Er liebt Mode, hat jedoch ein begrenztes Budget. Er ist sehr aktiv in sozialen Medien und interessiert sich für innovative Apps.

- **Ziele:** Max möchte coole und einzigartige Kleidungsstücke finden, die seinen Stil unterstreichen, ohne dafür viel Geld auszugeben. Ausserdem möchte er durch den Austausch von Kleidungsstilen einen Beitrag zu einem nachhaltigeren Konsumverhalten leisten, durch die Reduktion vom Kauf von ungenutzten Kleidung.

- **Herausforderungen:** Max sucht nach einer Plattform, die einfach zu bedienen ist und die Möglichkeit bietet, Kleider-Inspirationen zu finden ohne grossen Aufwand.

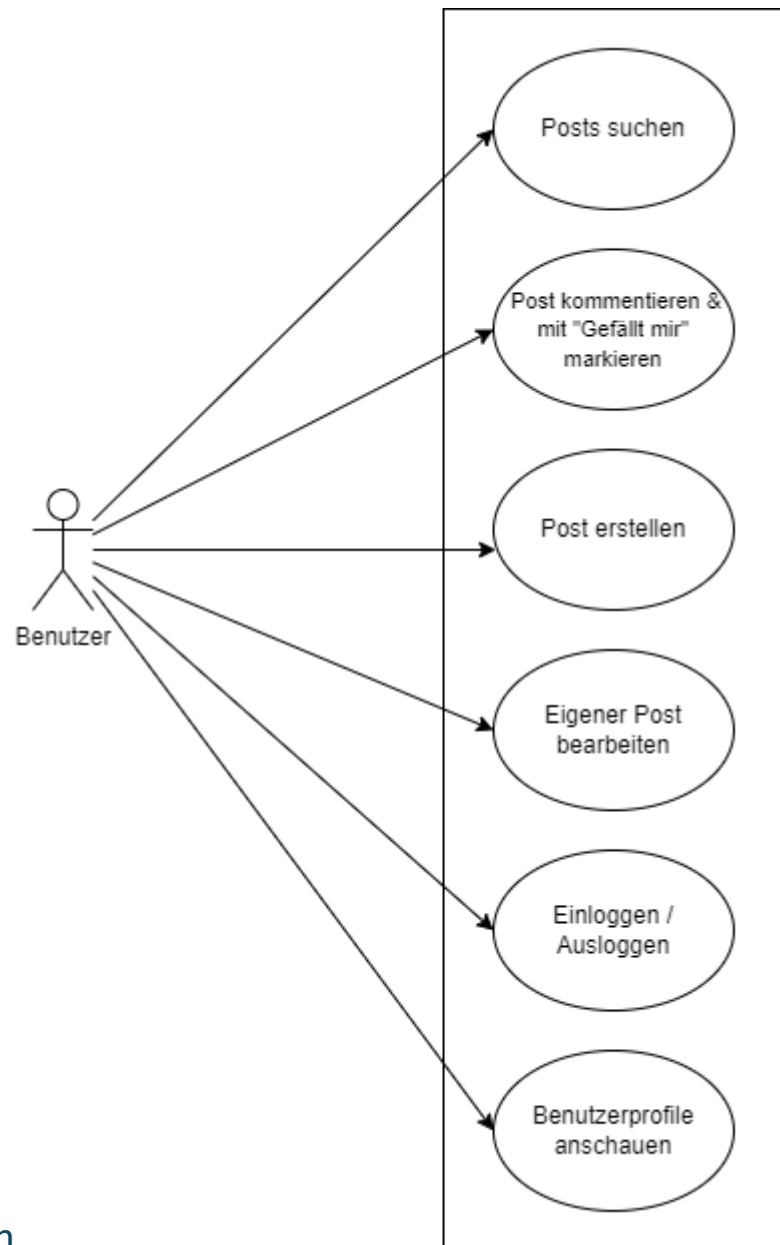
3. Lisa, 34 Jahre, Fashion-Bloggerin

- **Hintergrund:** Lisa ist eine erfolgreiche Mode-Bloggerin, die regelmässig ihre Outfits auf Instagram und ihrem Blog teilt. Sie ist immer auf der Suche nach neuen und einzigartigen Kleidungsstücken, um ihre Follower zu inspirieren.

- **Ziele:** Lisa möchte ihren Followern zeigen, dass es viele coole unentdeckte Kleidungsstücke gibt. Sie ist daran interessiert, ihre Garderobe zu erweitern und somit mehr posten zu können.

- **Herausforderungen:** Lisa sucht eine Plattform, auf der sie ihren Kleidungsstil einfach mit anderen austauschen kann.

3 Use Cases



3.1 Use Case Diagramm

3.2 Szenarien

Aktor: Benutzer

Ausgangslage: Benutzer möchte sich im Forum einloggen oder ausloggen

Szenario 1: Erfolgreiches Einloggen

1. Benutzer navigiert zur Login-Seite.

Der Benutzer klickt auf die Schaltfläche "Einloggen" oder "Login".

2. Benutzer gibt Anmeldedaten ein.

Der Benutzer gibt seinen Benutzernamen und sein Passwort ein.

3. Benutzer klickt auf die Login-Schaltfläche.

Der Benutzer klickt auf die Schaltfläche "Einloggen" oder "Login".

4. Benutzer wird eingeloggt.

Das System überprüft die Anmeldedaten und loggt den Benutzer ein.

5. Benutzer wird zur Startseite weitergeleitet.

Der Benutzer wird zur Startseite weitergeleitet.

Aktor: Benutzer

Ausgangslage: Benutzer hat einen interessanten Post gefunden

Szenario 2: Erfolgreiches Kommentieren und Markieren eines Posts

1. Benutzer liest den Post.

Der Benutzer öffnet den Post und liest den Inhalt.

2. Benutzer klickt auf die "Gefällt mir"-Schaltfläche.

Der Benutzer klickt auf die Schaltfläche "Gefällt mir", um den Post zu markieren.

3. Benutzer klickt auf die Kommentarfunktion.

Der Benutzer klickt auf die Schaltfläche oder das Textfeld zum Kommentieren.

4. Benutzer gibt einen Kommentar ein.

Der Benutzer schreibt seinen Kommentar in das Textfeld.

5. Benutzer veröffentlicht den Kommentar.

Der Benutzer klickt auf die Schaltfläche "Posten" oder "Veröffentlichen".

Aktor: Benutzer

Ausgangslage: Benutzer möchte einen neuen Post erstellen

Szenario 3: Erfolgreiches Erstellen eines Posts

1. Benutzer navigiert zur Post-Erstellungsseite.

Der Benutzer klickt auf die Schaltfläche "Neuer Post" oder eine ähnliche Option.

2. Benutzer gibt den Inhalt des Posts ein.

Der Benutzer gibt einen Titel und den Inhalt des Posts ein.

3. Benutzer fügt optional Medien hinzu.

Der Benutzer kann Bilder oder Videos hinzufügen, um den Post zu ergänzen.

4. Benutzer veröffentlicht den Post.

Der Benutzer klickt auf die Schaltfläche "Veröffentlichen" oder "Posten".

5. Post wird im Forum angezeigt.

Der neue Post wird im Forum sichtbar und andere Benutzer können ihn sehen und darauf reagieren.2w

4 User Stories

User Story 1: Post erstellen (essential)

Als Benutzer möchte ich in der Lage sein, Posts zu erstellen, die meine Kleidungsstücke zeigen, damit ich diese mit der Community teilen kann, um Inspiration und Feedback zu erhalten.

Akzeptanzkriterien:

- Ein Post kann Bilder enthalten.
- Der Post kann eine Beschreibung des Kleidungsstücks enthalten.
- Der Post wird erfolgreich in meinem Profil und im allgemeinen Feed der Community angezeigt.

User Story 2: Posts kommentieren und mit „Gefällt mir“ markieren (Wichtigkeit: 2/5)

Als Benutzer möchte ich Posts anderer Nutzer kommentieren und mit „Gefällt mir“ markieren können, damit ich meine Meinung teilen und mein Interesse an bestimmten Kleidungsstücken ausdrücken kann.

Akzeptanzkriterien:

- Kommentare können in Textform unterhalb des Posts hinzugefügt werden.

- Die „Gefällt mir“-Funktion ist durch ein Herz- oder Daumen-Symbol repräsentiert.
- Die Anzahl der „Gefällt mir“-Markierungen wird auf dem Post angezeigt.
- Ich kann meine „Gefällt mir“-Markierung jederzeit zurücknehmen.

User Story 3: Benutzerprofile anschauen (Wichtigkeit: 1/5)

Als Benutzer möchte ich die Profile anderer Benutzer ansehen können,
damit ich deren Kleidungsstil kennenlernen und mich inspirieren lassen kann.

Akzeptanzkriterien:

- Benutzerprofile zeigen die veröffentlichten Posts, die Anzahl der „Gefällt mir“-Markierungen und Kommentare an.
- Das Profil enthält eine kurze Biografie oder Beschreibung des Benutzers.

User Story 4: Eigene Posts bearbeiten (Wichtigkeit: 3/5)

Als Benutzer möchte ich meine eigenen Posts bearbeiten können,
damit ich Änderungen vornehmen kann.

Akzeptanzkriterien:

- Der Benutzer kann Elemente eines Posts bearbeiten.
- Änderungen werden sofort im Feed und im Profil aktualisiert.

User Story 5: Posts suchen (Wichtigkeit: 4/5)

Als Benutzer möchte ich nach bestimmten Posts suchen können,
damit ich schnell Kleidungsstücke finde, die meinen Interessen entsprechen.

Akzeptanzkriterien:

- Die Suche unterstützt Keywords und Benutzernamen.
- Die Suchergebnisse zeigen den Post oder Benutzer an.

User Story 6: Einloggen / Ausloggen (essential)

Als Benutzer möchte ich mich in die App einloggen und ausloggen können,
damit ich auf meine persönlichen Daten und Funktionen zugreifen kann.

Akzeptanzkriterien:

- Der Benutzer kann sich mit einer E-Mail-Adresse oder Nutzernamen und einem Passwort einloggen.
- Nach erfolgreichem Einloggen wird der Benutzer auf das Dashboard weitergeleitet.
- Der Benutzer kann sich jederzeit ausloggen.

User Story 7: Eigene Posts löschen (Wichtigkeit: 4/5)

Als Benutzer möchte ich in der Lage sein, meine eigenen Posts zu löschen,
damit ich nicht mehr benötigte Inhalte entfernen kann.

Akzeptanzkriterien:

- Der Benutzer kann einen Post mit einem Klick auf einen Button entfernen.
- Der gelöschte Post wird sofort aus dem Feed und dem Profil entfernt.

5 Mockups

Register

Benutzername

E-Mail

Passwort

Register

Sign In



Start



Profil

Login

Benutzername / E-Mail

Passwort

Login

Register



Start



Profil

Feedback & Fragen

Ankündigungen

Text Text Text Text Text Text Text

Feedback & Fragen

Text Text Text Text Text Text Text

Diskussionen

Luxus-Marken

Text Text Text Text Text Text Text

Styling-Tipps

Text Text Text Text Text Text Text

Undercover Brands

Text Text Text Text Text Text Text

Outfit Inspiration

Text Text Text Text Text Text Text



Start



Profil

Home > Diskussionen > Styling-Tipps

Neuer Post

Saisonale Mode

Frühling-Trends

Text Text Text Text Text Text Text

T-Shirt korrekt stylen

Text Text Text Text Text Text Text

Diskussionen

Was ist eure Meinung zu monochromen Outfits?

Von Test-User, erstellt gestern

Was ist eure Meinung zu monochromen Outfits?

Von Test-User, erstellt gestern

Was ist eure Meinung zu monochromen Outfits?

Von Test-User, erstellt gestern



Start



Profil

Home > Diskussionen > Styling-Tipps

Was ist eure Meinung zu monochromen Outfits?

Von Test-User, 10 Stunden her

Geschrieben von Test-User

Underrated

Lorem ipsum odor amet, consectetur adipiscing elit.
Lobortis semper vulputate risus a <semper lacinia. Lorem



Geschrieben von User

Overhated

Lorem ipsum odor amet, consectetur adipiscing elit.
Lobortis semper vulputate risus a <semper lacinia. Lorem



Geschrieben von User

Finde ich basic

Lorem ipsum odor amet, consectetur adipiscing elit.
Lobortis semper vulputate risus a <semper lacinia. Lorem



Start



Profil

Erstelle einen neuen Post im Thema Styling-Tipps

Titel:

Inhalt:

Abbrechen

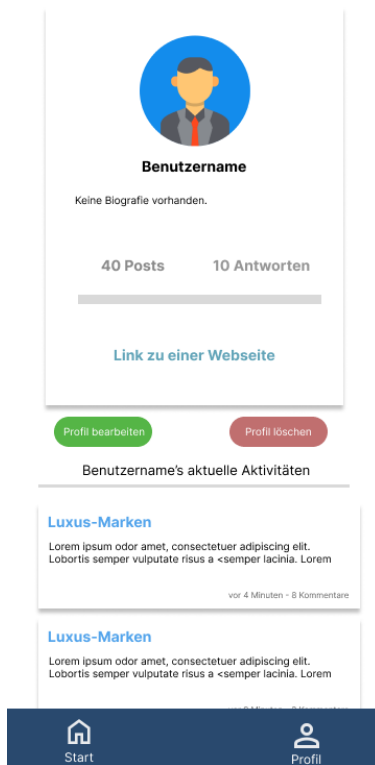
Veröffentlichen



Start



Profil



6 Special Feature

Unsere App bietet auch besondere Funktionen, die die Nutzung der Smartphone-Kamera erleichtern und die Benutzererfahrung verbessern. Erstens können Nutzer ganz einfach ein Profilbild hochladen, indem sie die Kamera direkt in der App verwenden, um ein Foto aufzunehmen und es sofort als Profilbild festzulegen. Dies macht den Prozess schnell und unkompliziert.

Zweitens ermöglicht die App das Hochladen von Bildern von Kleidungsstücken als Posts. Nutzer können die Kamera nutzen, um Fotos ihrer Kleidungsstücke aufzunehmen und diese direkt als Beiträge in ihrem Feed oder in anderen relevanten Bereichen der App zu teilen. Diese Funktionen sorgen dafür, dass Nutzer ihre Mode und Inspirationen bequem und intuitiv präsentieren können, ohne die App verlassen zu müssen.

7 Persistenz

Da unsere App vollständig auf eine Cloud-basierte Speicherung setzt, werden sämtliche Daten zentral in der Cloud gespeichert und verwaltet. Dies umfasst sowohl strukturierte Daten als auch grosse Dateien wie Bilder. Hier sind die spezifischen Überlegungen zu den verschiedenen Arten von Daten und deren Persistierung:

- Datenbank (SQLite in der Cloud)
- Art der Daten: Nutzerprofile, Post-Inhalte, Metadaten (z. B. Likes, Kommentare).
- Technologie: SQLite-Datenbank, die in einer Cloud-Umgebung gehostet wird
- Gründe: SQLite ist eine leichtgewichtige, aber leistungsfähige relationale Datenbank, die sich gut für die Anforderungen unserer App eignet. Da alle Daten zentral in der Cloud gespeichert werden, ist eine SQLite-Datenbank, die auf einem Cloud-Server läuft, eine effiziente Lösung, um sowohl die Struktur als auch die Integrität der Daten zu gewährleisten.
- Speicherung: Die SQLite-Datenbank wird auf einem Cloud-Server gehostet und verwaltet, was globale Verfügbarkeit, Skalierbarkeit und Datensicherheit gewährleistet.

8 Testkonzept

8.1 Testobjekt

Das Testobjekt ist unsere Applikation **KleiderForum**. Sie bietet Nutzern die Möglichkeit, Beiträge zu erstellen, Bilder hochzuladen, Kommentare abzugeben und mit anderen Nutzern zu interagieren. Die App nutzt lokale Speicherung über SQLite, Bildverarbeitung über ImagePicker und bietet eine benutzerfreundliche Oberfläche, die in React Native entwickelt wurde.

8.2 Testumgebung

Die Testumgebung besteht aus einem **Android Emulator**, der auf **Android Studio** konfiguriert ist. Als Referenzgerät wird ein **Pixel 5 Emulator** verwendet. Dies ermöglicht eine konsistente und kontrollierte Testumgebung, in der die Funktionalitäten der App überprüft werden können.

- **Emulator:** Android Studio Emulator
- **Gerätemodell:** Pixel 5
- **Betriebssystem:** API 35
- **Entwicklungsplattform:** React Native / Expo
- **Testmethoden:** Manuelle und automatisierte Tests zur Überprüfung der Funktionalitäten der Applikation.

8.3 Testfälle

Unit Tests

ID	T-01
User Story	US-01 - Als Benutzer möchte ich mich in die App einloggen können, um auf meine personalisierten Inhalte zugreifen zu können.
Testziel	Überprüfen, ob die LoginScreen-Komponente die Benutzeranmeldungen korrekt verarbeitet, einschließlich

	erfolgreicher Anmeldungen und Fehlerbehandlungen bei falschen Eingaben.
Voraussetzungen	<ul style="list-style-type: none"> - Die LoginScreen-Komponente ist in das Projekt integriert. - Die SQLite-Datenbank ist eingerichtet und enthält Benutzerdaten. - AsyncStorage ist korrekt implementiert, um Benutzerinformationen zu speichern. - Die Alert-Komponente ist vorhanden, um Benutzer über den Anmeldestatus zu informieren.
Ablauf	<ol style="list-style-type: none"> 1. Die LoginScreen-Komponente wird gerendert. 2. Der Benutzer gibt gültige Anmeldeinformationen (Benutzername/Email und Passwort) ein. 3. Die Datenbank wird abgefragt, um den Benutzer zu authentifizieren. 4. Die Benutzer-ID und der Benutzername werden in AsyncStorage gespeichert. 5. Eine Erfolgsmeldung wird angezeigt.
Erwartetes Ergebnis	Der Benutzer wird erfolgreich eingeloggt und die Anmeldeinformationen werden in AsyncStorage gespeichert. Eine Erfolgsmeldung wird angezeigt.

ID	T-02
User Story	US-02 - Als Benutzer möchte ich mich registrieren können, um ein Konto in der App zu erstellen
Testziel	Überprüfen, ob die SettingsScreen-Komponente die Benutzerregistrierung korrekt verarbeitet, einschließlich erfolgreicher Registrierungen und Fehlerbehandlungen bei bereits existierenden Benutzern.
Voraussetzungen	<ul style="list-style-type: none"> - Die SettingsScreen-Komponente ist in das Projekt integriert. - Die SQLite-Datenbank ist eingerichtet und kann Benutzerdaten speichern. - Die Alert-Komponente ist vorhanden, um Benutzer über den Registrierungsstatus zu informieren.
Ablauf	<ol style="list-style-type: none"> 1. Die SettingsScreen-Komponente wird gerendert. 2. Der Benutzer gibt neue Registrierungsinformationen (Email, Benutzername, Passwort) ein. 3. Die Datenbank wird abgefragt, um zu prüfen, ob der Benutzer bereits existiert. 4. Wenn der Benutzer noch nicht existiert, wird er der Datenbank hinzugefügt. 5. Eine Erfolgsmeldung wird angezeigt.

Erwartetes Ergebnis	Der Benutzer wird erfolgreich registriert und die Daten werden in der SQLite-Datenbank gespeichert. Eine Erfolgsmeldung wird angezeigt.
----------------------------	---

ID	T-03
User Story	US-03 - Als Benutzer möchte ich in der Lage sein, einen neuen Blogpost zu erstellen, um Inhalte mit anderen Benutzern zu teilen.
Testziel	Überprüfen, ob ein Benutzer erfolgreich einen neuen Blogpost erstellen kann, der Titel, Inhalt und ein Bild enthält, und ob die Daten korrekt in der Datenbank gespeichert werden.
Voraussetzungen	<ul style="list-style-type: none"> - Die CreatePostScreen-Komponente ist in das Projekt integriert. - Die SQLite-Datenbank ist eingerichtet und funktionsfähig. - Das Gerät hat Zugriff auf die Bildergalerie. - Ein Benutzer ist angemeldet und die userId ist im AsyncStorage gespeichert.
Ablauf	<ol style="list-style-type: none"> 1. Die CreatePostScreen-Komponente wird gerendert. 2. Der Benutzer gibt den Titel des Blogposts in das Titel-Eingabefeld ein. 3. Der Benutzer gibt den Inhalt des Blogposts in das Inhalts-Eingabefeld ein. 4. Der Benutzer wählt ein Bild aus der Bildergalerie aus. 5. Der Benutzer klickt auf die Schaltfläche "Veröffentlichen". 6. Der Test überprüft, ob der SQL-Befehl zum Einfügen des neuen Blogposts mit den korrekten Werten an die SQLite-Datenbank gesendet wurde.
Erwartetes Ergebnis	<ul style="list-style-type: none"> - Der Benutzer kann erfolgreich einen neuen Blogpost erstellen. - Der SQL-Befehl INSERT INTO blogs wird mit den korrekten Werten für Titel, Inhalt, Bild-URI und user_id ausgeführt. - Der Blogpost wird in der Datenbank gespeichert und eine Erfolgsmeldung wird angezeigt.

ID	T-04
User Story	US-04 - Als Benutzer möchte ich alle verfügbaren Blogs im Forum sehen, um die Inhalte anderer Benutzer zu lesen und zu interagieren.
Testziel	Überprüfen, ob die ForumScreen-Komponente die Blogs korrekt aus der SQLite-Datenbank lädt und anzeigt.
Voraussetzungen	<ul style="list-style-type: none"> - Die ForumScreen-Komponente ist in das Projekt integriert. - Die SQLite-Datenbank ist eingerichtet und enthält Blogeinträge. - Der Benutzer ist angemeldet und die userId ist im AsyncStorage gespeichert.

Ablauf	1. Die ForumScreen-Komponente wird gerendert. 2. AsyncStorage wird aufgerufen, um die userId des aktuell angemeldeten Benutzers abzurufen. 3. Die SQLite-Datenbank wird abgefragt, um alle Blogs zu laden, die im Forum verfügbar sind. 4. Der Test überprüft, ob die Blogs korrekt aus der Datenbank geladen und auf dem Bildschirm angezeigt werden.
Erwartetes Ergebnis	- Die userId wird korrekt aus dem AsyncStorage abgerufen. - Die Blogs werden korrekt aus der SQLite-Datenbank geladen. - Die geladenen Blogs werden auf dem Bildschirm angezeigt.

Snapshot Tests

ID	T-05
User Story	US-05
Testziel	Überprüfung, dass die CreatePostScreen-Komponente korrekt gerendert wird und keine unbeabsichtigten Änderungen am UI erfolgen.
Voraussetzungen	- Die Komponente CreatePostScreen muss implementiert und in der App verfügbar sein. - Die Testumgebung muss eingerichtet sein, einschliesslich der Installation von jest, react-test-renderer, @react-native-async-storage/async-storage, expo-sqlite, und expo-image-picker.
Ablauf	1. Mocking: Vor dem Rendern der Komponente werden verschiedene Module und Funktionen gemockt: <ul style="list-style-type: none"> • @react-native-async-storage/async-storage: Die Methoden getItem, setItem, und removeItem werden gemockt. • expo-sqlite: Die Methode openDatabase und die execSync Funktion werden gemockt. • expo-image-picker: Die Methoden launchImageLibraryAsync und requestMediaLibraryPermissionsAsync werden gemockt. 2. Rendering: Die CreatePostScreen-Komponente wird gerendert und das resultierende JSON-Objekt wird mit einem gespeicherten Snapshot verglichen. 3. Vergleich: Der Test vergleicht das aktuelle gerenderte JSON-Objekt mit dem gespeicherten Snapshot. Wenn es Abweichungen gibt, schlägt der Test fehl.
Erwartetes Ergebnis	Der Snapshot-Test soll erfolgreich sein und bestätigen, dass die CreatePostScreen-Komponente korrekt gerendert wurde. Es dürfen keine unbeabsichtigten UI-Änderungen aufgetreten sein.

ID	T-06
User Story	US-06
Testziel	Überprüfung, dass die ProfileScreen-Komponente korrekt gerendert wird und keine unbeabsichtigten Änderungen am UI erfolgen.
Voraussetzungen	<ul style="list-style-type: none"> - Die Komponente ProfileScreen muss implementiert und in der App verfügbar sein. - Die Testumgebung muss eingerichtet sein, einschliesslich der Installation von jest, react-test-renderer, @react-native-async-storage/async-storage, expo-sqlite, und expo-image-picker.
Ablauf	<p>1. Mocking: Vor dem Rendern der Komponente werden verschiedene Module und Funktionen gemockt:</p> <ul style="list-style-type: none"> • @react-native-async-storage/async-storage: Die Methoden getItem, setItem, und removeItem werden gemockt. • expo-sqlite: Die Methode openDatabase und die transaction Funktion werden gemockt. • @react-navigation/native: Die useNavigation Funktion wird gemockt, um Navigationsoperationen zu verhindern. • expo-image-picker: Die Methoden launchCameraAsync und requestCameraPermissionsAsync werden gemockt. <p>2. Rendering: Die ProfileScreen-Komponente wird gerendert und das resultierende JSON-Objekt wird mit einem gespeicherten Snapshot verglichen.</p> <p>3. Vergleich: Der Test vergleicht das aktuelle gerenderte JSON-Objekt mit dem gespeicherten Snapshot. Wenn es Abweichungen gibt, schlägt der Test fehl.</p>
Erwartetes Ergebnis	Der Snapshot-Test soll erfolgreich sein und bestätigen, dass die ProfileScreen-Komponente korrekt gerendert wurde. Es dürfen keine unbeabsichtigten UI-Änderungen aufgetreten sein.

9 Umsetzung

9.1 Libraries

```

"@expo/vector-icons": "^14.0.2",
"@react-native-async-storage/async-storage": "^2.0.0",
"@react-navigation/native": "^6.1.18",
"@react-navigation/native-stack": "^6.11.0",
"@react-navigation/stack": "^6.4.1",
"expo": "~51.0.28",
"expo-constants": "~16.0.2",
"expo-font": "~12.0.9",
"expo-image-picker": "~15.0.7",
"expo-linking": "~6.3.1",
"expo-router": "~3.5.23",

```

```
"expo-splash-screen": "~0.27.5",  
"expo-sqlite": "~14.0.6",  
"expo-status-bar": "~1.12.1",  
"expo-system-ui": "~3.0.7",  
"expo-web-browser": "~13.0.3",  
"react": "18.2.0",  
"react-dom": "18.2.0",  
"react-native": "0.74.5",  
"react-native-gesture-handler": "~2.16.1",  
"react-native-image-picker": "^7.1.2",  
"react-native-reanimated": "~3.10.1",  
"react-native-safe-area-context": "^4.10.5",  
"react-native-screens": "^3.31.1",  
"react-native-sqlite-storage": "^6.0.1",  
"react-native-web": "~0.19.10"
```

Wir haben in unserem Projekt verschiedene Libraries verwendet. Für unser Special Feature haben wir die Library „*expo-image-picker*“ genutzt, da sie es ermöglicht, Bilder hochzuladen und aufzunehmen.

Eine weitere, für uns sehr wichtige Library ist „*expo-sqlite*“. Diese Library erlaubt es uns, unsere Daten lokal zu speichern.

Die wichtigste Library ist natürlich „*expo*“, die wir nutzen, um eine effiziente Entwicklungsumgebung für unsere App zu schaffen.

9.2 ...

Wir dachten, dass wir zeitlich sehr gut im Plan lagen, doch je länger wir am Projekt arbeiteten, desto mehr zog sich die Umsetzung in die Länge. Obwohl unsere praktische Umsetzung etwas langsamer voranging, konnten wir dennoch sehr gut mit GitHub arbeiten und effektiv kommunizieren, wer welche Aufgaben übernimmt.

9.3 Abweichungen zum Konzept

Während der Programmierung haben wir gemerkt, dass es viel mehr Arbeit ist, als wir im ursprünglichen Mockup erwartet hatten. Die tatsächlichen Anforderungen waren umfangreicher, und wir mussten zusätzliche Aufgaben erledigen, um die App so umzusetzen, wie wir es geplant hatten. Die Komplexität und die Details waren im ursprünglichen Konzept nicht vollständig erfasst. Deshalb konnten wir unser Projekt nicht zum vollen Glanz bringen.

9.4 Testprotokoll

9.4.1 Testlauf 1

Datum und Zeit: 30.08.2024. 16:42

Testperson: Terence

ID	Resultat	Bemerkung	Massnahme
----	----------	-----------	-----------

Drücken Sie auf den "Registrieren"-Button.	✓	-	-
Geben Sie eine E-Mail, einen Benutzernamen und ein Passwort in die entsprechenden Eingabefelder ein. Drücken Sie auf den "Register"-Button.	✓	-	
Drücken Sie oben links auf den "index"-Button.	✓	Man könnte noch anstatt «index» zurück schreiben	Falls noch Zeit erledigen
Geben Sie entweder die E-Mail oder den Benutzernamen in das Eingabefeld ein und darunter das Passwort. Drücken Sie auf den "Login"-Button.	✓		
Drücken Sie auf den "Neuer Post"-Button.	✓		
Geben Sie einen Titel in das Titelfeld ein und den Inhalt in das darunterliegende Inhaltsfeld.	✓		
Drücken Sie oben links auf den "(tabs)"-Button.	✓	Hier auch anstatt «(tabs)» kann man zurück schreiben.	Falls noch Zeit anpassen

9.5 Testfazit

Unsere Tests haben gezeigt, dass die App in den meisten Bereichen stabil und zuverlässig funktioniert. Die automatisierten Unit- und Integrationstests sorgten für eine gute Code Coverage, was uns Vertrauen in die Kernlogik der App gibt. Das manuelle End-to-End Testing half dabei, Usability-Probleme zu identifizieren, die in realen Benutzerinteraktionen auftreten könnten.

Insgesamt haben die Tests die Qualität der App erhöht und wertvolle Verbesserungsmöglichkeiten aufgezeigt.