

Studienarbeit

Programmmentwurf Mobile Computing

Kalenderapp

Fachbereich Elektrotechnik und Informationstechnik
An der Dualen Hochschule Baden-Württemberg
Center of Advanced Studies

Eingereicht von:
Ruslan Adilgereev
Nadine Abu El Komboz
Lukas Renz

Matrikelnummer:
7719665
5618268
4681850

Jahrgang
2023
2024
2024

Betreuer:

Prof. Dr. Kai Becher

Ehrenwörtliche Erklärung

Hiermit erkläre wir, dass die vorliegende Studienarbeit mit dem Thema

Programmentwurf Mobile Computing

selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Aus fremden Quellen direkt oder indirekt übernommene Gedanken habe ich als solche kenntlich gemacht. Die Arbeit habe wir bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher auch nicht veröffentlicht.

Heilbronn, 5. Dezember 2024

Nadine Abu El Komboz

Ruslan Adilgereev

Lukas Renz

Inhaltsverzeichnis

Inhaltsverzeichnis	III
1 Einleitung	1
2 Aufbau der Software	2
3 Bug und Revision	3
4 Testing	4
4.1 Unit-Test	4
4.2 Widget-Test	4
4.3 Implementierung	4
5 Bedienung des Benutzers	6
6 Fazit und Ausblick	7
A Anhang	8
A.1 ergebnisse Testing	8

1 Einleitung

In der folgenden Dokumentation wird die Implementierung einer Kalender-App beschrieben, mit der Trainingseinheiten gebucht werden können. Die Dokumentation erläutert, wie diese Anwendung konzipiert und umgesetzt wurde. Aus der Aufgabenstellung ergaben sich die folgenden Vorgaben für die Bearbeitung:

- Anbindung einer Datenbank mittels MariaDB,
- Implementierung einer Kalenderjahresansicht,
- Farbschema bestehend aus den Farben Blau, Grau und Weiß.

Zusätzlich zu diesen vorgegebenen Features konnten noch die folgenden Funktionen integriert werden:

- Startseite der App mit allgemeinen Informationen und rechtlichen Hinweisen,
- weitere Kalenderansichten (Jahr, Monat, Woche),
- Darstellung der Trainingseinheiten in einer Listenansicht (alle Trainings, gebuchte Trainings),
- Sortierfunktion für die Listenansichten,
- E-Mail-Bestätigung bei der Anmeldung,
- Implementierung von Anbieter- und Benutzerrollen:
 - Anbieter können Trainings und Trainer hinzufügen,
 - Benutzer können Trainings buchen.
- zusätzliches Farbschema im Dark Mode,
- Filterfunktionen (z. B. Suche nach Begriffen und Tags).

2 Aufbau der Software

3 Bug und Revision

4 Testing

Um die Kalender-App zu testen, wurden zwei Arten von Tests durchgeführt: Widget-Tests und Unit-Tests.

4.1 Unit-Test

Mithilfe eines Unit-Tests können einzelne Methoden und Funktionen überprüft werden. Im Test werden die erforderlichen Eingabeparameter simuliert und die Ausgabeparameter auf ihre Korrektheit hin überprüft. Ein Unit-Test greift während der Testdurchführung nicht auf die Festplatte zu. Dadurch werden weder Grafiken gerendert noch Benutzereingaben berücksichtigt.

Unit-Tests dienen daher ausschließlich dazu, die Logik einzelner Methoden zu überprüfen. Sie eignen sich jedoch nicht, um die Schnittstellenfunktionalität oder die Interaktion mit Benutzern zu testen.

4.2 Widget-Test

Der Widget-Test wird verwendet, um die Interaktion mit dem Benutzer und die grafische Darstellung der App zu überprüfen. In einer isolierten Testumgebung werden Widgets erzeugt und getestet, ob sie korrekt gerendert wurden. Darüber hinaus wird überprüft, ob Benutzereingaben korrekt verarbeitet werden und das erneute Rendering der Benutzeroberfläche ordnungsgemäß funktioniert.

4.3 Implementierung

Es wurden zwei Arten von Tests implementiert, um die Funktionalität der Kalender-App sicherzustellen:

- Unit-Tests, um die Logik einzelner Methoden innerhalb von Flutter zu überprüfen.
- Widget-Tests, um die Benutzerinteraktion und die korrekte grafische Darstellung der Anwendung zu validieren.

Der Code für die Testprogramme wurde in den folgenden Dateien umgesetzt: `"training_service_test.dart"`, `"training_service_test.mocks.dart"` und `"widget_test.dart"`. Bei `"training_service_test.dart"` und der Datei `"training_service_test.mocks.dart"` handelt es sich um den Unit-Test, wobei die Datei `"training_service_test.mocks.dart"` eine Simulationsumgebung umsetzt und die andere Datei den eigentlichen Test enthält. Durch die begrenzte Zeit wurde jeder Test jeweils für eine Methode implementiert. Mittels des Unit-Tests wurde die Funktion der Trainingsuche getestet. Dabei wurden die folgenden vier Szenarien betrachtet:

- Erfolgreiche Antworten (gültige Trainingsdaten).
- Fehlerhafte Antworten (Fehlerstatus oder ungültige JSON-Daten).
- Leere Antworten (keine Trainings gefunden).
- Formatierung von null-Werten oder fehlenden Daten.

Beim Widget-Test wurde die ordnungsgemäße Initialisierung der *ThemeService*- und *AuthService*-Provider getestet. Das Ergebnis des Testprotokolls befindet sich im Anhang.

5 Bedingung des Benutzers

6 Fazit und Ausblick

A Anhang

A.1 ergebnisse Testing

Flutter Application Test Report

Date: 2024-01-18

Overview

This report documents the test results for the Training Calendar Flutter application. The tests cover both unit testing of the TrainingService and widget testing of the main application.

Test Environment

- Framework: Flutter Test
- Testing Libraries:
 - flutter_test
 - mockito
 - build_runner

Unit Tests Results

TrainingService Tests

Location: test/training_service_test.dart

1. Search Functionality Test ¶

Description: Tests the normal search functionality for trainings

Status: PASSED

Verified:

- Correct API endpoint called
- Proper data transformation
- Expected response structure

2. Error Handling Test ¶

Description: Tests error handling for API failures

Status: PASSED

Verified:

- Exception thrown on 500 status code
- Proper error message propagation

3. Empty Response Test ¶

Description: Tests handling of empty search results

Status: PASSED

Verified:

- Empty list returned
- No exceptions thrown

4. Malformed Response Test ¶

Description: Tests handling of invalid JSON responses

Status: PASSED

Verified:

- Exception thrown on invalid JSON
- Proper error handling

5. Data Formatting Test ¶

Description: Tests null value handling and data formatting

Status: PASSED

Verified:

- Default values applied for null fields
- Proper data transformation

Widget Tests

App Initialization Test ¶

Location: test/widget_test.dart