

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, 6. Dezember 2024

Nadine Abu El Komboz

Ruslan Adilgereev

Lukas Renz

Inhaltsverzeichnis

Inhaltsverzeichnis	III
1 Einleitung	1
2 Ausführen 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 durch den Benutzer	6
5.1 Homescreen	6
5.2 Optionen Darstellung	6
5.2.1 Kalenderansicht	8
5.2.2 Schulungsinformationen	8
5.2.3 Anbieteransicht	10
6 Fazit und Ausblick	11
A Anhang	12
A.1 ergebnisse Testing	12

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 Ausführen der Software

Um die Software ausführen zu können, müssen sowohl Flutter als auch Dart installiert sein. Zudem müssen die in der *"pubspec.yaml"*-Datei angegebenen Pakete installiert werden. Dies erfolgt durch Ausführen des folgenden Befehls in der Konsole, während man sich im Projektordner befindet:

```
1 flutter pub get
```

Codeausschnitt 2.1 Installieren der benötigten Pakete

Für die Einbindung der MariaDB-Datenbank sind möglicherweise Anpassungen in der Datei *"flutter_backend/.env"* erforderlich, um den Port und den Host korrekt zu konfigurieren.

Zum Starten der App muss zunächst das Backend gestartet werden. Dies geschieht durch Ausführen des folgenden Befehls im Ordner *"flutter_backend"*:

```
1 npm run dev
```

Codeausschnitt 2.2 Starten des Backends

Um das Frontend zu starten, muss in einer weiteren Konsole der folgende Befehl ausgeführt werden, während man sich im Projektordner befindet. Dadurch wird das Frontend am richtigen Port gestartet:

```
1 flutter run --web-port 3000
```

Codeausschnitt 2.3 Starten des Frontends

Um sich als Anbieter in der Software anzumelden gibt es einen Admin account, welcher *admin* sowohl als Benutzer wie auch als Passwort verwendet.

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 Bedienung durch den Benutzer

5.1 Homescreen

Die Kalender-App wurde so konzipiert, dass sie nicht nur zur Buchung von Schulungen dient, sondern auch als Plattform genutzt werden kann, um sich als Unternehmen zu präsentieren. Die Startseite, die vor dem eigentlichen Buchungstool angezeigt wird, ist so gestaltet, dass sie allgemeine Informationen über die App bereitstellt und deren Nutzungsmöglichkeiten erläutert.

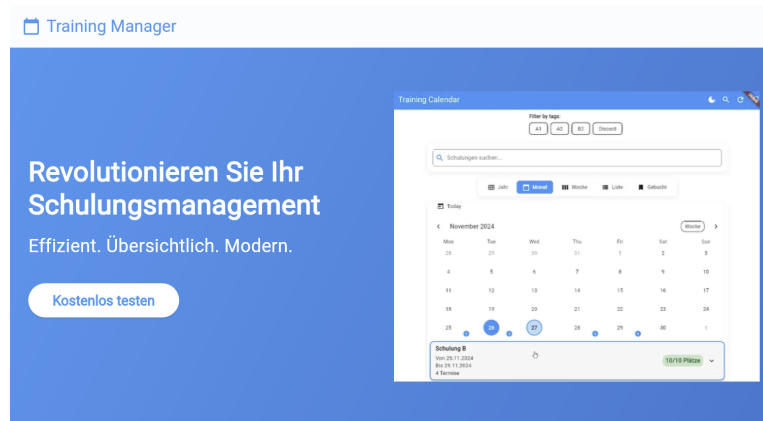
Sollte die App beispielsweise primär für Schulungen genutzt werden, könnte ein Unternehmen diese Seite dazu verwenden, sich vorzustellen. Hier könnten Informationen über das Unternehmen, seine Alleinstellungsmerkmale sowie das angebotene Leistungsspektrum bereitgestellt werden. Ebenso bietet die Startseite Platz für rechtliche Hinweise wie Datenschutzinformationen und ein Impressum, für das am unteren Seitenrand ein entsprechender Link eingebaut ist.

Von dieser Startseite gelangt der Benutzer über den Button *"Kostenlos testen"*, der in Abbildung 5.1 dargestellt ist, zur eigentlichen Kalenderübersicht. Diese bildet den zentralen Bestandteil des Buchungssystems.

5.2 Optionen Darstellung

Im Kalender wurden mehrere Optionen implementiert, die in der Abbildung 5.2 dargestellt sind. Eine dieser Optionen ist die blaue Leiste, die der Navigation zwischen den einzelnen Fenstern dient und Änderungen des Farbschemas ermöglicht. Über diese Funktion gelangt man zurück zum Homescreen, öffnet ein Fenster zur Suche, wechselt das Farbschema in den Dark Mode oder kann die Einstellungen des persönlichen Profils bearbeiten. Sollte der Nutzer noch nicht angemeldet sein, wird diese Leiste als Login-Fenster angezeigt.

Unterhalb der Navigationsleiste befindet sich eine Funktion, mit der Schulungen nach bestimmten Tags gefiltert werden können. In diesem Fall wurden als



Alles was Sie brauchen

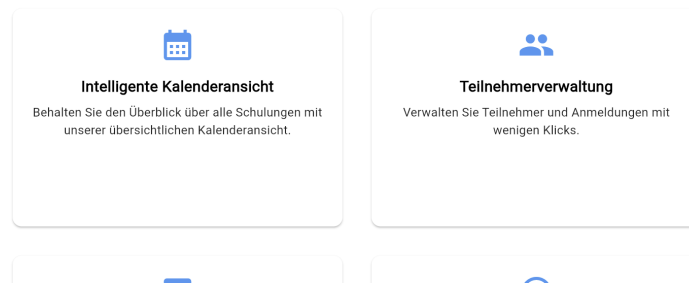


Abbildung 5.1 Startseite

Beispiele die Fachübergreifenden Kompetenzen des CAS gewählt. Bei anderen Schulungen, wie etwa Sicherheitsunterweisungen, könnte hier auch die dazugehörige ID angezeigt werden. Durch das Anklicken eines Tags wird eine Filterung nach diesem Kriterium vorgenommen.

Unter der Tag-Filterung ist eine Freitextsuche implementiert, die es ermöglicht, durch alle relevanten Informationen eines Trainings zu suchen. So kann nach dem Trainingsnamen, dem Startdatum oder auch dem Namen des Trainers gesucht werden. Sowohl die Filterung durch die Suche als auch die Tag-Filterung wird bei der Darstellung jeder Kalenderansicht berücksichtigt.

Eine weitere Option ermöglicht es, die Schulungen in verschiedenen Zeiträumen anzuzeigen. Der Nutzer kann zwischen den Ansichten „Jahr“, „Monat“ und „Woche“ wählen, wobei die Monatsansicht die Standarddarstellung darstellt. Neben der Darstellung im Kalender kann die Schulungsliste auch in Listenform angezeigt werden. In dieser Ansicht kann der Nutzer zwischen einer Filterung nach allen Schulungen oder nur nach den gebuchten Schulungen wählen.

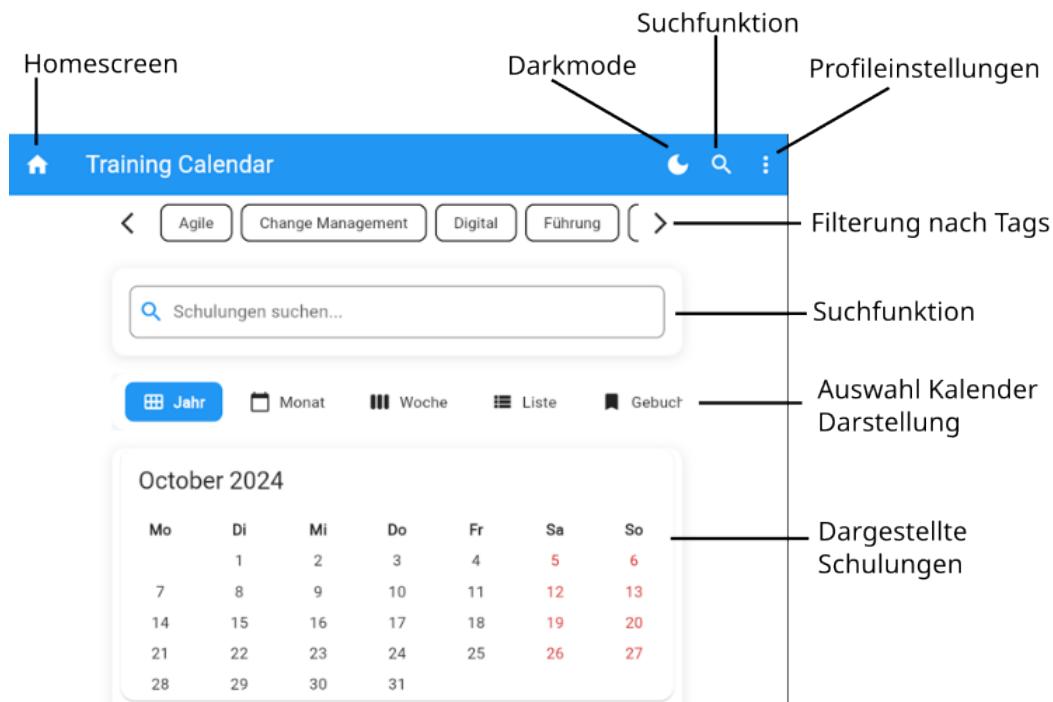


Abbildung 5.2 Option in der Kalenderansicht

5.2.1 Kalenderansicht

In der Kalenderansicht werden Tage, an denen Trainings angeboten werden, mit orangen und blauen Punkten markiert. Ein orangefarbener Punkt bedeutet, dass es sich um eine mehrtägige Schulung handelt, während ein blauer Punkt darauf hinweist, dass die Schulung nur an einem Tag stattfindet. In der Monats- und Wochenansicht sind zusätzlich Nummern in diesen Punkten enthalten, die angeben, wie viele Schulungen an diesem Tag angeboten werden. Dieses unterschiedliche Verhalten ist in der Abbildung 5.3 dargestellt. Es ist ersichtlich, dass in der Jahresansicht am 24. Dezember nur zwei Punkte angezeigt werden, während in der Monatsansicht zusätzlich Zahlen zu sehen sind, die die Anzahl der Schulungen an diesem Tag angeben. Ist eine Schulung an einem Tag gebucht, wird dieser Tag zusätzlich rot umrandet.

5.2.2 Schulungsinformationen

Um genauere Informationen zu erhalten, muss der Tag im Kalender angeklickt werden. Anschließend erscheint die Ansicht, die in Abbildung 5.4 dargestellt ist. Durch Scrollen kann man alle Schulungen durchsehen. Zudem erhält man

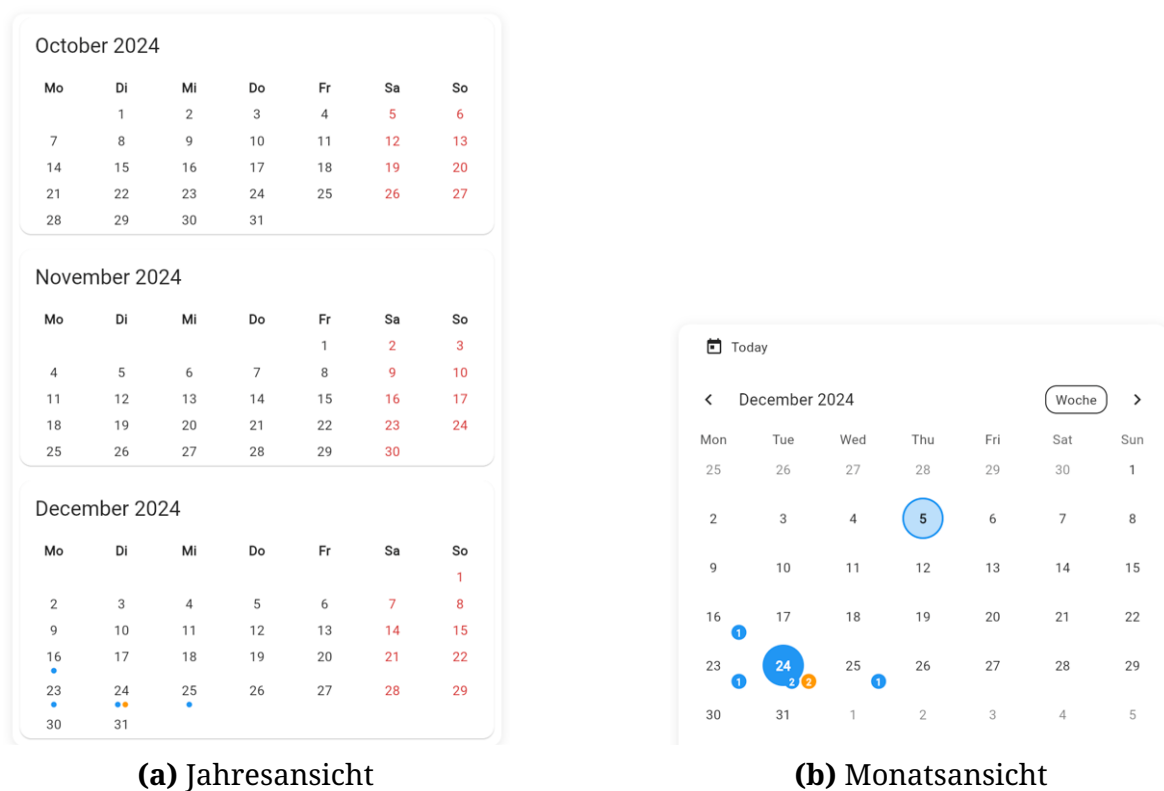


Abbildung 5.3 Darstellung der unterschiedlichen Markierungen der verfügbaren Schulungen

durch das Anklicken einer Schulung weitere Informationen, wie beispielsweise die Kontaktdaten des Dozenten, eine Beschreibung der Schulung sowie die Startzeiten.

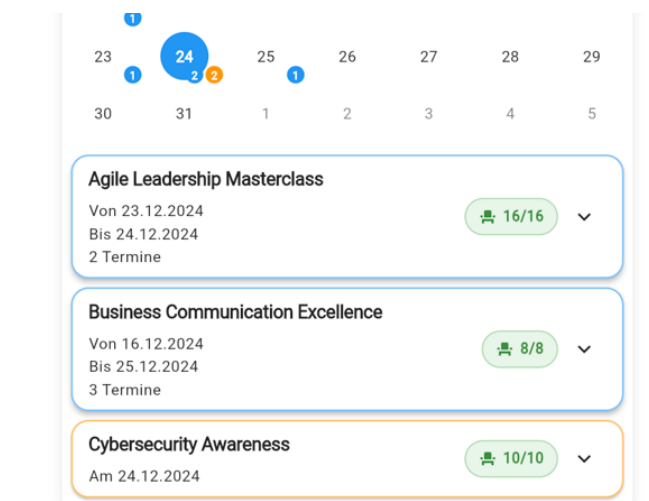
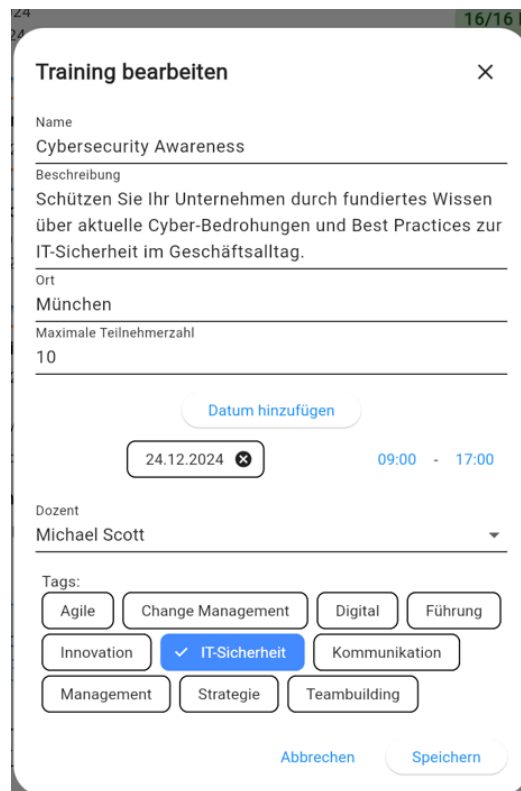


Abbildung 5.4 Informationen zu den Trainings

Entscheidet man sich, ein Training zu buchen, so kann man einen Haken setzen, um eine Buchungsbestätigung zu erhalten. Dadurch wird dem Benutzer eine E-Mail als Buchungsbestätigung zugesendet.

5.2.3 Anbieteransicht

Meldet man sich als Anbieter von Schulungen in der App an, so hat man die Möglichkeit, Trainings zu erstellen und diese zu bearbeiten. In Abbildung 5.5 sind die verschiedenen Beschreibungsmöglichkeiten für eine Schulung dargestellt. Die Auswahl des Datums kann dabei in einer Kalenderansicht erfolgen, in der der gesamte Zeitraum markiert wird, in dem die Schulung stattfindet.



Training bearbeiten ✕

Name
Cybersecurity Awareness

Beschreibung
Schützen Sie Ihr Unternehmen durch fundiertes Wissen über aktuelle Cyber-Bedrohungen und Best Practices zur IT-Sicherheit im Geschäftsalltag.

Ort
München

Maximale Teilnehmerzahl
10

Datum hinzufügen

24.12.2024 ✕ 09:00 - 17:00

Dozent
Michael Scott ▼

Tags:

Agile Change Management Digital Führung

Innovation **✓ IT-Sicherheit** Kommunikation

Management Strategie Teambuilding

Abbrechen Speichern

Abbildung 5.5 Bearbeiteransicht

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