

MoonFlow App Dokumentation

ENTWICKLUNG MOBILER ANWENDUNGEN (EMA)

HOCHSCHULE WORMS – DEPARTMENT INFORMATIK

INHALTSVERZEICHNIS

1. Einführung
2. Projektübersicht
3. Anforderungen
4. User Stories
5. Wireframes und Designentwicklung
6. Implementierungsdetails
7. Codequalität
8. Wartbarkeit
9. Entwicklungsstand und zukünftige Pläne
10. Fazit

1. Einführung

Dieses Dokument gibt einen Überblick über die Entwicklung und Testung der MoonFlow App, einer mobilen Anwendung zur Zyklusverfolgung und zum Austausch in einer Community, die im Rahmen des Kurses „Entwicklung mobiler Anwendungen (EmA)“ an der Hochschule Worms entstanden ist. Die App richtet sich an Frauen, die ihren Menstruationszyklus nachvollziehen und sich mit Gleichgesinnten austauschen möchten. Die Entwicklung erfolgte mit Flutter und Dart, wobei Firebase für sichere Authentifizierung, Datenspeicherung und Messaging genutzt wird. Besonderer Wert wurde auf ein diskretes Design, Benutzerfreundlichkeit und Datenschutz gelegt.

2. Projektübersicht

- Projektname: MoonFlow
- Kurs: Entwicklung mobiler Anwendungen (EmA)
- Institution: Hochschule Worms
- Entwickler: Auriol Sopning Tiotsop, Suzie Djouoko und Davina Daouda Mama
- Technologien: Flutter, Dart, Firebase

2.1 Projektidee

Moonflow ist eine plattformübergreifende App, die Nutzerinnen ermöglicht, ihren Menstruationszyklus zu verfolgen, Symptome zu dokumentieren und sich in einem Diskussionsforum mit anderen auszutauschen. Der Partner-Modus erlaubt es, den Zyklus diskret mit einer weiteren Person zu teilen. Die App legt Wert auf ein einfaches, intuitives Bedienkonzept und den Schutz sensibler Daten.

3. Anforderungen

3.1 Funktionale Anforderungen

1. Mehrere Bildschirme:

- Onboarding
- Splash Screen
- Registrierung/Login
- Passwort-Wiederherstellung
- Startbildschirm mit Zyklus kalender
- Symptom-Tracker
- Community-Diskussionsforum
- Partner-Modus Verwaltung
- Mehrsprachigkeit

2. Nutzergenerierte Inhalte:

- Erfassung und Bearbeitung von Zyklus- und Symptominformationen
- Verfassen und Lesen von Beiträgen im Community-Forum
- Bearbeitung von Profilbilder

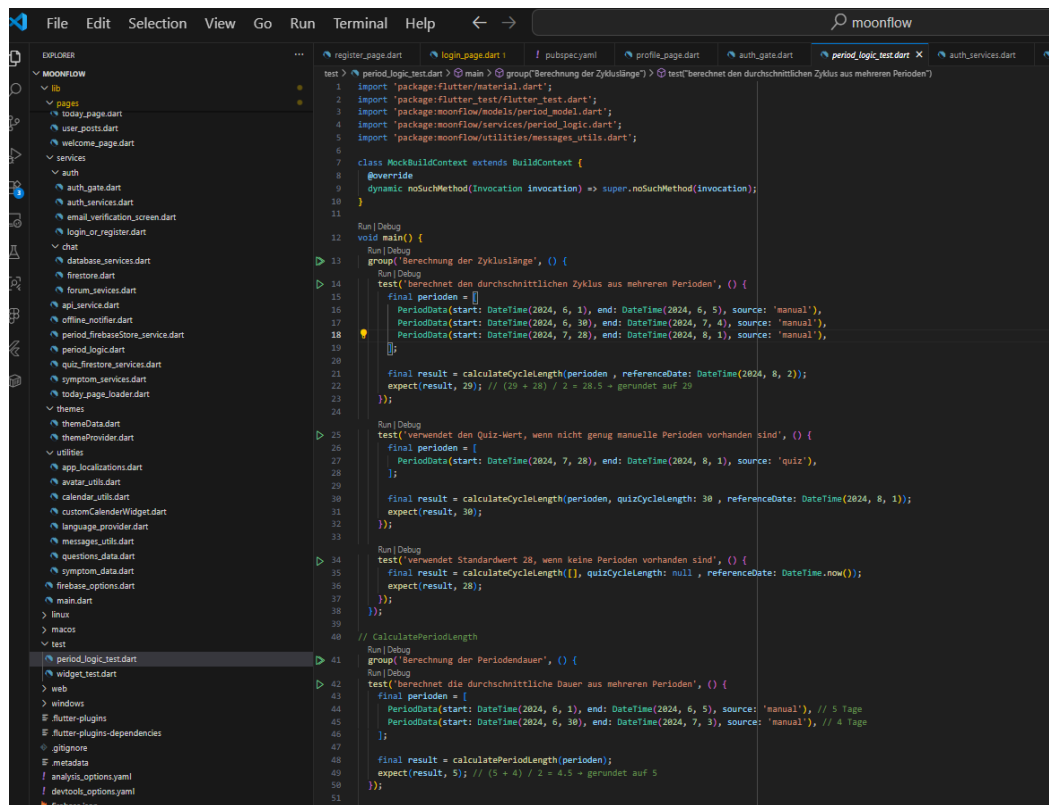
3. Lokale Datenspeicherung:

- Manche Fotos werden lokal gespeichert, um offline darauf zugreifen zu können.

4. Integration von Google Firebase:

- Firebase Authentication zur Nutzerregistrierung und Anmeldung.
- Firebase Firestore zur Speicherung von Ereignisdaten und Nutzergruppen.
- Firestore zur Speicherung von Fotos

5. Unit Test: Beispiel im Code



```
test > period_logic_test.dart > main > group('Berechnung der Zykluslänge') > test('berechnet den durchschnittlichen Zyklus aus mehreren Perioden')
1 import 'package:flutter/material.dart';
2 import 'package:flutter_test/flutter_test.dart';
3 import 'package:moonflow/models/period_model.dart';
4 import 'package:moonflow/services/period_logic.dart';
5 import 'package:moonflow/utilities/messages_utils.dart';
6
7 class MockBuildContext extends BuildContext {
8   @override
9   dynamic noSuchMethod(Invocation invocation) => super.noSuchMethod(invocation);
10 }
11
12 Run[Debug]
13 void main() {
14   Run[Debug]
15   group('Berechnung der Zykluslänge', () {
16     Run[Debug]
17     test('berechnet den durchschnittlichen Zyklus aus mehreren Perioden', () {
18       final perioden = [
19         PeriodData(start: DateTime(2024, 6, 1), end: DateTime(2024, 6, 5), source: 'manual'),
20         PeriodData(start: DateTime(2024, 6, 30), end: DateTime(2024, 7, 4), source: 'manual'),
21         PeriodData(start: DateTime(2024, 7, 28), end: DateTime(2024, 8, 1), source: 'manual'),
22       ];
23       final result = calculateCycleLength(perioden, referenceDate: DateTime(2024, 8, 2));
24       expect(result, 29); // (29 + 28) / 2 = 28.5 = gerundet auf 29
25     });
26
27     Run[Debug]
28     test('verwendet den Quiz-Wert, wenn nicht genug manuelle Perioden vorhanden sind', () {
29       final perioden = [
30         PeriodData(start: DateTime(2024, 7, 28), end: DateTime(2024, 8, 1), source: 'quiz'),
31       ];
32       final result = calculateCycleLength(perioden, quizCycleLength: 30, referenceDate: DateTime(2024, 8, 1));
33       expect(result, 30);
34     });
35
36     Run[Debug]
37     test('verwendet Standardwert 28, wenn keine Perioden vorhanden sind', () {
38       final result = calculateCycleLength([], quizCycleLength: null, referenceDate: DateTime.now());
39       expect(result, 28);
40     });
41   });
42
43   Run[Debug]
44   group('Berechnung der Periodendauer', () {
45     Run[Debug]
46     test('berechnet die durchschnittliche Dauer aus mehreren Perioden', () {
47       final perioden = [
48         PeriodData(start: DateTime(2024, 6, 1), end: DateTime(2024, 6, 5), source: 'manual'), // 5 Tage
49         PeriodData(start: DateTime(2024, 6, 30), end: DateTime(2024, 7, 3), source: 'manual'), // 4 Tage
50       ];
51       final result = calculatePeriodLength(perioden);
52       expect(result, 5); // (5 + 4) / 2 = 4.5 = gerundet auf 5
53     });
54   });
55 }
```

4. User Stories

1. Zyklusverfolgung

Als Nutzerin möchte ich meinen Menstruationszyklus und Symptome protokollieren, um meine Gesundheit besser zu verstehen.

Beschreibung: Nutzerinnen können Daten eingeben, eine Kalenderübersicht einsehen und Erinnerungen erhalten.

2. Community-Austausch

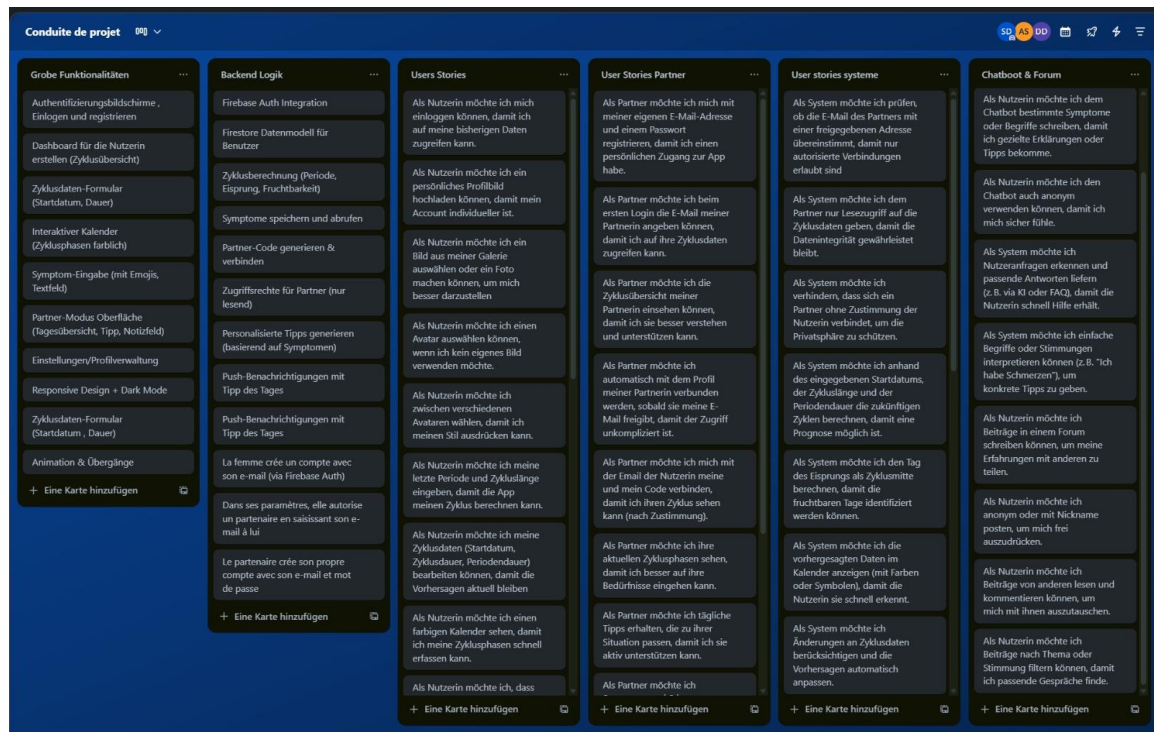
Als Nutzerin möchte ich Erfahrungen mit anderen teilen und mich austauschen, um Unterstützung zu finden.

Beschreibung: Die App bietet ein Diskussionsforum zum Posten und Kommentieren.

3. Partner-Modus

Als Nutzerin möchte ich meinen Zyklus mit einer vertrauten Person teilen können, ohne die Kontrolle über meine Daten zu verlieren.

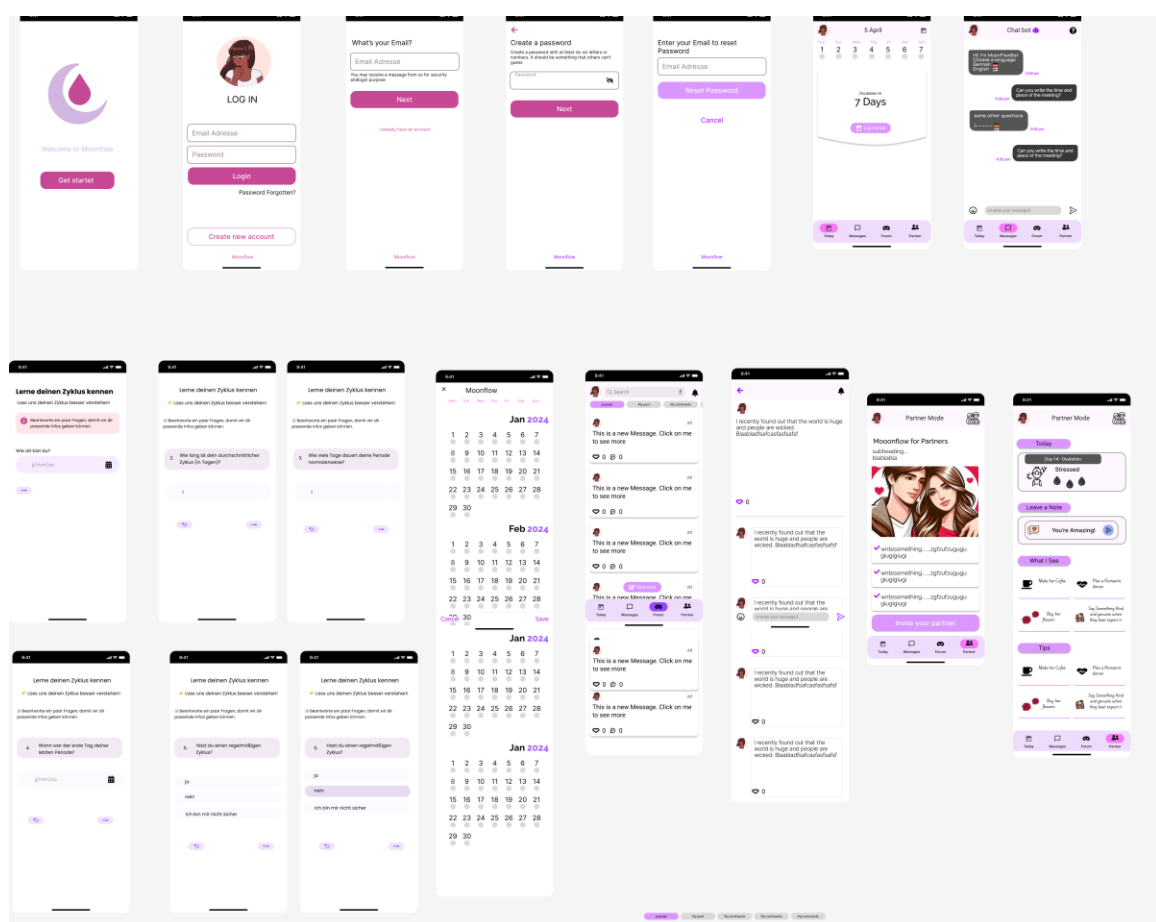
Beschreibung: Partner:innen erhalten Leserechte und Benachrichtigungen bei Updates.



5. Wireframes und Designentwicklung

Wireframes

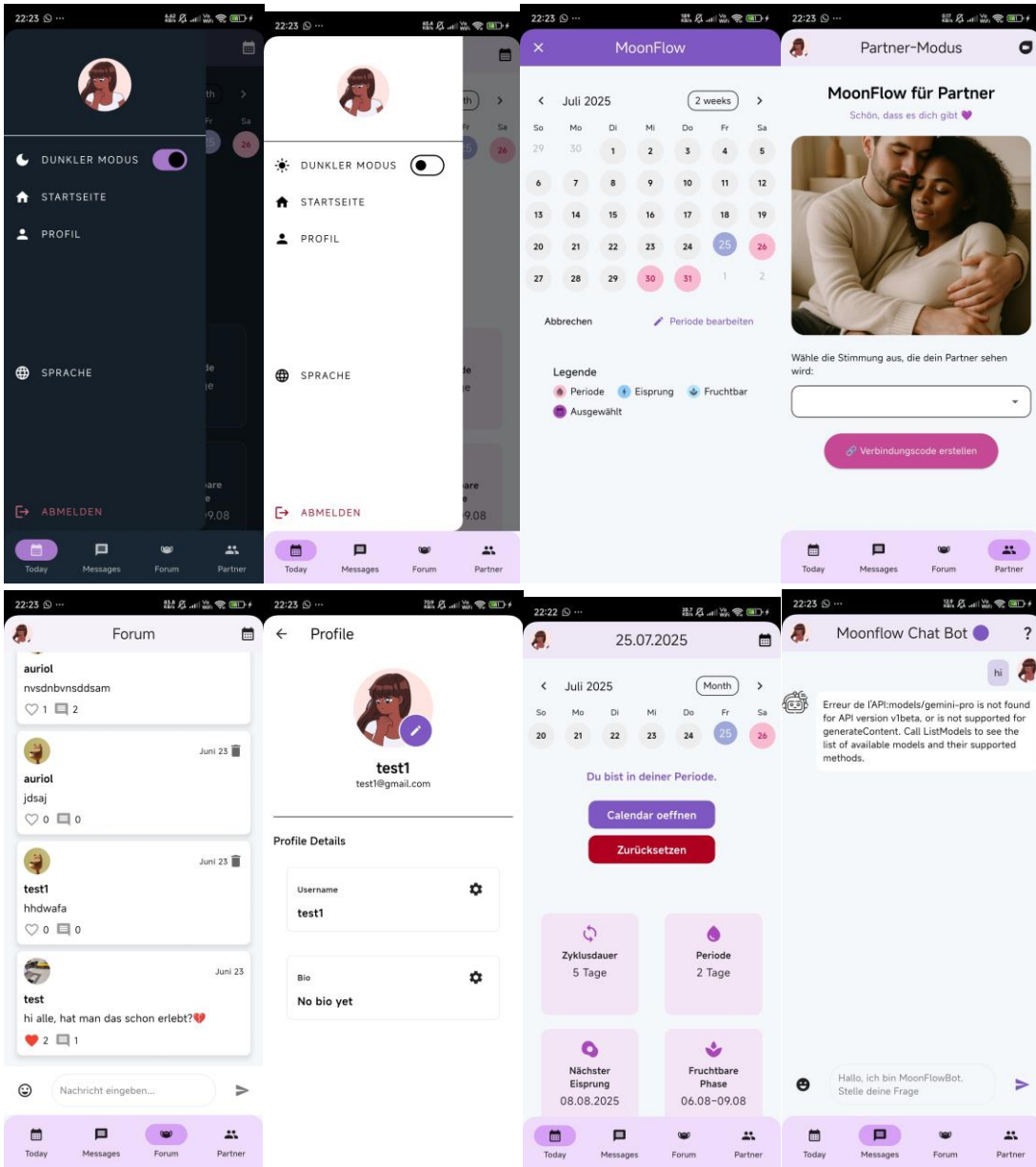
Die ersten Entwürfe wurden mit Figma erstellt und zeigen die Grundstruktur der App, einschließlich Zyklus kalender, Symptom-Tracker und Community-Forum.



Designentwicklung

Während der Entwicklung wurde das Design mehrfach überarbeitet, um Benutzerfreundlichkeit und Diskretion zu verbessern. Änderungen umfassen:

- Vereinfachte Navigation
- Dezentere Farbgestaltung
- Verbesserte Barrierefreiheit
- Ergänzende Funktionen, die im ursprünglichen Entwurf nicht enthalten waren



6. Implementierungsdetails

6.1 Architektur

- Architektur-Muster: MVVM (Model-View-ViewModel)
- Framework: Flutter (Dart)
- Backend: Firebase Authentication, Cloud Firestore

6.2 Schlüsselmodule

1. Authentifizierung

- Registrierung, Login und Passwort-Wiederherstellung über Firebase Auth

2. Zyklus- und Symptom-Modul

- Erfassung, Bearbeitung und Anzeige von Zyklusdaten und Symptomen

3. Community-Modul

- Beiträge verfassen, lesen und kommentieren im Diskussionsforum

4. Partner-Modus

- Verwaltung der Partnerzugriffe

5. ChatBot Modul

-Bietet Nutzerinnen eine dialogbasierte Unterstützung bei Fragen zur Zyklusverfolgung, App- Nutzung und allgemeinen Gesundheitsthemen.

6.3 Datenfluss

- Nutzerregistrierung und Login über Firebase Auth
- Zyklus-, Symptom- und Community-Daten in Firebase Firestore gespeichert
- Bilder werden in Firstorage gespeichert

7. Codequalität

7.1 Code-Standards

- Programmiersprache: Dart
- Styleguide: Effective Dart
- Linting und statische Analyse mit Dart Analyzer

7.2 Testing

- Unit Tests für zentrale Funktionen (z.B. Zyklusdatenerfassung)
- Integrationstests in Arbeit
- End-to-End Tests geplant

7.3 Code-Review

- Regelmäßige Peer-Reviews
- Checklisten für Style, Sicherheit und Testabdeckung

8. Wartbarkeit

8.1 Dokumentation

- Inline-Kommentare und Dartdoc für wichtige Klassen und Methoden

8.2 Versionsverwaltung

- Repository auf GitHub
- Branching nach GitFlow-Modell
- Commit-Messages im Conventional Commits Stil

8.3 Abhängigkeitsmanagement

- Verwaltung der Flutter-Pakete über pub.dev
- Regelmäßige Updates und Sicherheitsprüfungen

9. Entwicklungsstand und zukünftige Pläne

9.1 Aktueller Entwicklungsstand

- Implementierte Funktionen: Nutzerregistrierung, Zyklus- und Symptom-Tracker, Community-Forum, Partner-Modus, Mehrsprachigkeit
- In Entwicklung: Erweiterte Analysefunktionen, Offline-Modus, Moderationstools für Community

9.2 Zukünftige Pläne

- Erweiterung um zusätzliche Tracking-Möglichkeiten (z.B. Stimmungs- und Schlaftracker)
- Verbesserte Datenschutz-Features und Verschlüsselung
- Integration weiterer Kommunikationsmöglichkeiten innerhalb der Community

10. Fazit

Diese Dokumentation bietet einen umfassenden Überblick über die Entwicklung der Moonflow App, deren Ziel es ist, Frauen eine sichere und einfache Möglichkeit zur Zyklusverfolgung und zum Austausch zu bieten. Durch den modularen Aufbau und den Einsatz moderner Technologien ist die App zukunftssicher und erweiterbar gestaltet.