

Name	Mirel
Vorname	Apetrei
Arbeitstitel der App	SmartCook
GitHub Repository URL	https://github.com/SI-Classroom-Batch-006/ios-abschluss-mirelApetrei

Zielsetzung

Ziel der Abschlussaufgabe ist es, das Gelernte innerhalb des Swift-Moduls in einer eigenen App umzusetzen und diese App abschließend in einer Präsentation vorstellen und erklären zu können. Diese App soll anschließend als Teil eures Portfolios auf GitHub veröffentlicht werden.

Anforderungen an die App

Mindestens 5 Screens

Deine App muss mindestens 5 verschiedene Screens anzeigen. Jeder Screen soll dabei sinnvoll genutzt werden.

Navigation

Zur grundlegenden Navigation kann ein TabView genutzt werden (muss aber nicht). Wichtig ist, dass mindestens 1 Sheet und 1 NavigationLink genutzt wird. Optimal wäre, die Navigation mit destinations zu integrieren.

Anzeigen von Daten innerhalb einer Liste

In einem deiner Views muss eine Liste angezeigt werden, welche sinnvoll integriert ist und nützliche Inhalte liefert. Je nach App können optional auch Swipe Actions integriert werden.

Abspeichern von Daten (Core Data / Firebase)

Daten, die von deinen Usern generiert werden, sollen entweder zur Offline-Nutzung mit Core Data zur Verfügung stehen oder du integrierst Firebase, um die Daten zu synchronisieren. Wenn du dich für Firebase entscheidest, nutze den Firestore zum Speichern und Authentication, um User zu authentifizieren.

API Call

Such dir eine API, welche thematisch relevanten Inhalt für deine App bereitstellt und nutze die API, indem du einen API Call integrierst.

MVVM

Zur Integration der bisher gelernten Softwarearchitektur soll auch deine App MVVM nutzen. Hier wäre optimal, wenn du auch Repositories nutzt (diese sind aber keine Pflicht).

Pitch

Schreibe in diesem Feld 3-5 Sätze, die beschreiben, welche Funktionalitäten deine App bieten wird und welchen Nutzen sie erfüllen soll.

Bei dieser App handelt es sich um eine einfache Koch App, bei der sich der Benutzer anmelden und Rezepte nach mehreren Kriterien finden kann, wobei das wichtigste dasjenige nach den gewünschten/zu Hause verfügbaren Zutaten ist.

Aufschlüsselung der Anforderungen

Schreibe hier in jedes Feld mit wenigen Sätzen, wie deine App die einzelnen Anforderungen erfüllen soll und welche Technologien du dafür verwenden möchtest (z.B. NavigationStack für Navigation).

Mindestens 5 Screens:

- | | |
|------------------|--------------------------|
| 1. LoginView | 5.RecipeListView |
| 2. SignupView | 6.RecipeDetailView |
| 3. HomeView | 7. ShoppingListView |
| 4.FindRecipeView | 8.ShoppingListDetailView |

Navigation:

- eine bottom TabBar, und NavigationLink

Anzeigen von Daten innerhalb einer Liste:

Rezepte, Zutaten

Abspeichern von Daten (Core Data / Firebase):

Firestore

API Call:

Spoonacular.com

GET <https://api.spoonacular.com/recipes/findByIngredients>

POST <https://api.spoonacular.com/recipes/visualizeRecipe>

MVVM:

1. Models : z.B. UserModel

2. Views: z.B. HomeView

3. ViewModels: z.B. HomeViewModel

4. Repositories: z.B. UserRepository

5. Utilities: z.B. FirebaseAuthService

Meilensteine

Hier soll festgelegt werden, welche Aufgaben zu welchem Meilenstein erledigt sein sollen. Die Meilensteine sind hierbei immer freitags während der Projektphase und ersetzen damit die Freitags-Aufgabe. Einige Meilensteine sind vorgegeben.

Es ist Pflicht, dass jeden Freitag der aktuelle Stand der App auf Github gepusht wird!

Freitag 25.08.23

Figma-Prototyp ist fertiggestellt und zusammen mit diesem ausgefüllten Dokument abgegeben.

Freitag 01.09.23	Firestore and Firestore fertig, mit User Authentication LoginView und SignupView und MVVM-Struktur Model Definitions: Define the models (UserModel, RecipeModel, RecipeDetailModel, ShoppingListModel).
Freitag 08.09.23	Erstellen Sie Netzwerkklassen, um API-Aufrufe an Spoonacular und Firebase zu verarbeiten. Repositories und ViewModels mit eigenen Funktionen
Freitag 15.09.23	User Repository: Implement UserRepository with methods for signup, login, and user data retrieval. Authentication ViewModel: Implement AuthenticationViewModel.
Freitag 22.09.23	Popular Recipe Repository: Implement PopularRecipeRepository. Home ViewModel: Implement HomeViewModel. Home View: Create HomeView to display user info and popular recipes.
Freitag 29.09.23	Recipe Repository: Implement RecipeRepository with methods for recipe fetching. Find Recipe ViewModel: Implement FindRecipeViewModel. Find Recipe View: Create the FindRecipeView. Recipe Detail ViewModel: Create RecipeDetailViewModel. Recipe Detail View: Create the RecipeDetailView.
Freitag 06.10.23	Shopping List Repository: Implement ShoppingListRepository. Shopping List ViewModel: Implement ShoppingListViewModel. Shopping List View: Create ShoppingListView. Shopping List Detail View: Create ShoppingListDetailView.
Freitag 13.10.23	Die App ist fertig, erfüllt alle Anforderungen und hat super viel Spaß gemacht. Wir können das nächste Projekt kaum erwarten!

Bonus Features

Die oben beschriebenen Anforderungen stellen lediglich Mindestanforderungen dar. Wenn ihr zusätzliche Features integrieren möchtet, ist das natürlich super, sowohl für die Bewertung als auch für euer Portfolio!

Wir raten stark dazu, an diesen Features erst zu arbeiten, wenn alle anderen Anforderungen erfüllt sind!

Feature	Erklärung
Meal planning	Ich möchte für jeden Tag eine Planungsmahlzeit umsetzen

Dieses Dokument muss unterschrieben werden, um einen "Vertrag" über die zu erbringende Leistung mit uns zu simulieren.

Mir ist bewusst, dass meine App anhand der Angaben in diesem Dokument bewertet wird.

Ort, Datum

24.08.2023

Unterschrift

