

Name	Slawomir Dragon
Titel der App	Spitirual TAROT App
GitHub Repository URL	https://github.com/drAgon-235/TAROT.git

<https://github.com/SI-Classroom-Batch-009/ios-abschluss-drAgon-235.git>

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 muss dabei sinnvoll genutzt werden.

Navigation

Zur grundlegenden Navigation muss ein **TabView** genutzt werden. Wichtig ist auch, dass mindestens **1 Sheet** und **1 NavigationLink** vorhanden sind.

Anzeigen von Daten innerhalb einer Liste

In einem der Views muss eine **Liste** angezeigt werden, welche sinnvoll integriert ist und nützliche Inhalte liefert. Dies kann entweder die List von SwiftUI sein oder ein ScrollView mit einer ForEach.

Abspeichern von Daten (Core Data / Firebase)

Daten, die von Usern generiert werden, sollen entweder zur Offline-Nutzung mit **Core Data** zur Verfügung stehen oder über **Firebase**, um die Daten zu synchronisieren. Bei der Speicherung in Firebase muss zusätzlich Authentication genutzt werden.

API Call

Die API sollte thematisch relevante Inhalte für die App bereitstellen. Diese kann mit oder ohne API-Key sein.

MVVM

Zur Integration der bisher gelernten Softwarearchitektur muss auch die App MVVM nutzen. Optional kann hier direkt mit Repositories gearbeitet werden.

Pitch

Schreibe in diesem Feld 3-5 Sätze, die beschreiben, welche Funktionalitäten deine App bieten wird und welchen Nutzen sie erfüllen soll. Stell dir vor, du beschreibst die App einem/einer Freund:in.

Ob als kleiner Spaß unter Freunden, auf einer Feier, zum Zeitvertreib oder als Entscheidungs- und Lebenshilfe – TAROT Karten sind voll im Trend. Es ist auch das Thema meiner App. Der User kann nicht nur verschiedene Legearten nutzen, um sich die Karten legen zu lassen; er kann auch sein Wissen über Tarot erweitern. Er kann sich eine „Karte des Tages“ ziehen. Er kann sich auch von einem „Zitat für den Tag“ inspirieren lassen, welcher durch eine API randomized generiert wird und in einer FavoritenListe gespeichert und evtl. auch verschickt werden kann.

Aufschlüsselung der Anforderungen

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

Mindestens 5 Screens:

- Login/Registration
- TabView mit mind. 2 Tabs: („Tarot“, „Quotes“)
- „Tarot“ (Optionen für Tageskarte, Lernbereich, Kartenlegen – diverse Legearten & Interpretation)
- „Quotes“ (Optionen für Zitat des Tages(API), Favoritenliste(FirestoreDB))
- evtl. Setting, Profile etc

Navigation:

- TabView mit mind. 2 Tabs: („Tarot“, „Quotes“, evtl. Setting, Profile)
- NavigationStack, NavigationLink, sheet
- Automatisch zwischen Login <-> Registration Views (programmed with Enum)

Anzeigen von Daten innerhalb einer Liste:

- Liste mit Favorit-Zitaten (live aus FirestoreDB, Zitate sind löschar, nicht änderbar, no limit)
- Vertikale Liste mit 5 horizontalen Listen für die Karten der 5 Arkanas-je 22 Karten (Lernbereich)
- Liste mit den Interpretationen pro Karte mit – je nach Legeart – zw. 3 und 7 Karten.

Abspeichern von Daten (Core Data / Firebase):

- Lieblingszitate werden in einer Favoriten DB in Firestore live gespeichert und gelöscht
- Die Karten bekommen eine eigene DB (noch unsicher ob CoreData oder Firestore oder sogar beide fürs Caching)

API Call:

- <https://zenquotes.io/>
- <https://zenquotes.io/api/random/>
- <https://zenquotes.io/api/today/>
generiert Zitate mit Lebensweisheiten berühmter Persönlichkeiten, gratis, kein Key nötig

MVVM:

Wird für jedes Feature extra implementiert: also für Login, Cards, Quotes etc... jeweils mit eigener MVVM Struktur z. T. mit extra Repository
Da drüber steht der Firebasemanager und der Navigator mit der TabView

Meilensteine

Die Projektphase ist in vier große Meilensteine unterteilt (alle zwei Wochen). Überlege dir, was du für deine App in dieser Zeit erreichen möchtest, z.B. UI oder bestimmte Features.

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

Woche 1 Freitag 26.01.24	<ul style="list-style-type: none">• Konzept erstellt• Figma-Design fertiggestellt• Pflichtenheft ausgefüllt und abgegeben
Woche 2 Freitag 02.02.24	Top Navigation , TabView, Firebase Authentication
Woche 3 Freitag 09.02.24	API implementieren FirestoreDB für Favoriten Zitate
Woche 4 Freitag 16.02.24	Repository mit Karten erstellen Core Data oder Firestore DB oder beides implementieren

Woche 5

Alle Kartenlegearten implementieren mit Animationen

Woche 6

Alle Kartenlegearten implementieren mit Animationen

Woche 7

Letzte Optimierungen am Code und Vorbereitung der Präsentation / README

Woche 8

App ist fertiggestellt – Abschlusspräsentation

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!

Es sollten unbedingt zuerst die Mindestanforderungen erfüllt sein, bevor an Bonus-Features gearbeitet wird.

Feature	Erklärung
Diverse Animationen für fliegende, sich drehende und tanzende Karten	<ul style="list-style-type: none">- Im HomeScreen könnten im Hintergrund Karten von oben nach unten Fallen, gleichsam Schnee.- Beim Kartenlegen sollen auch noch nicht genau definierte Animationen zum Einsatz kommen.
Gut wäre die Option, zwischen online und offline Betrieb wählen zu können.	<ul style="list-style-type: none">- Ein häufiger UserCase ist in einer Hütte in den Bergen jenseits vom InternetEmpfang, wo man gerne ohne Login einfach nur mal das Smartphone durchreicht und alle legen sich Karten und interpretieren sie gemeinsam.
<ul style="list-style-type: none">- Nur 1 Tageskarte pro Tag/User- Nur 1 Zitat pro Tag/User	<ul style="list-style-type: none">- Um dem Terminus „Karte bzw. Zitat DES TAGES“ gerecht zu werden wird überlegt, die App bzw. das API so zu programmieren, dass wirklich NUR EIN Zitat und nur EINE Tageskarte pro Tag „gezogen“ werden können.

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.

Mannheim, 26.01.2024

Ort, Datum

Slawomir Dragon

Unterschrift