



Apollo

Music Dating and Social Network Application

Samantha Kamath | Megan Page | Yunting Zhao

CSC 431 | Introduction to Software Engineering

March 15, 2021

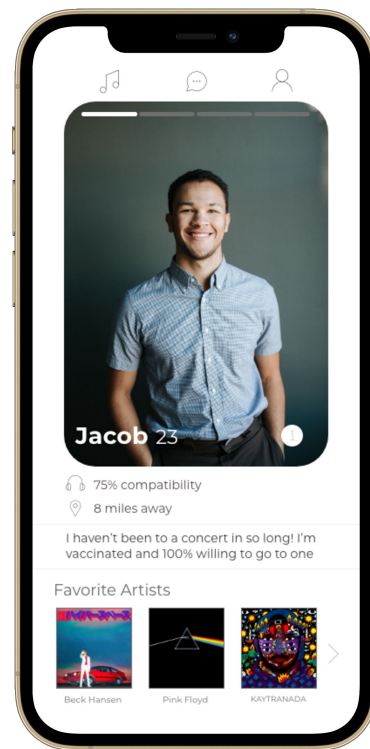


Review of the Project

Project Review

Type here

Type here





System Analysis



System Overview

The system consists of **three** main parts.

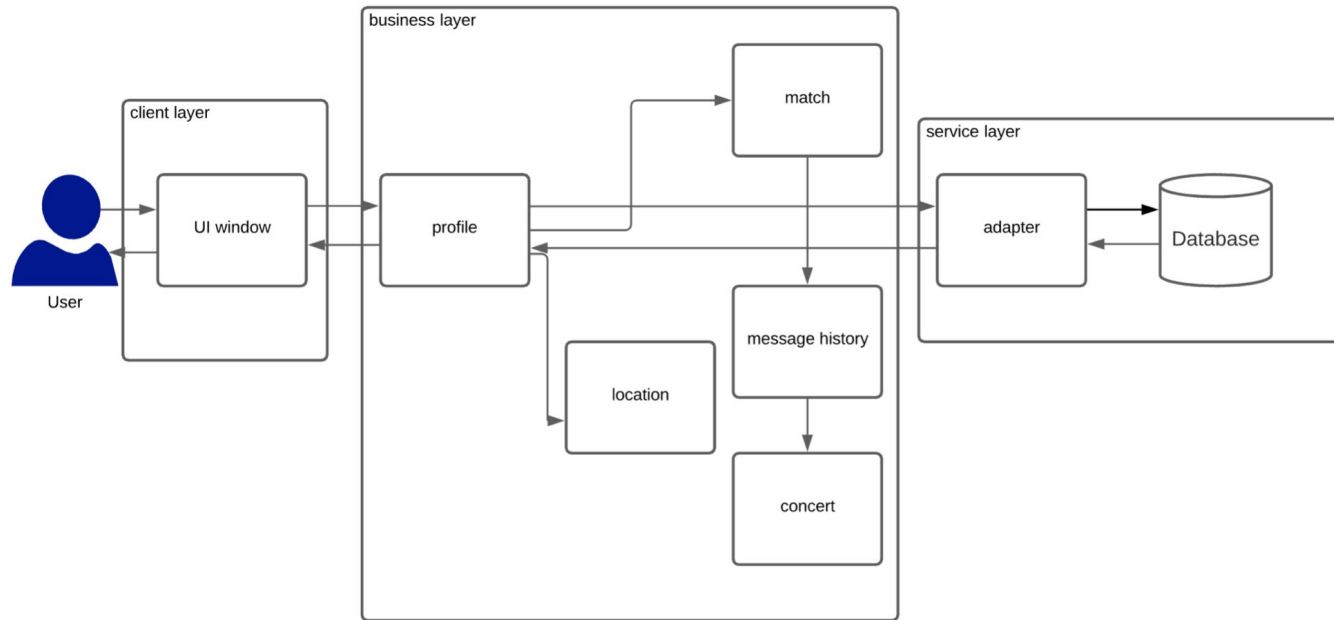
1. **Business logic server**, which includes a Profile class object; this object stores the user's information, including each user's set of matches, potential matches, and user preferences. The Profile class is responsible for storing message history information for each of the user's matches.
2. **Adaptor class**, which converts the information from the Profile class to the JSON data used to store information in Firebase Realtime Database; the Adaptor class helps carry out functions like compatibility matching and creating matches.
3. **Service layer**, which includes Firebase Realtime Database; this database provides functions such as registration, login, and authentication; the database also stores user information.

client layer

busniess layer

service layer

System Diagram



Actors Identification



New Users



Registered Users



Platform Server

Architectural Style

UI layer

business logic layer

service layer

Design Patterns

We predict that the **Adaptor Design pattern** will be primarily used in our project. Adaptor classes will be used to communicate between the business logic layer and service layer. Data from the database in the service layer will be converted to the Profile class in the business logic layer using an adaptor class; similarly, user profiles will be converted to JSON data to add to the database in the service layer using another adaptor class.

Frameworks

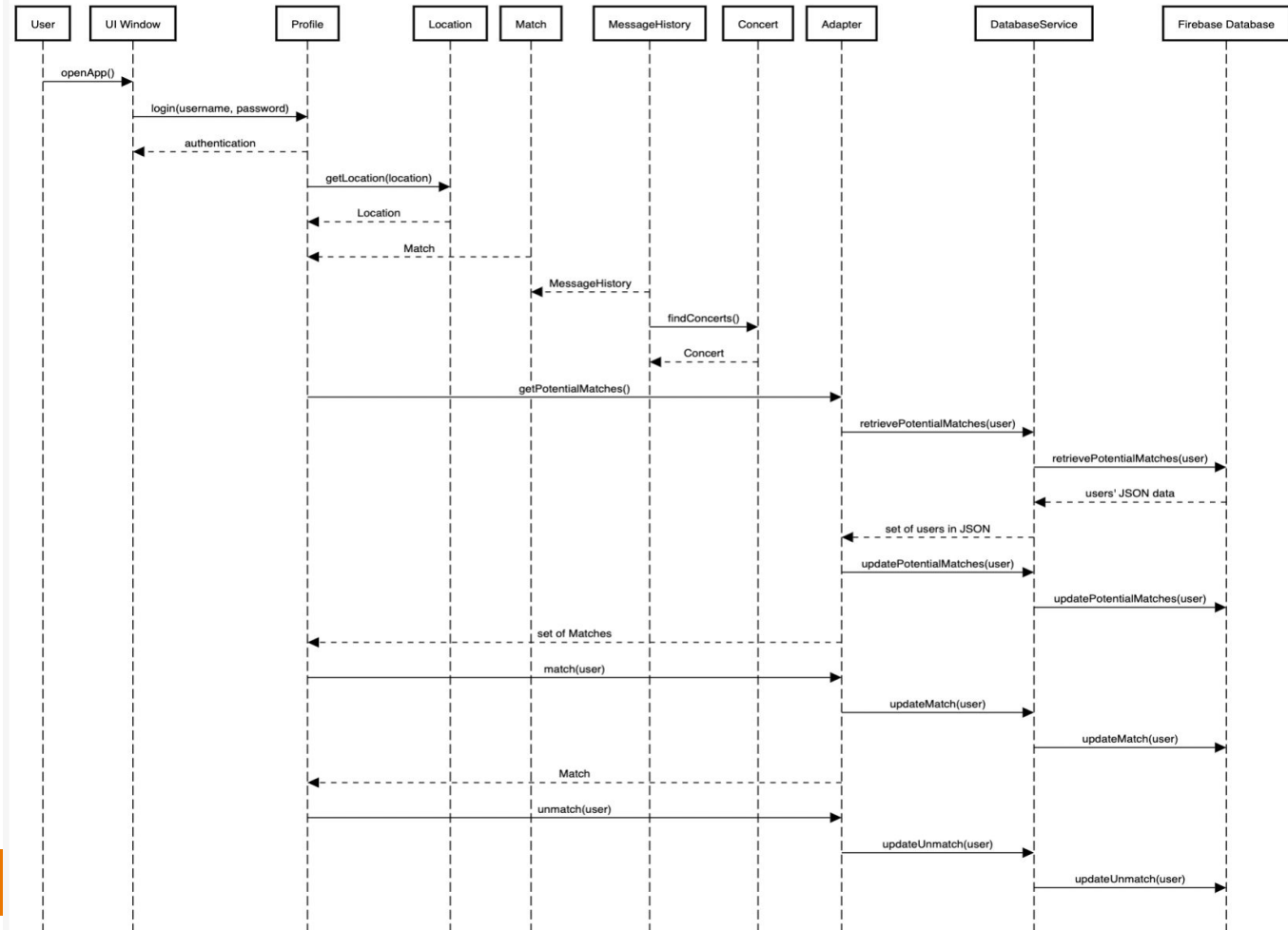
We will be using **Flutter** for the **UI framework**; Flutter is well-documented, and developers on our team have previous experience with the framework. However, our app also requires additional backend services, such as authentication, database storage, and business logic to implement compatibility algorithms and the upcoming concerts algorithm. **Firebase** is a Backend-as-a-Service that can implement authentication, cloud database storage, and security functions for our app with ease. For the other **backend logic**, such as the app's algorithms, the **ExpressJS** framework will be used. ExpressJS has access to all the NodeJS packages, and developers on our team have previous experience using JavaScript and ExpressJS.



express



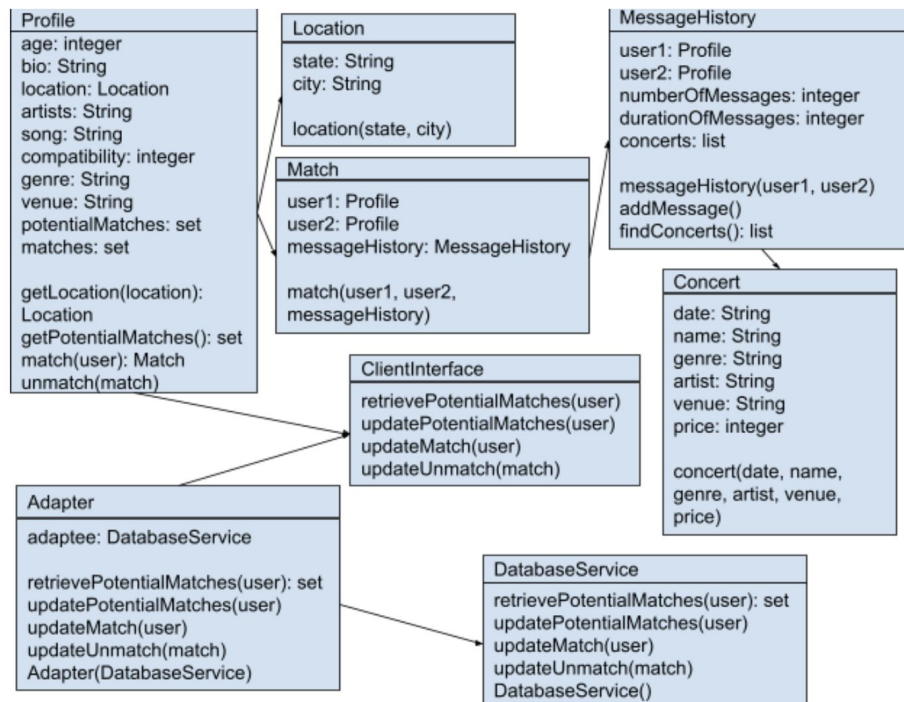
Functional Design



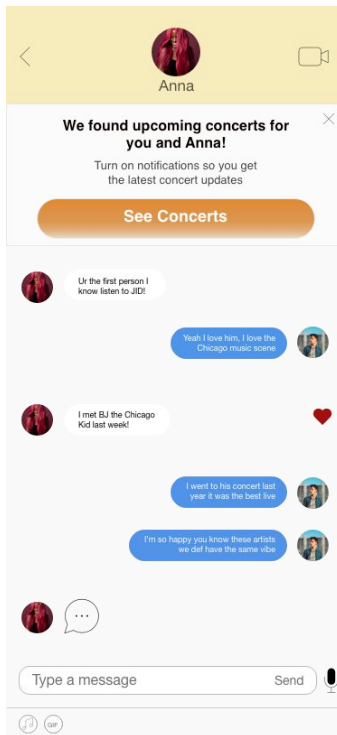
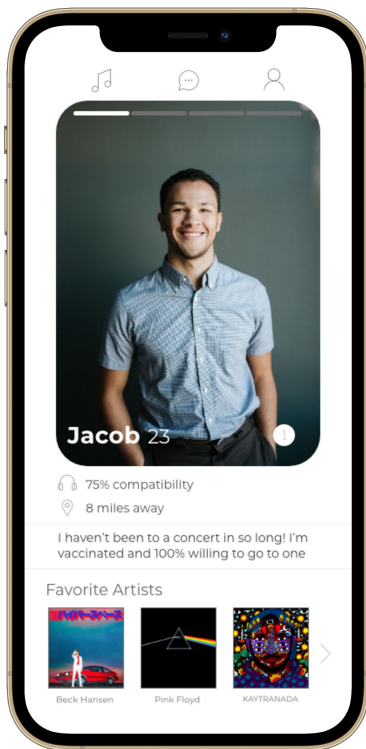


Structural Design

Class Diagram



Project Mockup



Project Planning + Github Link

1 In Progress + ...

- Final slides
#3 opened by skamath2000
presentations
Final Slides

0 Needs Review + ...

5 Complete + ...

- Design Draft 2
#2 opened by skamath2000
design draft 2
Design Doc
- Design Draft 1
#1 opened by skamath2000
design draft 1
Design Doc
- Requirements Doc Draft 1
#4 opened by skamath2000
srs draft 1
SRS
- SRS Draft 2
#5 opened by skamath2000
srs draft 2

main 1 branch 0 tags

Go to file Add file Code

skamath2000 added in the updated docs 1499498 9 minutes ago 2 commits

presentations	added in the updated docs	9 minutes ago
proposal	added in the updated docs	9 minutes ago
sas	added in the updated docs	9 minutes ago
srs	added in the updated docs	9 minutes ago
.gitignore	Initial commit	14 days ago
README.md	Initial commit	14 days ago

README.md

apollo_CSC431

Apollo: A Musical Social Network and Dating App

Switch branches or tags

https://github.com/skamath2000/apollo_CSC431



Questions?