

**Software Engineering Assignment 3**  
**CS-400-01**  
**Software Design and Development**  
**Fall 2021**

**Developing a Software Architecture**

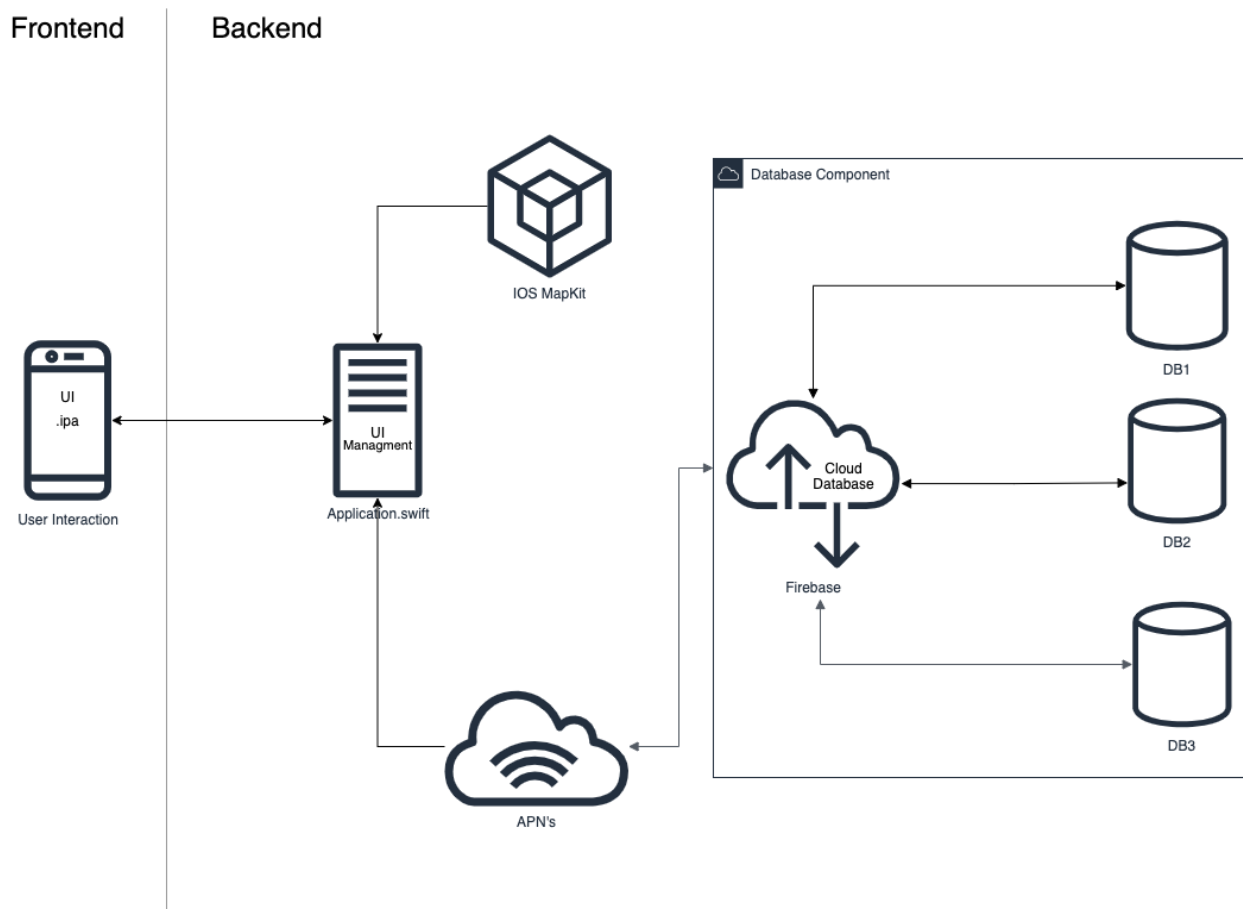
**Introduction**

For software there are many ways of designing the architecture, the system architecture will affect many aspects of the software itself, including responsiveness, reliability, availability, security, usability, maintainability and resilience. An issue however is that some of these aspects oppose each other, such as usability and security. If you want to have a very secure application, with lots of security layers, then that would lower the usability, because the user would have to pass all these layers and authentications to be able to interact with the application. On the other hand, if you want to have it more usable, then it would come at the cost of security, easier for users to use, but also for malicious users to exploit. Similarly, the other aspects mentioned above are also opposites and the developer must choose what aspects to optimize on the cost of those that are considered less important to the specific software. These trade-offs naturally depend on the type of software that is being made.

In our case, Kaffi will consist of an easy-to-use UI, so we will want to optimize usability for the average user, because most of our users would be people looking for coffee shops rather than business owners. We would also want it to be secure, for those who own businesses, so that their pages and data wouldn't be sabotaged and stuff like that. Also, with the future implementation of forums (Not MVP) we would want that data to be more secure as well so that people won't be able to go in and edit other people's data and such.

## Part I: Organization

*How should the system be organized as a set of architectural components, where each of these components provides a subset of the overall system functionality?*



### User Interface component

This would be the frontend of the system. It would be in the form of iOS app which would be available for download from the Appstore. This is where all user interaction would happen, like login, map-manipulating, searching, and viewing business pages and the users own profile.

### User Interface management component

This would be the connector between the UI component and the rest of the backend of the software. It would authenticate user login, apply the searches through the database component and import the map libraries to display for the user.

### Map component

This is the map libraries that would be imported to the User Interface management and then displayed in the UI. Here we are planning on using the iOS MapKit library, which would have everything we need to implement the map feature.

Database component

Finally, the database component. Here we are planning on using firebase or MySQL/SQLite. Firebase would work as the database manager, and we would add our databases into it. This would then contain the database(s), we are still deciding on what the best layout would be, but we would need to store the user data, the business pages' data and eventually the forums data. We could keep this data in separate databases or make a relational database containing the different data in one database, mapping the business page to the user data through a foreign key for example. All of this is still under consideration though, but we will figure this out when we start developing. Here we are also planning on having a multi-tenant rather than a multi-instance setup for our database.

## **Part II: Architectural Components**

How should these architectural components be distributed and communicate with each other?

The different components would communicate with each other, for example the database, at least with firebase, would communicate with the UI management component through APN's.

The MapKit would be imported as a library into the main application (UI management). The UI management component would work as the middleman between the UI and the backend stuff, like the databases and the maps lib, so all communications would go through that component.

Some aspects of the communication part and the exact way that it will be implemented, we are not sure yet, but the MapKit and firebase part we are fairly certain.

## **Part III: Component Services**

What technologies should you use in building the system and what components should be reused?

### **Firestore:**

As mentioned earlier, we would use firestore as a service for our database. It's a cloud service through google, so everything would be stored online. Here we would store user information, user preferences, business page information and forums information. It triggers backend code that runs in a secure environment with cloud functions, and it sends notifications through cloud messaging.

### **iOS MapKit:**

This is the maps library that we would reuse which will allow us to implement our map feature. We would implement our pins, that indicate a business location through this feature as well.

### **iPhone GPS tracker:**

As this is being designed as an iOS app, the user will have an iPhone/iPad/iPod touch which means that it would have an integrated GPS tracker, we would use this technology to implement live tracking, together with the map library, we would have fully functional, live map.

#### Part IV: **Additional information and comments:**

Through this design architecture we are planning on having