**Project Documentation**

**Key Decisions**

Over the course of designing *Gophur*, several decisions were implemented both on a technical and structural basis in order to keep things flowing without interruption. Below, some crucial decisions are highlighted, as well as their reasoning and effects.

*Decision 1: Database Lead*

The team was quite set on creating a mobile app to begin with, though no one had a high level of experience to begin with. A structural dilemma that was faced early on was the fact that we had more group members than there were team roles. We were told that a role could be split among two people, but this did not interest us early on. Back to brainstorming ideas for the app, everyone agreed on some technical structure that involved users creating accounts, being able to post images, and being able to chat with one another. The fundamental necessity for a program like this is the storage of user data in a safe and secure environment. In realizing this, we appointed one member as a new role: one that would be in charge of all databasing, data management, and database “gluing” from the front-end to the back-end. This member came to be known as the Database Lead.

*Decision 2: Specifications and Functionality*

A day or two of brainstorming led the team to ultimately decide on an app that would be social media-esque in a manner that would service businesses and customers. The app serves as a “middle-man” in all forms except financial transactions themselves. Users can either create accounts as businesses (business profiles) or customers (personal profiles), and interact with each other depending on their profile. For example, business profiles can post images about their services and dictate their hours, while personal profiles can schedule appointments, contact businesses directly, and even leave reviews. This wholly specific implementation of a social media platform was one that the six of us had never seen before. Part of the design process included gaining inspiration from other famous social media, but only in a style that would maximize ingenuity and unique design.

*Decision 3: XCode, Swift*

By now, the team had the general idea for the app, but we had no platform to implement it on. Ultimately, we strived to maximize the audience that would use the app without implementing a different version for different operating systems. In turn, the team chose to use Swift/XCode to code all of the front-end and back-end for the app. In doing so, we also sacrificed our ability to easily write code in a language that was more familiar to all of us, like C++. Furthermore, Swift was not incorporable into our entire team’s design experience: only five out of six of our group members were able to install and use Swift on their personal computers (this is outlined more in the Challenges section).

*Decision 4: Simplicity*

As the group moved forward with writing code and designing both the front-end and back-end, we didn’t know it until the deadline rapidly loomed, but each one of us had a lot on their plate. Much of the front-end implementation required tutorials online that were able to be modified once each of us had a sufficient understanding of Swift. Following combined hours of YouTube videos, website tutorials, and more, every member of the group had an extra feature or two they would have liked to see implemented before the deadline that was unfortunately scrapped. All of these features were complicated technically but may be included when we continue to improve the design of the app following the due date for the project.

These include:

* True “middle man” financial transactions within the app
* Using Kotlin for Android development
* Currently popular front-end features such as:
  + Taking live photos
  + Businesses live streaming
  + Encrypted messaging

In order to keep the design simple with enough time to finish back-end programming and full functionality of the app, these features (and more) were unfortunately scrapped.

**Challenges**

Creating *Gophur* was smooth for the most part, but there were a few key challenges that appeared over the course of the design process. They are outlined as follows:

The most prominent of challenges was learning how to continuously push and pull changes from GitHub. Using XCode, this feature was ingrained into the IDE; with a team of developers of this size, though, confusion quickly arose whenever the entire group was working at once. Individual work was able to be synthesized efficiently, as the programmer could notify everyone at once that they had just pushed some changes. This changed when everyone was changing things at once, and required a meticulous degree of organization in order to keep the right changes active. The solutions to this generally involved manual methods of troubleshooting. This includes having to drag-and-drop files, editing the source code itself, and just talking things out the old-fashioned way.

Learning to use Firebase was another challenge the group encountered. Though this was conquered as time went on, the database software was incredibly difficult to learn at first. Management of account records such as e-mails, usernames, passwords, and other data associated with each account was intuitive enough to talk about, but establishing these commands within the back-end of pages such as the signup/login screen, the main feed, and the profile pages was tough to get used to.

**Core**

The core algorithm of our app varies based on the type of profile. Personal profiles can interact with one another and with businesses. These types can write reviews on the pages of business profiles (under their discretion), and they can also make friends with other personal profiles. Businesses, on the other hand, are able to post images of their brand and their services. These may include sample hairstyles for hair salons, artwork for tattoo parlors, and so much more. The app is catered toward the service industry rather than the food industry. As such, personal profiles can schedule and book appointments under a business profile’s page using our calendar feature. All personal information for both types of profiles, including content they post and reviews they write, are stored in our secure database.