LetsTalk™

Developers | Abdul Baflah and Shrey Arora

Final State

Sign Up/Join

We have a Join feature that allows new users to create a new account with their first name, last name, email, and password. Email verification does occur in the backend (for password reset emails, if/when necessary).

Log in

We obviously also have a login feature that is to be utilized returning users to login into their already existing accounts.

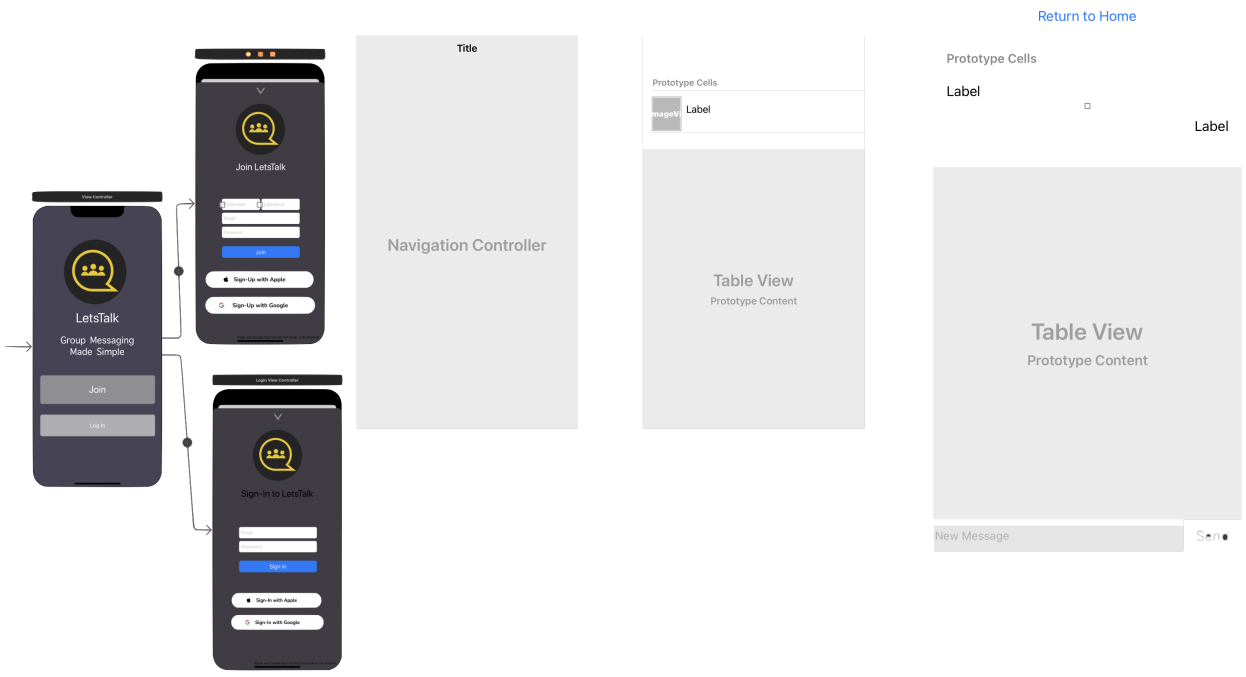
Home

Now in the home page, you will see all the current users that are registered with your org. You can pick any user to interact with and start a conversation

Refactoring & Alterations

- We deleted the group chat feature. This app only deals with one-to-one conversations. It was too difficult to implement and it caused many errors that crashed the app when implemented. With it, adding and deleting users from a group is also gone.
- Sending images was also scrapped from our app due to the complications of utilizing iOS's secure AVcam, and AVCaptureSession
- Leveraging Swift storyboards as opposed to utilizing SwiftUI for designing

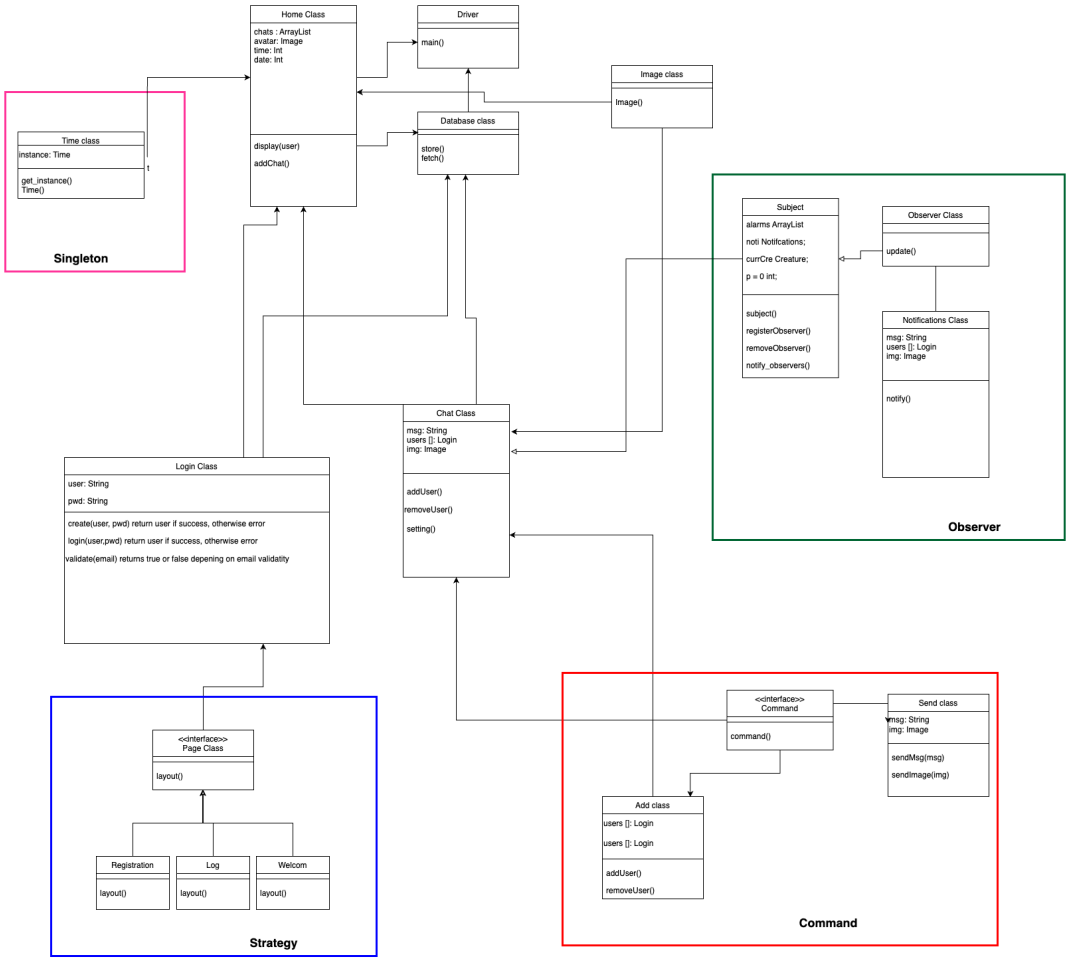
Final Swift View Controllers/UI



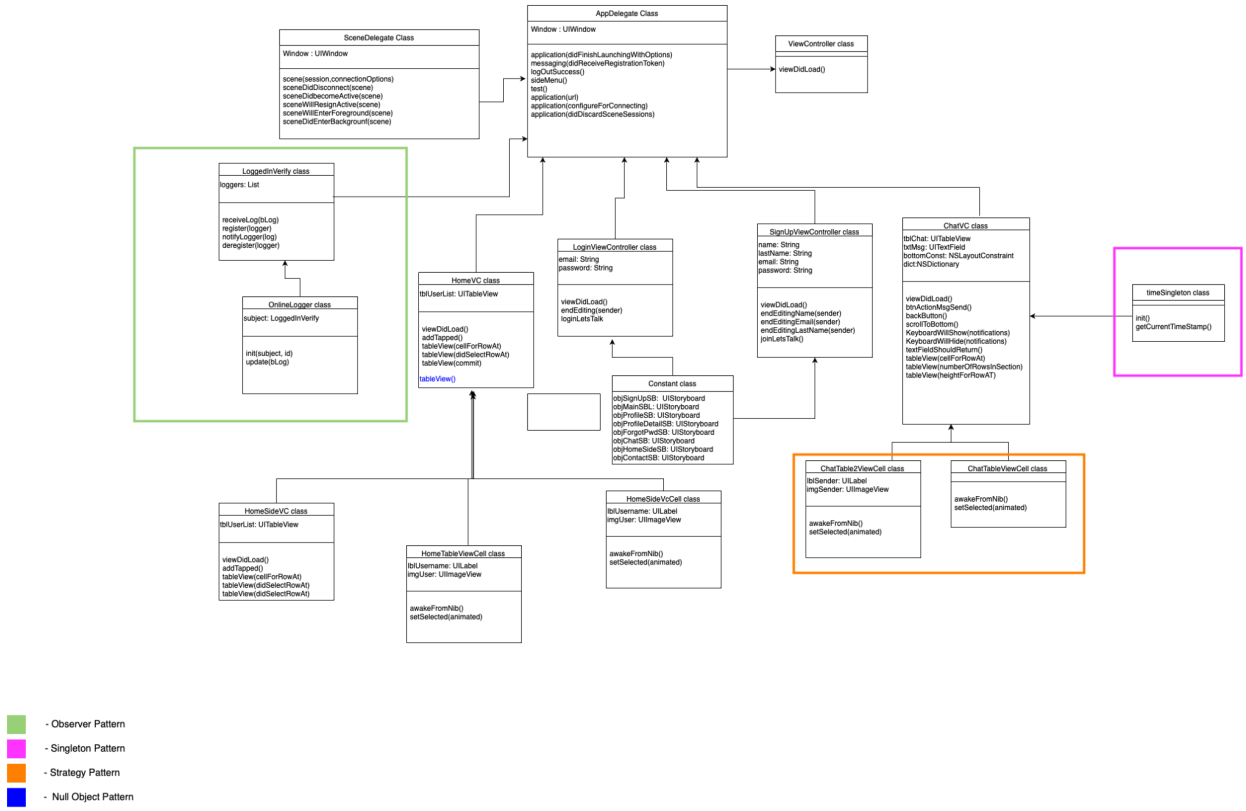
LetsTalk™

Developers | Abdul Baflah and Shrey Arora

Class Diagram(s)
Original Class Diagram | Project 5



Final Class Diagram | Project 7



Through the development process, LetsTalk pivoted to focusing on the enterprise based chatting as opposed to group chatting. During this pivot, changes such as alterations and removal of the Command pattern and eventual integration of the Null Object pattern proved to be fruitful in the development endeavor. And as we got familiar with the language and development environment, some classes seemed unnecessary such as the initial database class, its implementation was instead facilitated via Firebase's readily available functionality and thereby integration.

Design Process

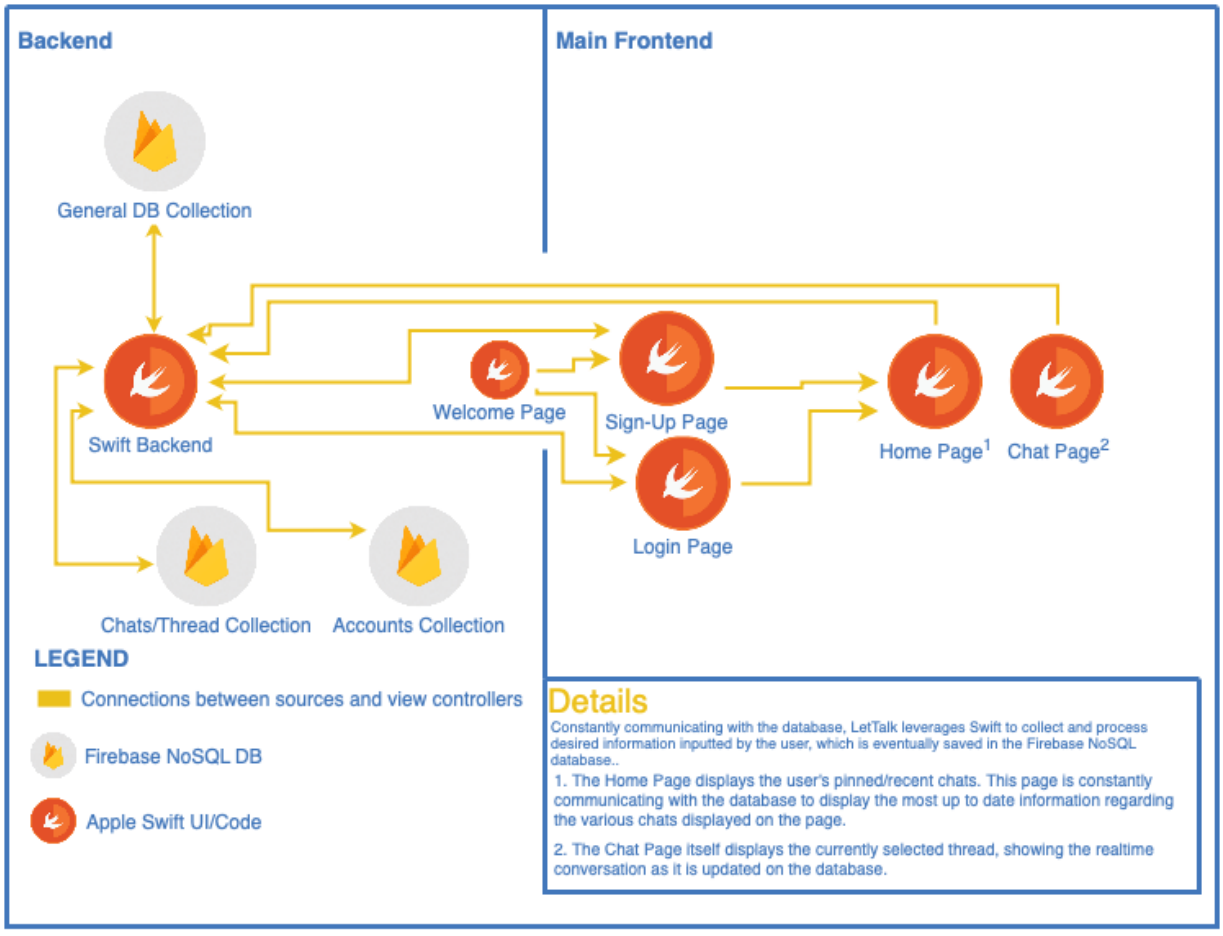
- One of the issues our team faced when developing this project was the lack of knowledge on Swift. Both members had little experience using Swift so deciding how to start was really tricky. This team spent a lot of time learning some basic concepts of Swift so that we could work on our project. As such doing uml and design was challenging
- One positive element that really helped us with this project was the many built-in functions/libraries that swift offers. These imports eased the process of developing the UI as it was a difficult task at first.
- Another element that helped us through this project was the inclusion of Firebase. Firebase was our database and it was really smooth to connect with Swift. All of our

project's data was safely stored, easily accessible, and smoothly managed thanks to Firebase functionality.

Third Party Code

- The only third party code that was implemented in the project was the cocoaPods dependencies and Firebase provided API access functions and code.
- Built in Swift classes and dependencies
    - Such as UIKit, UITableview, Foundation, etc.

Architecture Diagram



Resource Citations

- Infotech, Logistic. “Real-Time Chat with Firebase and IOS Swift Client.” *Logistic Infotech*, https://www.facebook.com/logisticinfotech/, 31 Dec. 2018,https://www.logisticinfotech.com/blog/real-time-chat-application-with-firebase/.

- Bush, Alex. “Swift Optionals and the Null Object Design Pattern | by Alex Bush | Smart Cloud.” *Medium*, Smart Cloud, 29 Mar. 2018, https://m.smartcloud.io/swift-optionals-and-the-null-object-design-pattern-7578b7448edf.