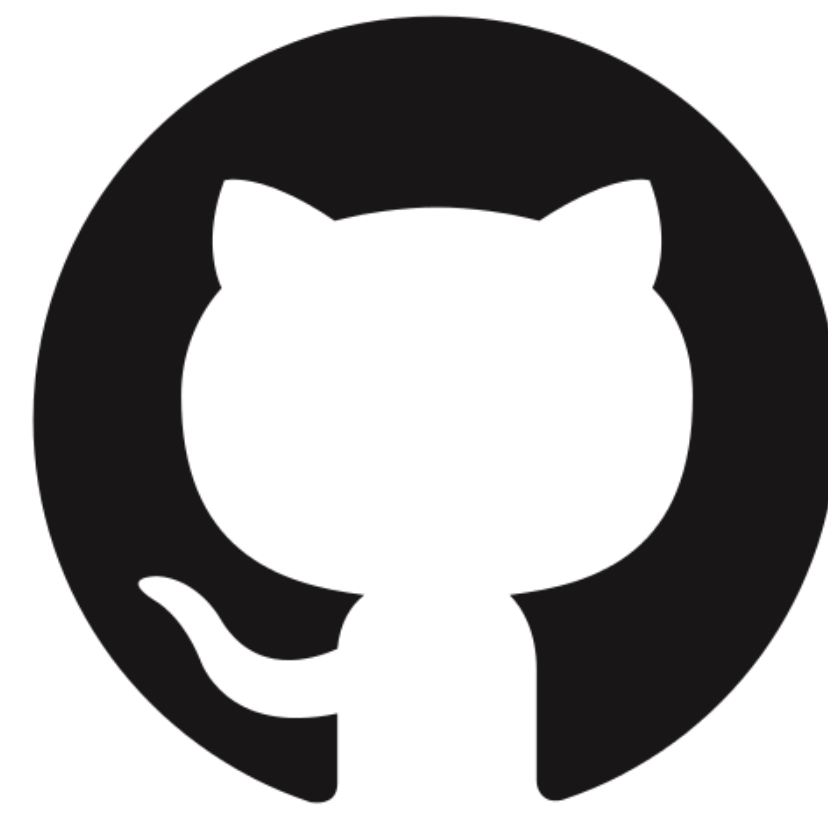




Biometric Identification

Student: Andy Garica, Florida International University
Product Owner: Rachelle Tobkes, Citrix
Instructor: Dr. Masoud Sadjadi, Florida International University



Problem

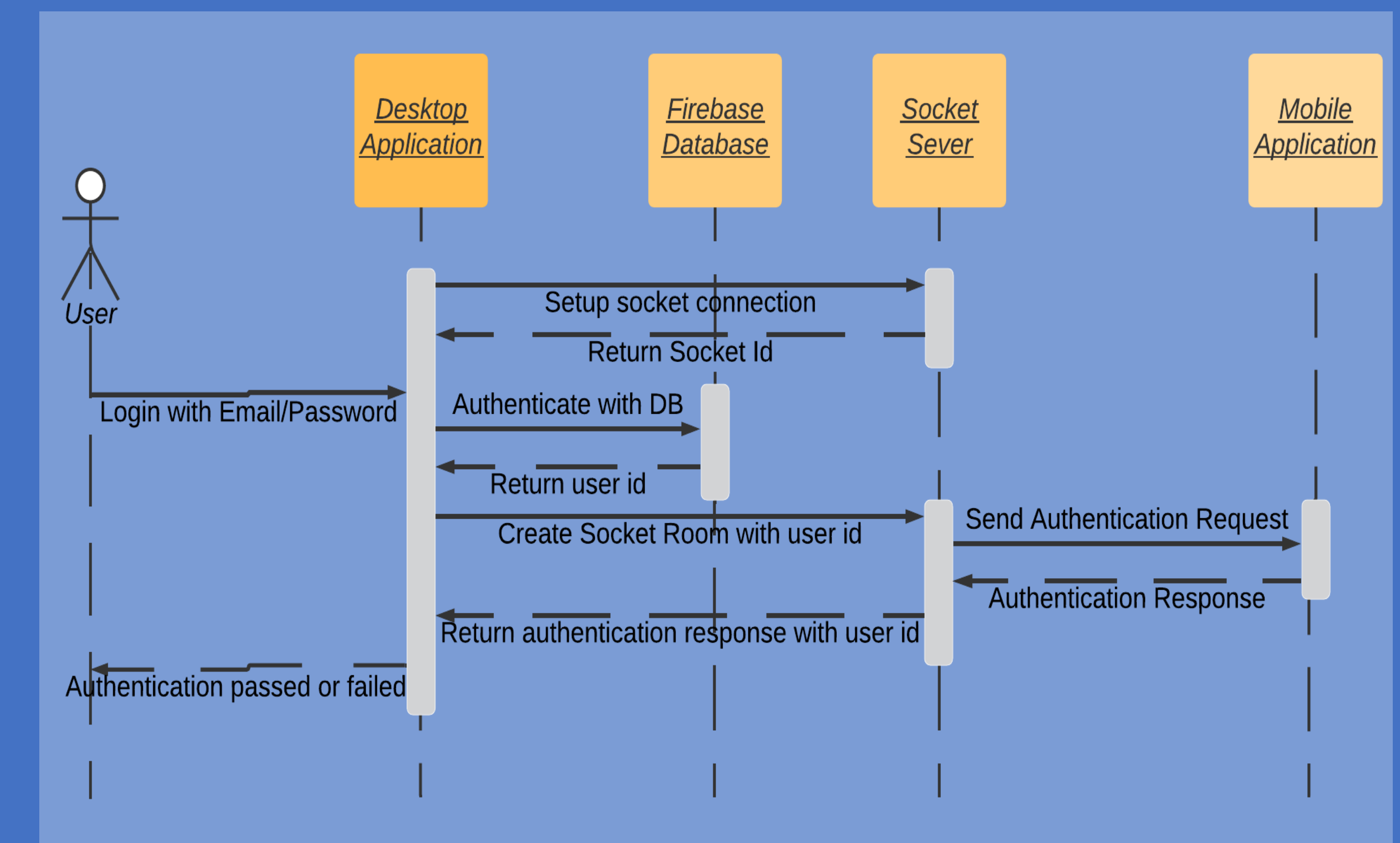
Being hacked and losing tons of information is a common event these days at both the company and individual level. Typically this all starts because somebody managed to get into your account by simply brute forcing your password. To avoid this a lot of companies set up counter measure but one of the most useful is 2 factor authentication. We look to use the same concept as 2 factor authentication but rather then have some code sent to us in order to authenticate the second step we are taking it a step forward and using biometrics to identify the user. Doing this would hopefully allow us to further secure companies and individual systems in order to better protect user/employee data.

Requirements

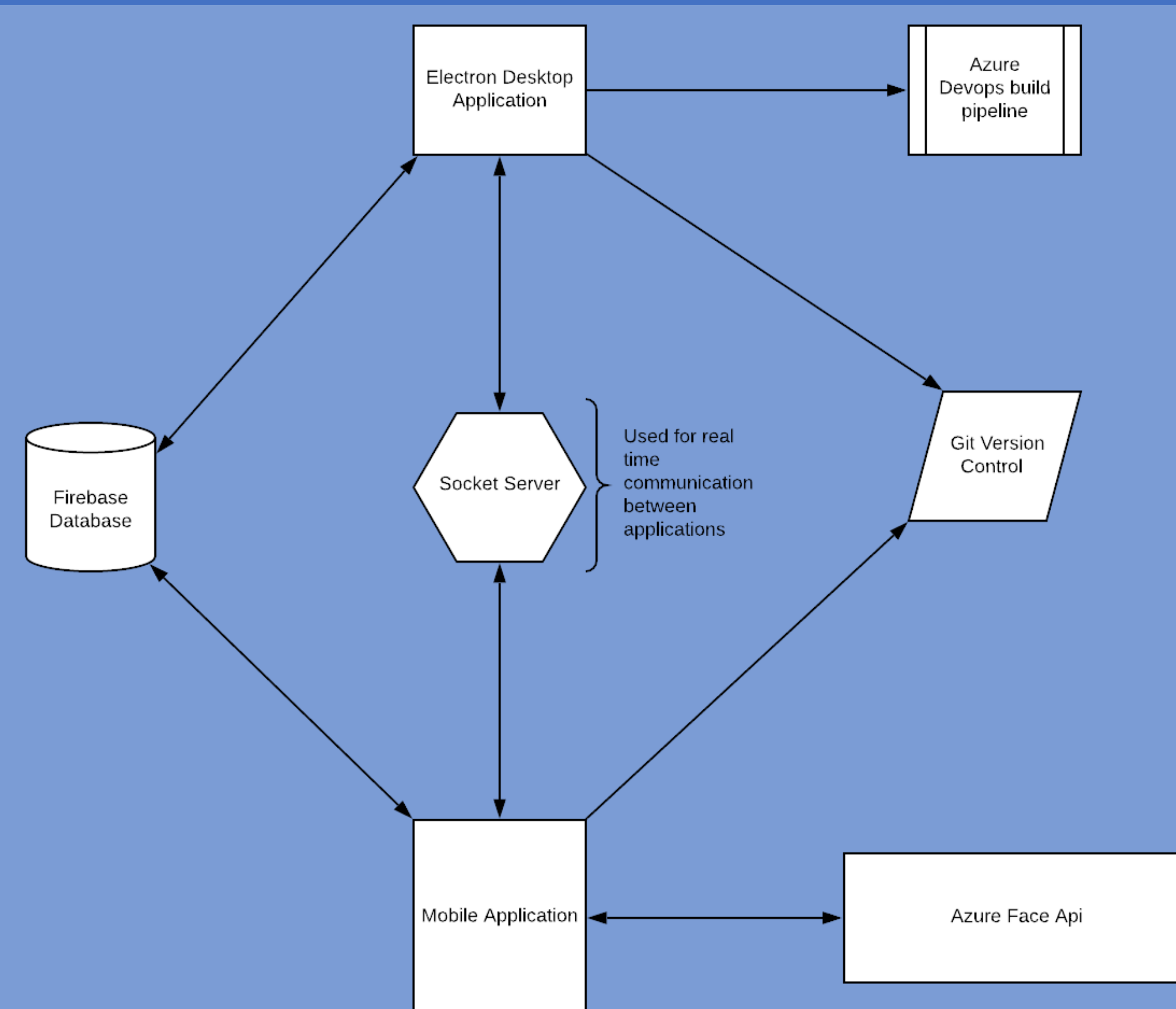
This project was divided up into two sections: A desktop and mobile application. For these requirements I will be talking exclusively about the desktop application.

- Allow the user to login through the desktop application
- If the user doesn't have an account prompt them to create one
- Establish a socket connection with the desktop application
- Create socket room with unique key for desktop application to send private messages to mobile application
- Once user logs in with valid information you must instruct the mobile application in real time to authenticate with faceld
- Allow the desktop application to take back the real time response from the mobile application on whether the user succeed or not.

Login Sequence Diagram



System Design



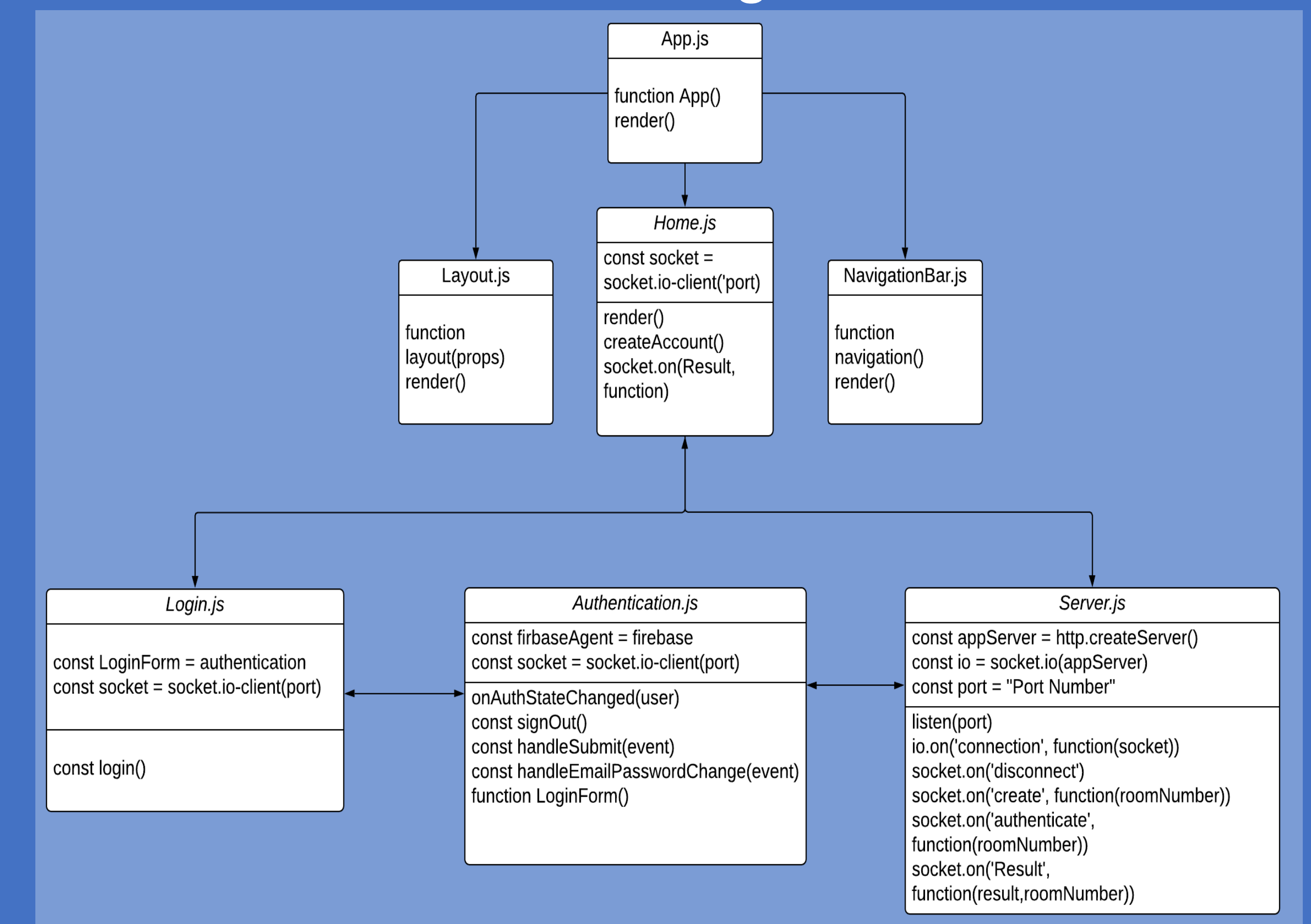
Implementation

The desktop application was built using React and bootstrap for frontend. We coupled this together with electron in order to make it exclusively a desktop application. For the database we used firebase for both the desktop and mobile side and used its authentication process to login and out users. Lastly in order to establish real-time communication between the two applications we used SocketIO to create our http socket server and SocketIO-client to allow our applications to connect.

Improvements

- Improved User interface for better user experience.
- Unit and integration testing.
- Admin portal access in order to add and remove users
- Push notification setup for when the app in closed on the users phone

Class Diagram



Acknowledge

We would like to thank our product owner, Rachelle Tobkes for all the advice and guidance she gave us. I would also like to thank my teammate Frederick Husson for all the hard work he contributed in order to finish this project.