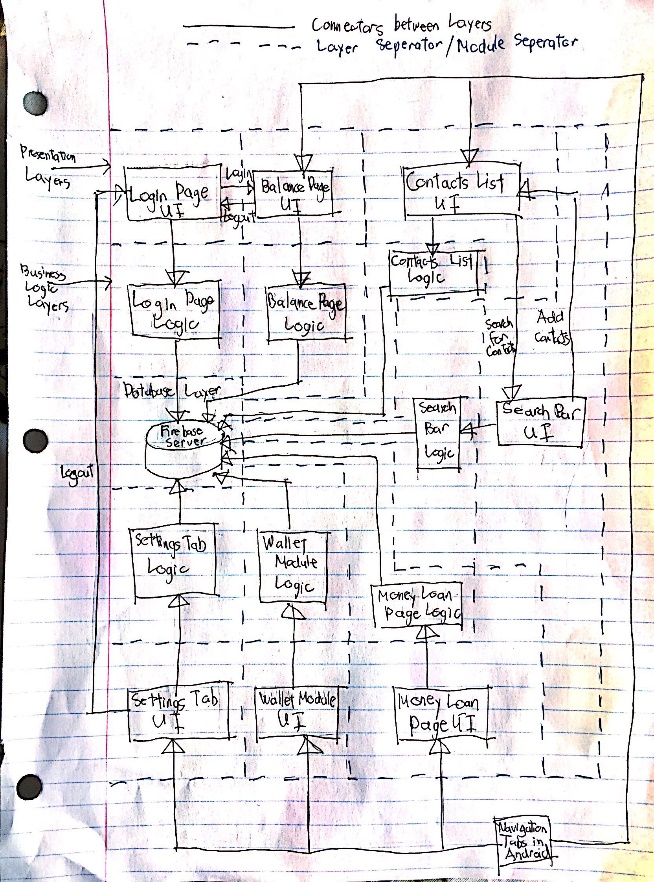
CS 446 Deliverable 5

Shanglin Ye, Zijian Wang, Shuangyou Huang, Haoqing Tian

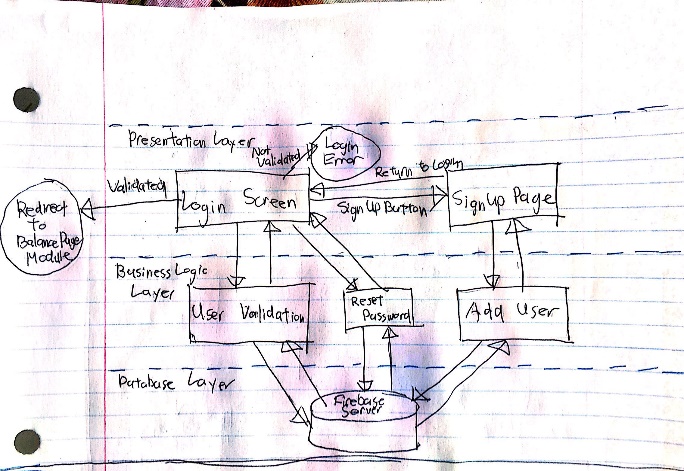
slye, z2495wan, s257huan, h33tian

**Architecture**

While researching the best architectural style for our project, we decided that the layered architecture or more specifically the 3-tiered architecture is best suited for our needs. More specifically, our system will be split up into 3 distinct layers. Our top layer will be the presentation layer and will consist of all the UI components that any user will be able to interact with. Our second layer will be our business logic. This will consist of all the logic that is required by the system to meet it’s functional and non-functional requirements. The business layer is also coupled with the persistence layer because we decided that it is more efficient for our business logic to interact with database objects rather than making calls to database objects in another layer. The final layer is the database layer. This is the layer that stores all our data is an external service that our business layer interacts with.

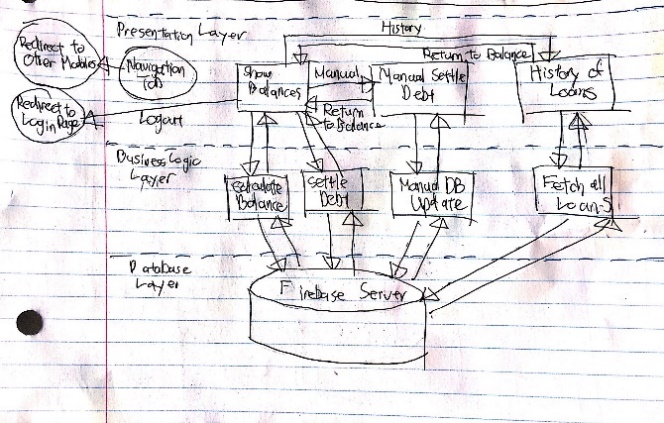
Our architecture style is can be further represented as 7 individual modules that all consist of the 3 distinct layers mentioned in the previous paragraph. Our system has 7 main functional properties that require little or no interaction with any of the other functional properties. This include the login page, the search bar, the contacts list, the balance/home page, the money loan page, the wallet module and the settings tab. Each of these modules has separate UI and each will represent the presentation layer of the respective modules. The only interactions that each presentation layer will have with another presentation layer is to redirect them to that module. For example, when you login in, you will need to be taken to the home page afterwards. The business layer of each module is separate from any other module’s business layer. They operate completely independently of each other. The database layer will be shared among all 7 modules since they will be using the same data. The rest of this document will be going into more detail about each individual module. Refer to the diagram to see how our system is structured.

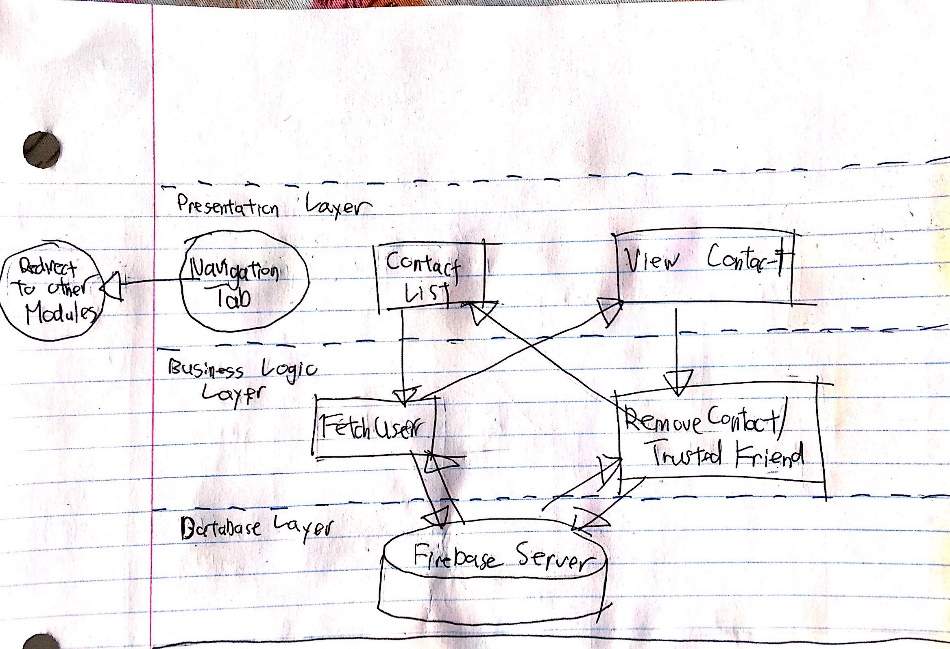
**Login Page**

****

Our login module has two separate pages, each represented as a component in the presentation layer. When a user opens the application for the first time, they will open the Login Screen. They will enter their username and password and click on the Login button. This button will make a call to the User Validation component which will check if the login information is correct. Then it will send a validation flag back to the Login Screen which will redirect the user to the balance page or report a login error. If the user forgot their password, then can click on the Forgot Password link which will send a request to the Reset Password component. This will handle the logic of sending a reset email to the user and subsequently update the database. On the login screen, there is a button to take you to the Sign-Up Page if the user doesn’t have an account yet. On this page, they will enter their new account information and click on Sign Up. This will send all the information to the Add User component which will store the new user in the database before returning to the Sign-Up Page with a success or failure message.

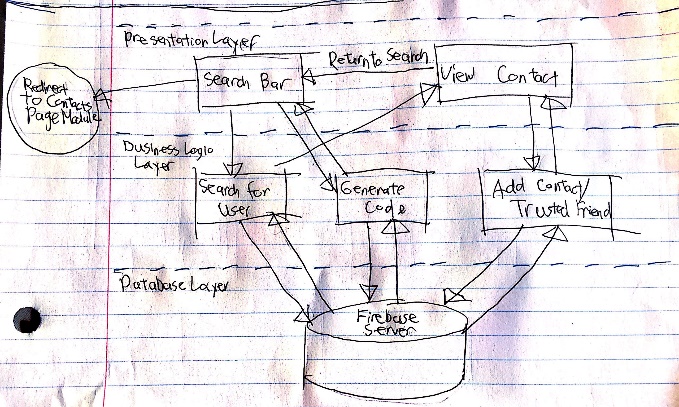
**Balance Page**

****After logging into the system, it will redirect the user to the Show Balances page which also serves as the home page. The page will load the balances by calling the Calculate Balance component. When the user clicks on their outstanding balance, the system calls the Settle Debt component and updates the balance with the money stored in the user’s wallet. There is a button that takes the user to the Manual Settle Debt page. They can find an individual user they owe money to and enter the amount they paid manually. This will make a call to the Manual DB Update component. Additionally, the user can click on history button that will take them to the History of Loans page. This page will call the Fetch all Loans component which grabs all the data from the database. Lastly, the logout button will redirect the user back to the login page

**Contacts List**

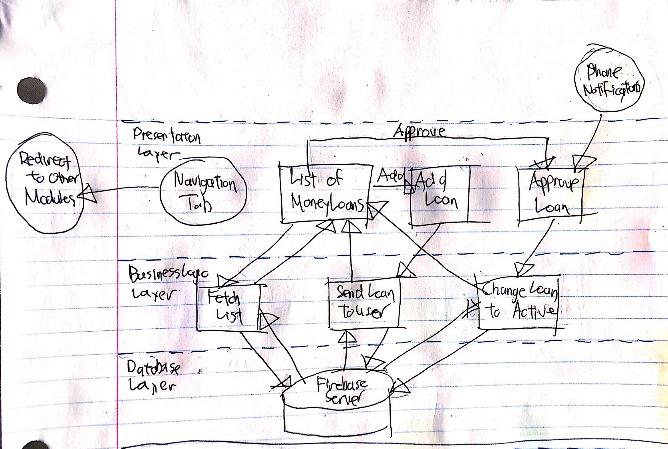
The Contact List page is the default page that is opened for the Contacts List module. On this page, all the user’s contacts and close friends are listed out. This is done by making a call to the database and fetching all the friends of the logged in user. If any of the names on the contact list are clicked on, the Contact List page sends the name to the Fetch User component. The component grabs all the desired user’s information from the database and return the information to the View Contact Page which will display the info. On this page, there is a button to either remove the user as a contact or trusted friend. If this button is clicked, the system sends the user’s info to the Remove Contact/ Trusted Friend component which updates the database so that both user’s friend lists do not contain each other anymore. After this action is complete, the component will redirect the user back to the Contact List page. Lastly, click on the search bar will take the user to the Search Bar module.

**Search Bar**

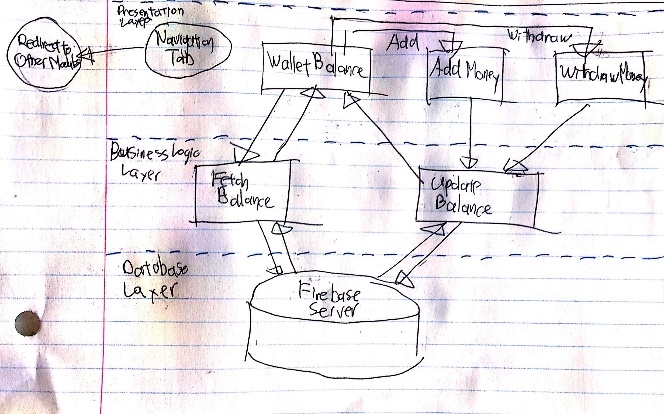
****

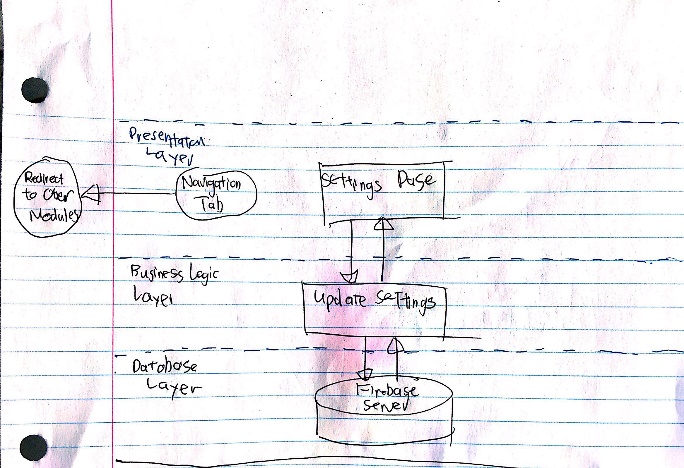
The search bar is specifically used to look up users in the system to add to your contact/trusted friend list. When you enter a name/id in the Search Bar and click on search, it will dynamically generate a list of all the users with similar names/ids. If you click on any of these users, the information will be sent to the Search for User component which looks up the user’s information and redirects the user to the View Contact page which displays the user’s info. On this page, there is a button to add this user to either your contacts or trusted friend list. Clicking on this button calls the Add Contact/Trusted Friend component which send a request to the user in question which they can accept or decline. The View Contact page also displays all your outstanding friend requests with the option to accept or decline. Lastly, there is a button on the View Contact page to go back to the Search Bar. If you X out of the search bar, you will go back to the Contacts Page module.

**Money Loan Page**

**** The List of Money Loans page is the default page of the Money Loan Page module. Upon loading this page, the Fetch List component will grab all the loans of the currently logged in user and send it to the List of Money Loans page which displays it to the user. There is a button that will take the user to the Add Loan page. A new loan’s information can be added here and after clicking send, the page will call the Send Load to User component which will send a loan request to the other user to be approved. Afterwards, they will redirect back to the List of Money Loans page. There is also a button to go to the Approve Loan page. There is a list of all outstanding loans and clicking accept button will send the request to the Change Loan to Active component and add the loan to the database. The user will once again be redirected to the List of Money Loans page. This page can also be accessed from your phone notifications.

**Wallet Module**

**** The Wallet Balance page is the default page of the Wallet Module. Upon opening the page, the Fetch Balance component will be called which grabs the logged in user’s wallet balance which is sent back to UI to be displayed. There are two other buttons on this page. One for adding money and one for withdrawing money. These buttons will take the user to the Add Money and Withdraw Money pages respectively. When you enter an amount on either page, they will send the request to the Update Balance component. The money information in the database is updated and the user is redirected back to the Wallet Balance page

**Settings Tab**

The Settings Page is the only page in the Settings Tab Module. This page lists out all the user’s settings. If any of the settings are changed, the system automatically calls the Update Settings component which updates the user’s settings in the database. The user can also logout of their account from this page.

**Navigation Tab**

As you may have noticed, there is a navigation tab in the presentation layer of most the modules which can redirect you to other modules. This tab is static and remains on screen while navigating between the different modules/pages. It is displayed at the bottom of every page. There tab consists of 5 buttons (5 different modules) and clicking on any of the buttons brings you to the module in question. It is easy to modularize the different sets of functions since they mostly go through this tab to switch to a different set of functions.

**Firebase Server**

All the different modules share a database layer which is labelled as Firebase Server in all the diagrams. Firebase is an external service that we use to store our database information. In the business logic layer, we make API calls that directly add, update, delete or select information from the database. We can also directly access the firebase database from their website and view or make changes to the data there as necessary. Even though this is an external service, we still included it in our architecture since it is crucial for our system to run.

**Design**