**ECE 651: Foundations of Software Engineering**

**Prototype Demo**

**Group 3**

**Zhechen Du 20340480(z6du)**

**Ke Chen 20456026(kchen)**

**Shangru Li 20637274(s488li)**

**Hongming Wang 20667749(h539wang)**

**Zhao Zhang 20689699(z664zhan)**

After the first delivery, we have had 6 meetings. We divided our work to individuals and integrate the results to test outcomes. Till now, the backend and frontend designs are both on the right track.

Connector Layer

We use connector layer to connect the Data Access Object layer and the clients. In this layer requests from clients are received, data from the Data Access Object layer are transferred to clients and requests are handled. JSON is our data format to transfer data objects and all the JSON-data which will be sent is encrypted in this layer. In this layer we call corresponding Action class in the Data Access Object layer that controls the database system. To make sure this part works well, we will firstly realize some simple functions such as register and login. And then we will finish all of the functions in this layer.

Data Access Object layer

On the data layer, we created a relational database as the data container, and we also created a common data access object layer for all kinds of data operations. We use hibernate4 as framework, and apache DBCP as the database connections pool. Now we have finished the data access object component, and the challenge is using java reflection and generics to create a common component for all entities. After building up the database, we used Uniform distribution and Normal distribution to generate data and simulate hundreds of users account information, balance status and transaction history in R. Also, we use the MD5 hash to encrypted the generated data for database. Next step, we would combine the service layer with data access object layer and finish all transaction and logic about this system.

Core Trade Algorithm

We have created the framework for trade algorithm. Three concurrent memory databases are used for real time trading. Currently, only the curreny\_pool database is implemented, we have created rules that is used to update the pool. Furthermore, the algorithm used to match trade is also implemented. Next step, we will implement the other two database and the connection between the three.

Androids End

Welcome activity and launch activity have been implemented. DialogFragment class is used in Launch activity to display log-in and sign-up modules as dialogs. This design strategy reduces the complexity of architecture of front end by restricting the number of activities in application. TextWatcher is set on all of the edit views to dynamically detect the validity of inputs and pops up proper prompts to users. And we are currently designing the main interface in a style similar as WeChat’s UI, which divides all function modules into four fragments held by only one container. This conception of design makes UI simple to users. To attain some functions such as “Real time exchange rate”, “Financial news”, we integrated API for androids end, and this step is in the process.

trade.png