**Design Document for CyBank**

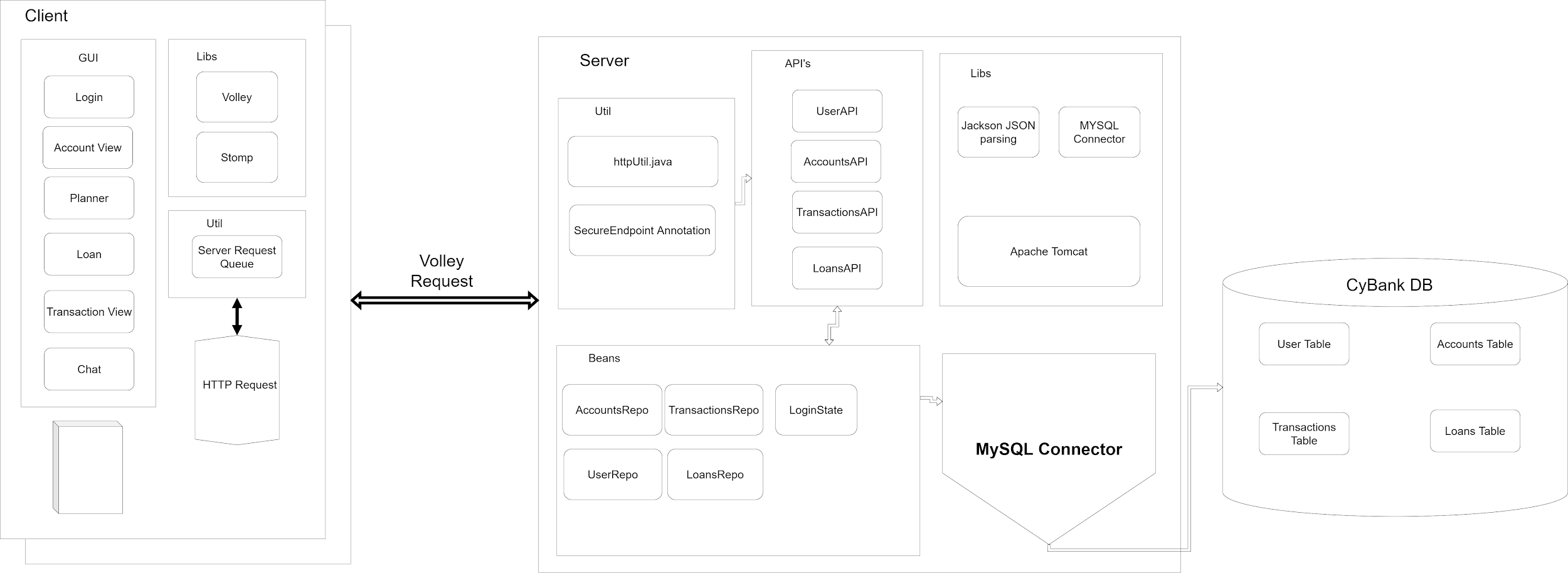
Group <VB-1>

Murathan Yildirim: 25% contribution

Raj Singh: 25% contribution

Jordan McGhee: 25% contribution

Cutler Thayer: 25% contribution

PUT THE BLOCK DIAGRAM PICTURE ON THIS PAGE! (Create the picture using pencil or drawIO)

Design Process:

Model:

Our application uses MySQL for database, SpringBoot framework for server-side connections with the database as well as Android front-end. Libraries for server-side contain JSON parsing(Jackson), MySQL connector, and Apache Tomcat. Our APIs are reliant on two types of beans, login state bean, which we use for holding user info, and the default JPA repo beans, which we use for database connections. The default way to set up the repos is by using them as beans, that way we don’t have to re-initialize our database connection every time we want to use it. Server-side utilities that are used to communicate with the front-end include HTTPutil and SecureEndpoints to make sure the message goes across. For front-end side utilities we use volley to communicate with the server. So here is a rundown of how things work: From the front-end, the user makes a request. It can be a get request or post request, then volley makes an HTTP request to the server. When the server receives a request, its API sends the information via repo beans which updates/generates tables in the database. The database response is passed through our APIs, which then get sent to the front-end as a response.

View:

We try hard to make the application close to real-world banking applications, therefore we try to make the UI simple and clean. Because this is a banking application, time is a very important aspect of it. We use the time for almost everything: to make future transactions, keeping track of transactions, keeping track of messaging in real-time, and to determine when users registered. Other than time, we use bottom navigation as well as top navigation for smooth navigating through screens. We made our own XML to make the accounts/transactions look rectangular with the information about them as well as a picture for the category of the transactions. With the nature of the banking applications, we get a lot of information from the user to determine whether a user is eligible for an account/loan. Therefore, we use a lot of input values(EditText) and alerts. We communicate with the back-end frequently using volley requests(using our helper methods), as well as tracking user-session information for security purposes.

PUT THE TABLE RELATIONSHIPS DIAGRAM on this fourth page! (Create the picture using MySQLWorkbench)

