Design Document for Cynance

Group <3-Rasel-4>

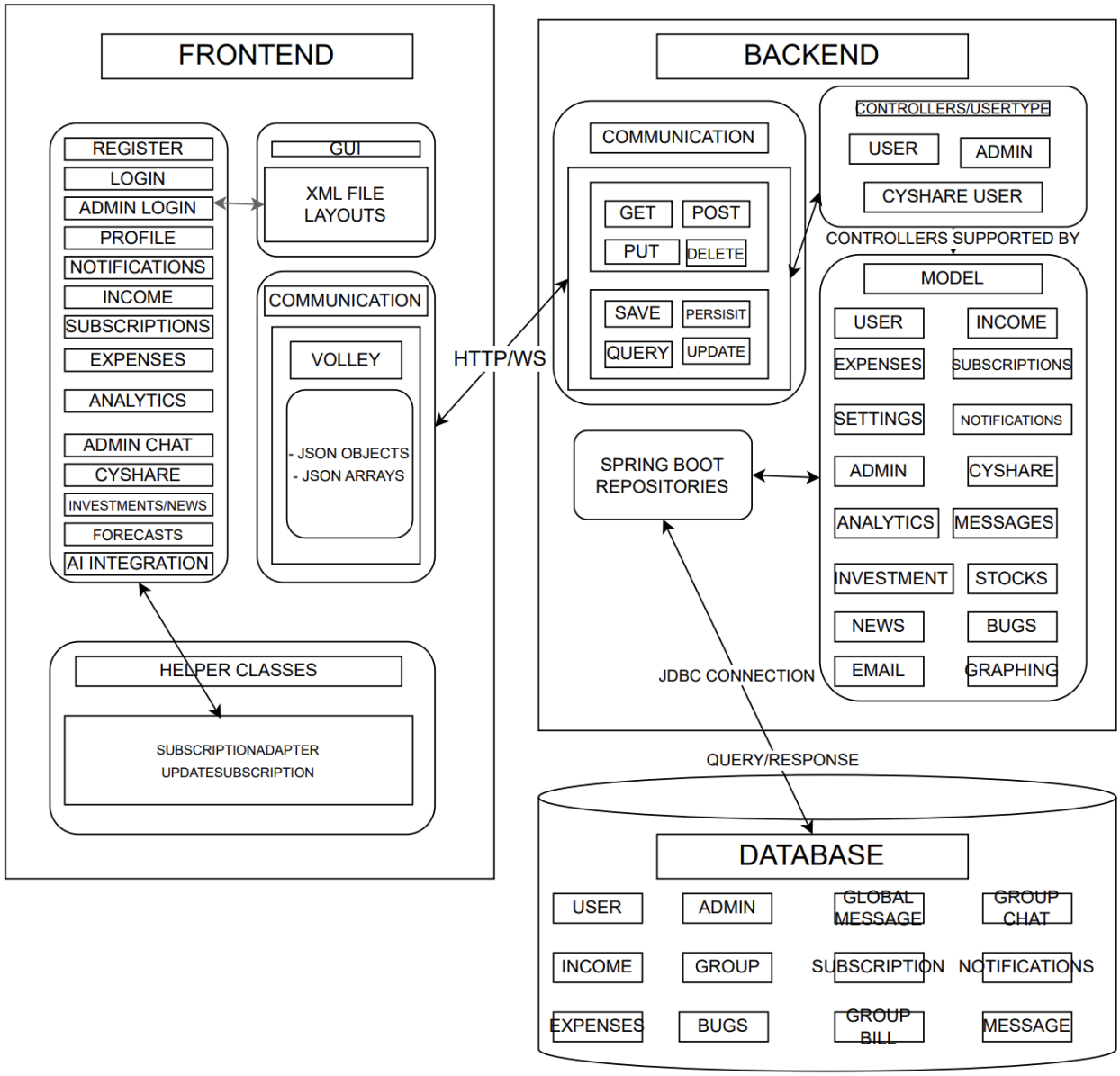
Aadi Shah: 25% contribution

Tanisha Magikar: 25% contribution

Shubam Chaudhari: 25% contribution

Nikhil Kumar: 25% contribution

BLOCK DIAGRAM



BLOCK DIAGRAM DESCRIPTION

FRONTEND:

The frontend of the Cynance application using Android Studio allows different types of users to access the app and perform their designated actions in the app. A regular user can register into the app via the corresponding button, and then login. The homepage for the user offers a whole host of functions, including an income tracker, subscription tracker, expenses tracker, access to a settings page and others. These other features include a notifications tab (with live notifications supported), an analytics page with graphs, a live splitwise feature with group creation and a fully functional chat available, as well as the ability to chat with the admin to report people and bugs. In addition to this, a different type of sub user is the group admin in the split wise feature of the app, which we are calling CyShare. The group leader is the one who obviously created the group, they also have the power to remove people from the group. Additionally there is the admin login, which only allows a set of predefined usernames and passwords to access. The UI for this involves the ability to respond to the global chat, report bugs that need fixing, and banning users that have been reported. All of these features are supported to HTTP and WS connections with the Backend done via Volley.

BACKEND:

The Backend uses SpringBoot with Apache Tomcat to run the server which is connected with the Database, and supports all the functionalities of the frontend. The backend consists of several packages that are used to implement all the functionalities that the frontend supports. These packages include Model, Service, Controller, Repository and Exception. The Model layer defines the different types of entities in the database layer, and how each of these interact with each other. It also has the getters and setters for each of the classes. Service layer is responsible for the core functionality and logic of the application. It supports the controllers that have the urls that connect to the application, through which certain actions can be performed and decided to be true or false. The repository layer is what is used to connect with the database, and this is where all the saving is done. The exception handling is done in the Exception package. This obviously connects to the database.

DATABASE:

The database stores all the entities and relations of the entire program. It is a MariaDB database and via queries and responses from these, we are able to get the desired results of the program.

ENTITY RELATIONSHIP DIAGRAM