

Design Document for CyRate

Group 1_JK_4

Alex Ong: 25% contribution

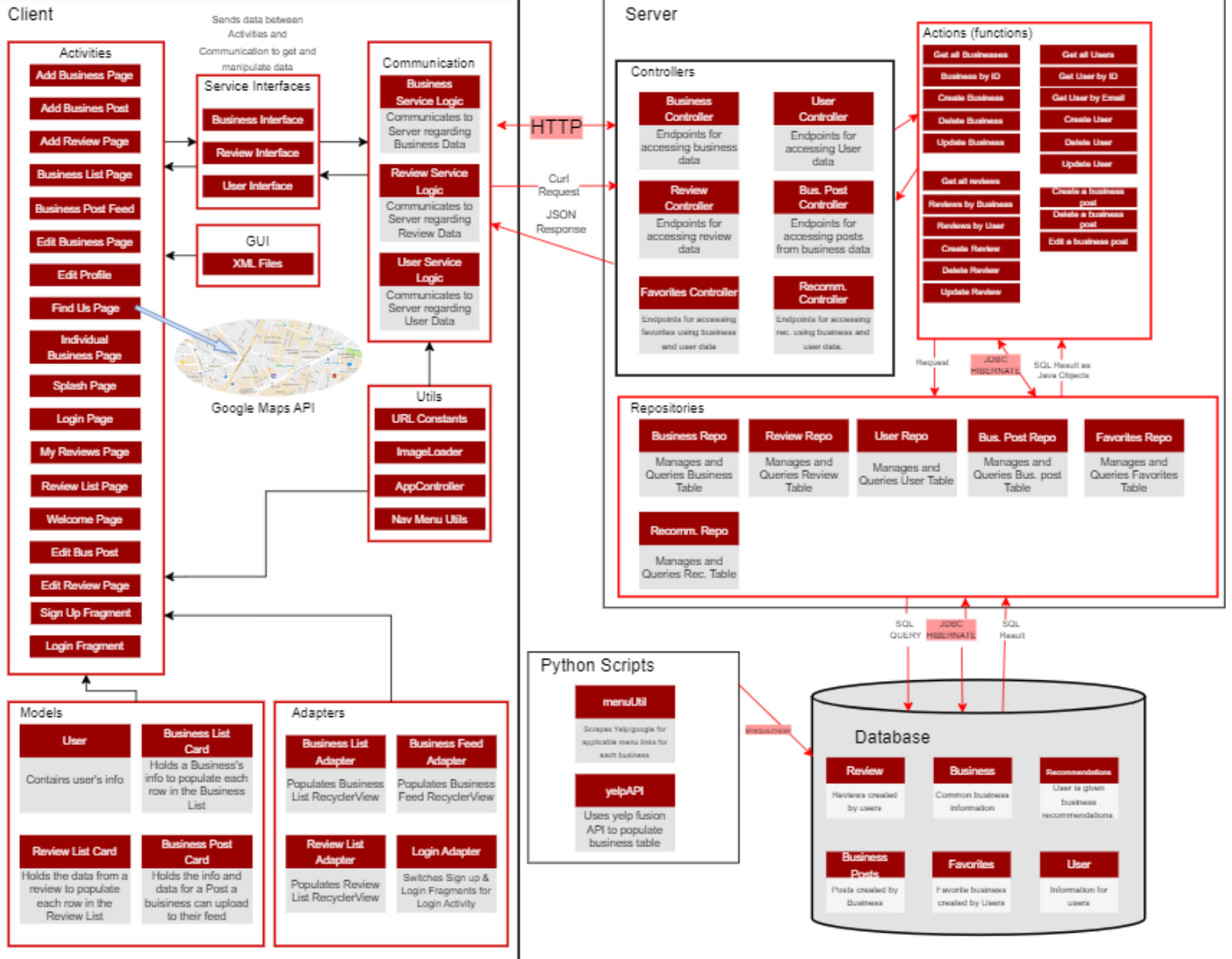
Megan Eberle: 25% contribution

Sam DeFrancisco: 25% contribution

Anbu Krishnan: 25% contribution

CyRate 1_JK_4

Alex Ong, Megan Eberle, Sam Defrancisco, Anbu Krishnan



Design Description

Android User Interface

Our CyRate application has 17 Activity classes along with 2 Fragment classes to make up the User Interface. Each activity has a corresponding XML file which describes the styling of the page. Along with the Activity and Fragment classes, we also utilize Util classes, Model classes, Adapters, and Interfaces to control the UI. Adapters are used to populate pages containing dynamic scrollable lists and other features. We use interfaces to communicate between the UI and Service Logic to retrieve and post data.

Android Models

We have utilized Model classes for many different objects. These include 2 different Business models, a User model, a Review model, and a Business Post model. These are all used to store information about specific Businesses, Users, Reviews, and Business Posts, respectively. They help us display the correct data and transfer the correct data between pages.

Android Service Logic

Our Service Logic classes act as our Controller to communicate with the Server. These classes include Business Service Logic, User Service Logic, Review Service Logic, and Favorites Service Logic, which each contain logic for making requests to the database for information from the corresponding table. The logic in these classes allows us to make GET, POST, PUT, and DELETE requests for each table.

Backend/Server

Our backend was created using Springboot & it's built in Apache server. The whole backend can be broken into components.

Models :

For each table in the database we created a POJO (plain old java object). These contain instance variables that match the table fields as well as getter and setter functions.

Controllers:

Controllers are responsible for receiving the requests made by the frontend. Each table in our database has it's own designated controller for handling get/post/put/delete requests. Within a controller class each function has a designated URL attached to it (endpoint). These functions utilize the repositories to retrieve and modify data.

Database:

Our MySQL relational database includes all of our stored information in the form of tables. Examples of these tables are the User table, Business table, Reviews table, Favorites table, Recommendations table, and Business Posts table. The database can be used for storing, retrieving or manipulating data. The tables can have relationships between them, for example the reviews table utilizes data from the user table and business table.

