Design Document for Cy-Cook

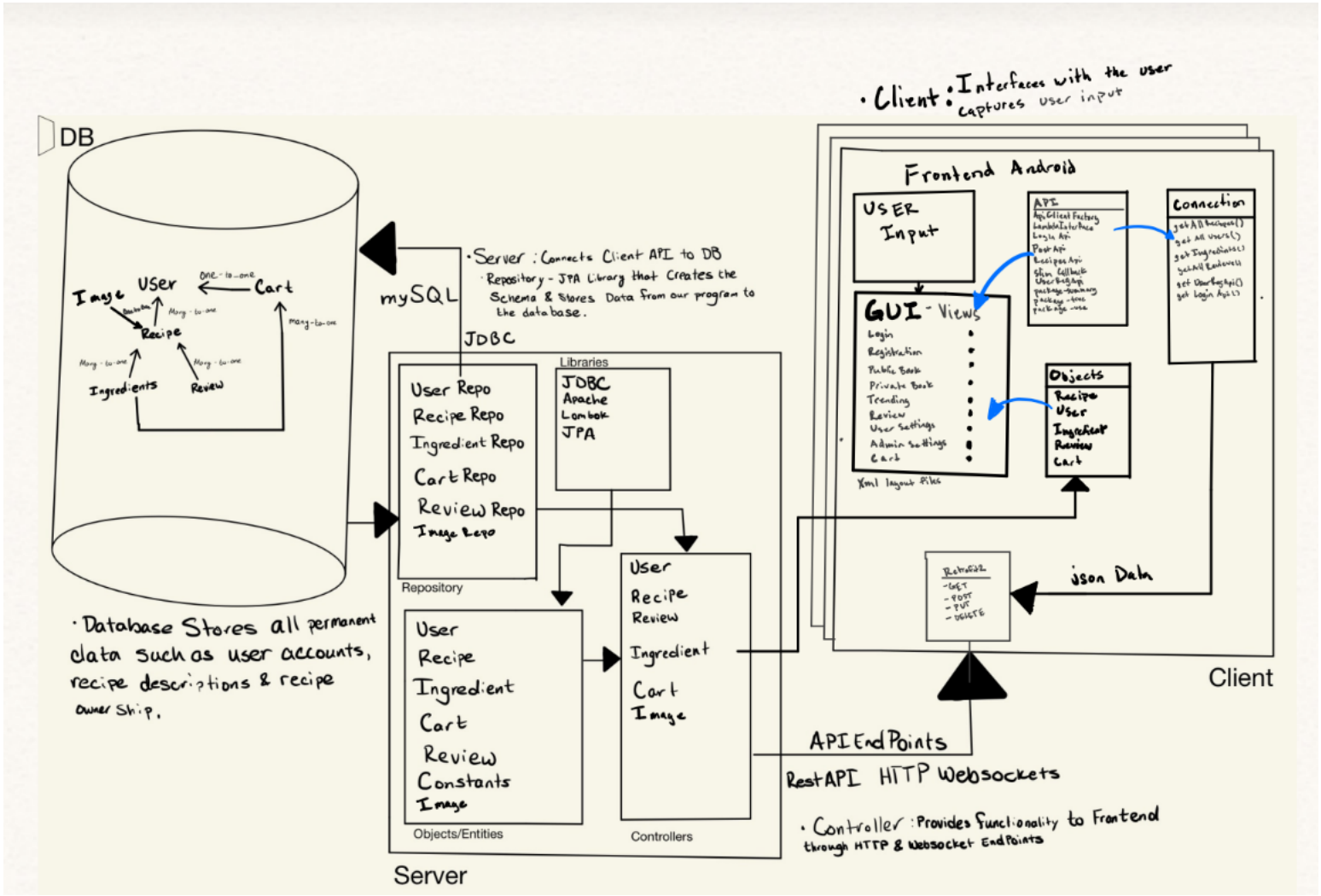
Group 1_JK_8

Jason Brittain: 25% contribution

Clay Kemper: 25% contribution

Robert Holeman: 25% contribution

Eranda Sooriyarachchi: 25% contribution


• Client : Interfaces with the user captures user input
DB
Image
User
Cart
Recipe
Ingredients
Review
one-to-one
Many-to-one
• Database Stores all permanent data such as user accounts, recipe descriptions & recipe owner ship.
mySQL
• Server : Connects Client API to DB
• Repository - JPA Library that Creates the Schema & Stores Data from our program to the database.
JDBC
Libraries
JDBC
Apache
Lombok
JPA
User Repo
Recipe Repo
Ingredient Repo
Cart Repo
Review Repo
Image Repo
Repository
User
Recipe
Ingredient
Cart
Review
Constants
Image
Objects/Entities
User
Recipe
Review
Ingredient
Cart
Image
Controllers
Server
Frontend Android
USER Input
API
Connection
GUI - Views
Login
Registration
Public Book
Private Book
Trending
Review
User Settings
Admin Settings
Cart
Xml layout files
Objects
Recipe
User
Ingredient
Review
Cart
Retrofit2
- GET
- POST
- PUT
- DELETE
json Data
Client
API EndPoints
RestAPI HTTP Websockets
• Controller : Provides Functionality to Frontend through HTTP & Websocket EndPoints

Use this third page to describe complex parts of your design.

The Database consists of multiple tables that have either one to one or one to many relationships. A single user can have multiple recipes. Therefore, a one to many relationship exists between user and recipe list. Each recipe has multiple ingredients, therefore, there exists a one to many relationship in that aspect. Then there is a one to one relationship between image and recipe, so there is an image for each recipe. Then there is a one to one relationship between cart and user, because it wouldn't make sense for a user to have more than one cart. Then each cart has multiple ingredients, so there is a one to many relationship between cart and ingredients.

Our design uses a simplified GUI layout consisting of one main activity and multiple fragments made with similar layouts. The Cart, Private Book, and Public book consist of a singular linear layout that uses animation files to smoothly scroll along the page. The login page determines the type of actor that is using the app. If the login screen is bypassed by using the menu button then users are defined as a guest and cannot post recipes or reviews. If a valid login is typed in on the login screen then a user will be logged in and gain access to posting privileges and user settings. If a specific valid admin login user is used then the Admin User will gain access to edit a recipe fragment where public recipes can be deleted from the public book.

The post recipe API gathers information in the recipeName, description, ingredients, and image fields to a new recipe id and enters them into a recipe entry in the recipe table. This table is visible to both users that created it in their Private book as well as to all users in the public book. The user registration works in a similar manner when a new email, username, and password are entered into the registration screen their login information will be stored in the users table on the server.

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