Block Diagram

CampusEats Food Delivery App

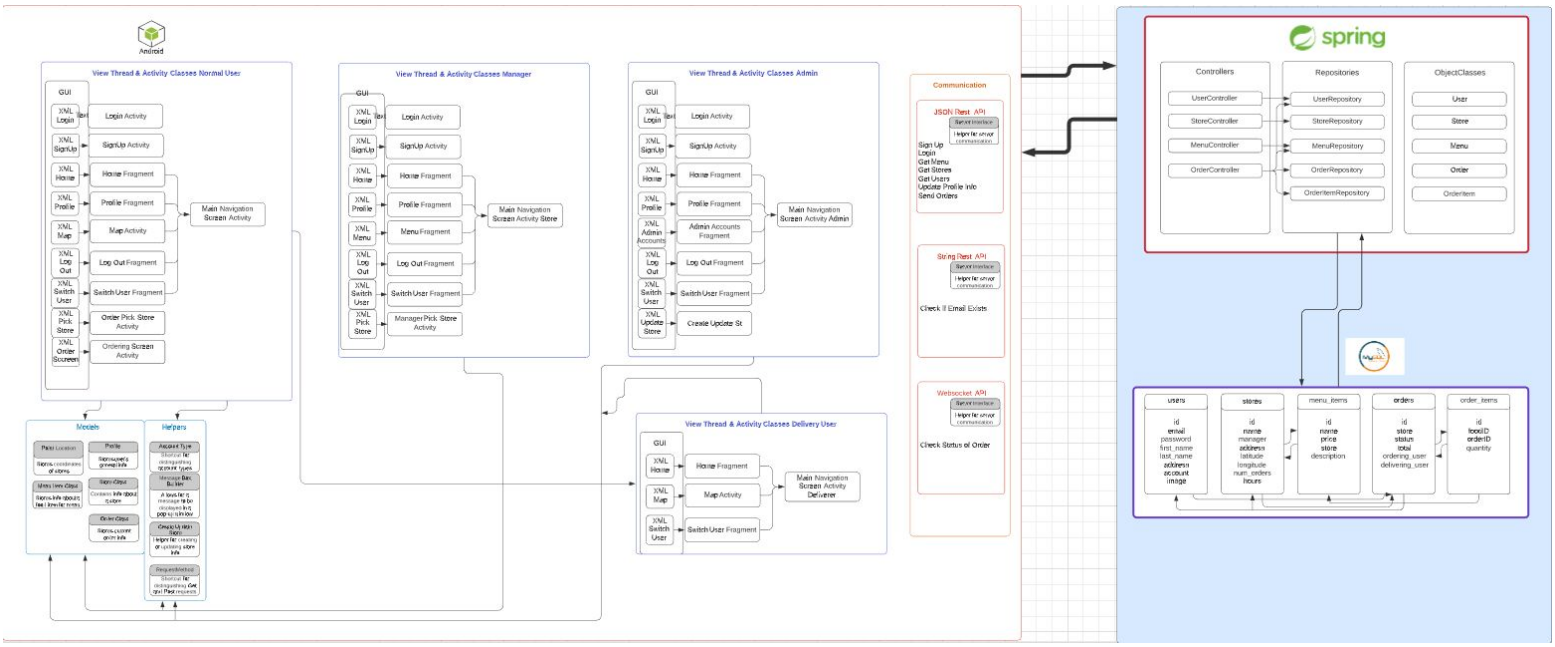
Group MC_08

Devin Milligan: Frontend/Structural Development

Ryan Hickok: Frontend/UI Development

Allison Finger: Frontend/UI Development

Ty Wallis: Backend/Database Development



https://lucid.app/invitations/accept/437b7ca8-22f3-4f8c-9e43-22b3efc3e384
(Use this link if the above image is unreadable)

Design Descriptions

Android: User GUI

The Android application will have a thread dedicated to the GUI called the “view thread” (existing within Activity Classes). Depending on which type of account the user has (normal, manager, admin, delivery) the application will have different activity classes which are all listed within the “View Thread and Activity Classes” sections for the appropriate user type. Each activity will have corresponding xml files which will describe the layout styling of the page. Regardless of user type, there will be a main navigation screen activity which allows the user to navigate to all the activities which are available to him/her. The activities themselves will contain most of the logic processing of user commands, but for some of the activities that share similar commands, we will build helpers, models, and interfaces that will be used throughout the activities.

Android: Code Helpers

There are four major code helpers so far, which include the Account Type, Message Box Builder, Create Update Store, and Request Method classes. The Account Type class is an enum class that makes it easier to distinguish accounts with different types, and the Request Method is also an enum that distinguishes request types. In addition, the Message Box class allows for information to be displayed in a text box. Lastly, the Create Update Store method allows for admins to create and update store information.

Android: Models

The project includes five models that each contain information about different aspects of the app. The Profile class contains general information that any type of user must contain, and there are three subclasses of this class that store more information depending on which account type a user is. The Store class contains information about a store, including a menu that contains Menu Items with different food items that users can purchase. There is also an Order class that contains all of the information within an order that a user is placing. Lastly, the Point Location class stores coordinates for each store in the database.

Android: Communication

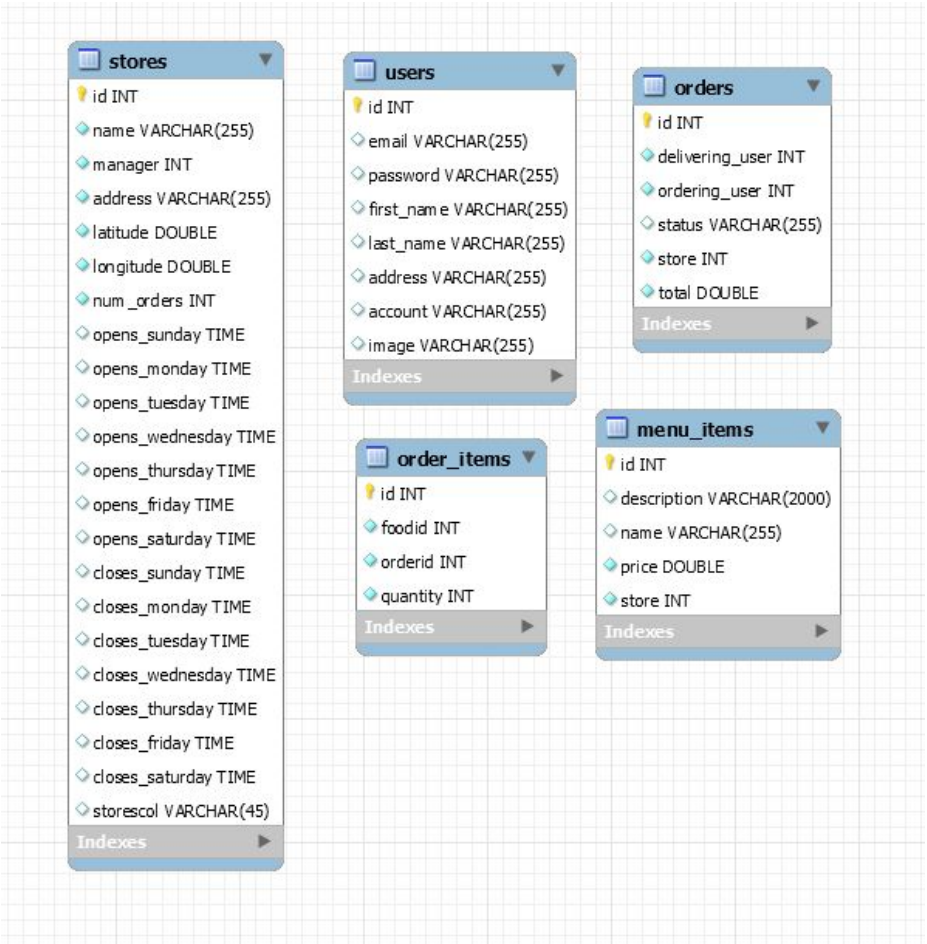
There are three different types of APIs used for communication between front and back end (JSON Rest, String Rest, and Websocket). JSON Rest methods return JSON objects (i.e. an array of store objects for GetStores) while CheckIfEmailExists within String Rest returns a string for whether or not an email account exists. Lastly, the websocket allows the user to check the status of his/her order.

Spring Framework

Our Spring framework consists of three main categories – controllers, repositories, and object classes. The controllers house a variety of functions for interacting with the database objects. The types of objects are users, stores, menus, orders, and orderitems.

MySQL Database

The tables within the database are users, stores, menu_items, orders, and order_items. These tables are interrelated in a variety of ways and contain fields that are valuable for frontend logic to be performed (see next page).


stores
id INT
name VARCHAR(255)
manager INT
address VARCHAR(255)
latitude DOUBLE
longitude DOUBLE
num_orders INT
opens_sunday TIME
opens_monday TIME
opens_tuesday TIME
opens_wednesday TIME
opens_thursday TIME
opens_friday TIME
opens_saturday TIME
closes_sunday TIME
closes_monday TIME
closes_tuesday TIME
closes_wednesday TIME
closes_thursday TIME
closes_friday TIME
closes_saturday TIME
storescol VARCHAR(45)
Indexes
users
id INT
email VARCHAR(255)
password VARCHAR(255)
first_name VARCHAR(255)
last_name VARCHAR(255)
address VARCHAR(255)
account VARCHAR(255)
image VARCHAR(255)
Indexes
orders
id INT
delivering_user INT
ordering_user INT
status VARCHAR(255)
store INT
total DOUBLE
Indexes
order_items
id INT
foodid INT
orderid INT
quantity INT
Indexes
menu_items
id INT
description VARCHAR(2000)
name VARCHAR(255)
price DOUBLE
store INT
Indexes