Bytez

Block Diagram

VB 02

Seth Gardner, Alex Freiberg, Zach Eisele, Noah Kelleher

Design Description

Views/UI

The Bytez application's frontend consists of Android views. These views use a combination of xml files and Java classes to formulate our user interface. The Login and Signup activities allow a user to enter their information to either login into the app or signup. The Friends and Friend Request activities show users their friends and friend requests or other users' friends. The Profile and Edit Profile activities allow users to view and edit their profile information or view others' profiles. The Home activity shows trending reviews of restaurants and is a hub to the other app functions. The Map activity lists all restaurants currently on the server and when clicked, opens the Google Maps API with that restaurant entered as the search field. The Review activity and Delete Review fragment lets users create or delete restaurant reviews. The Settings activity serves as a hub for the Report a Bug fragment, logout feature, and the Delete Review fragment.

Android Code Helpers

The two helper classes listed are SingletonVolley and SharedPrefManager. These are meant to help more with being efficient with the transfer of data rather than being used to build GUI and activities. SingletonVolley creates a single request queue to be used throughout the app for JSON requests. SharedPrefManager stores helpful user information that can be accessed at any place throughout the app. The information inside it is stored once a user logs in, and can be updated with specific actions throughout the app. Along with these two, we plan on implementing more fragments that are needed throughout the app to create a more modular design.

Android Communication

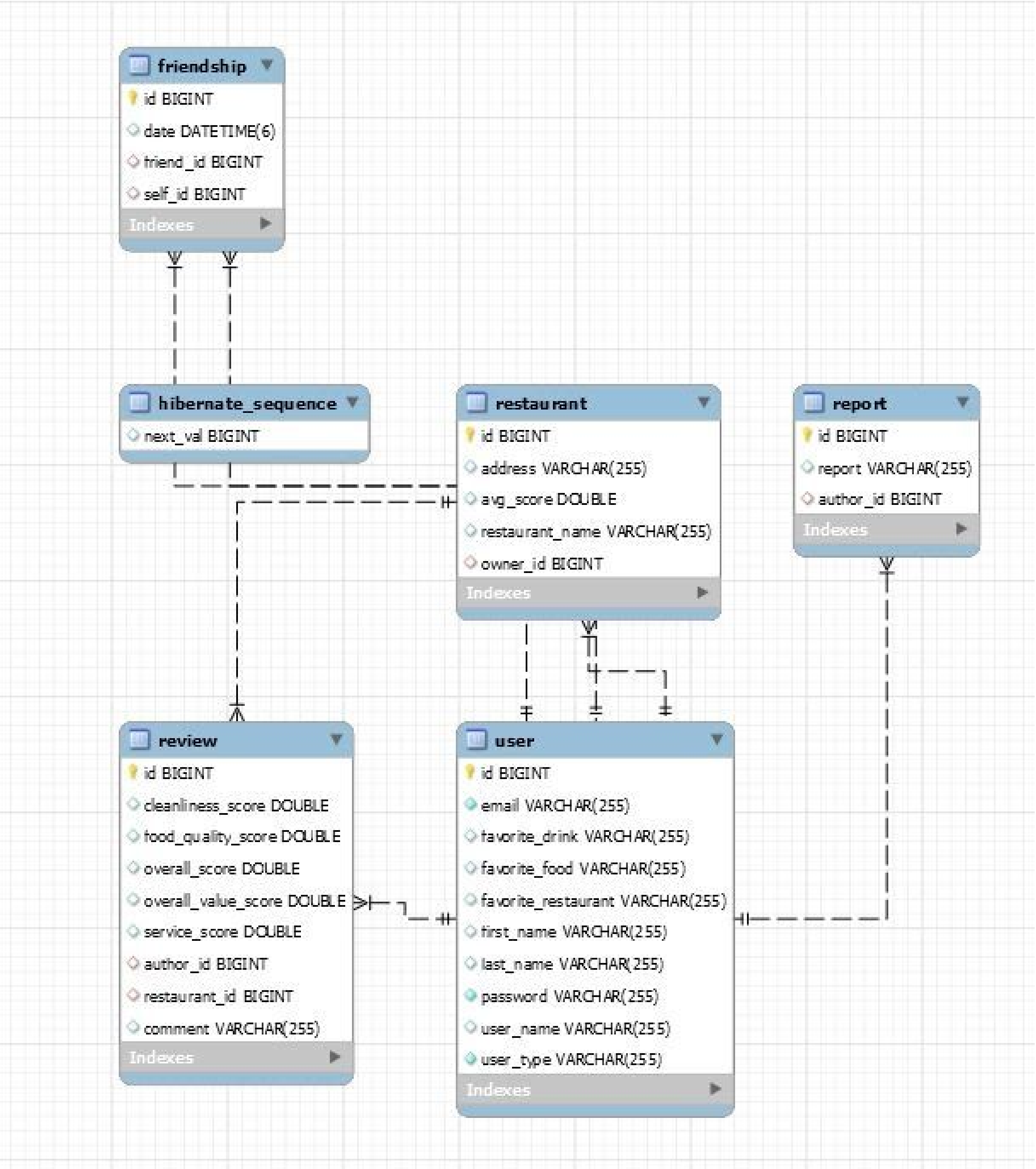
This section just shows all the JSON requests used throughout the app and what they do. This is how the front and back end communicate with each other.

Spring Communications

The backend is running a SpringBoot server with connection to a MySQL server. To interface with the frontend, our server exposes HTTP calls using REST controllers formatted in JSON. The controllers are then serviced by intermediate services. The services expose the JPA repositories where Hibernate is used to communicate with the MySQL database, which responds with an entity/s.

Database

The database we are using is a MySQL database containing 6 tables. Each table is linked to the other tables containing corresponding foreign keys. The User table contains all of the Users that are being used in the application and all of their information for login, restaurant ownership, writing reviews, etc. The User table is linked to the restaurant, review, report, and friendship tables using foreign keys. The friendship table contains all of the friendships created in the application. The Friendship table is linked to 2 different users contained in the user table to store a friendship. The Restaurant table contains all of the score, the address, the owner, and the name for each restaurant stored in the table. The review table contains all of the information inputted by the user to provide a review for a restaurant and is linked to the user who wrote the review and the restaurant the review is for. The report table contains any bugs reported within the application and is linked to the user who submitted the report. The hibernate_sequence table is generated by hibernate(Spring Boot Dependency).


friendship
id BIGINT
date DATETIME(6)
friend_id BIGINT
self_id BIGINT
Indexes
hibernate_sequence
next_val BIGINT
restaurant
id BIGINT
address VARCHAR(255)
avg_score DOUBLE
restaurant_name VARCHAR(255)
owner_id BIGINT
Indexes
report
id BIGINT
report VARCHAR(255)
author_id BIGINT
Indexes
review
id BIGINT
cleanliness_score DOUBLE
food_quality_score DOUBLE
overall_score DOUBLE
overall_value_score DOUBLE
service_score DOUBLE
author_id BIGINT
restaurant_id BIGINT
comment VARCHAR(255)
Indexes
user
id BIGINT
email VARCHAR(255)
favorite_drink VARCHAR(255)
favorite_food VARCHAR(255)
favorite_restaurant VARCHAR(255)
first_name VARCHAR(255)
last_name VARCHAR(255)
password VARCHAR(255)
user_name VARCHAR(255)
user_type VARCHAR(255)
Indexes