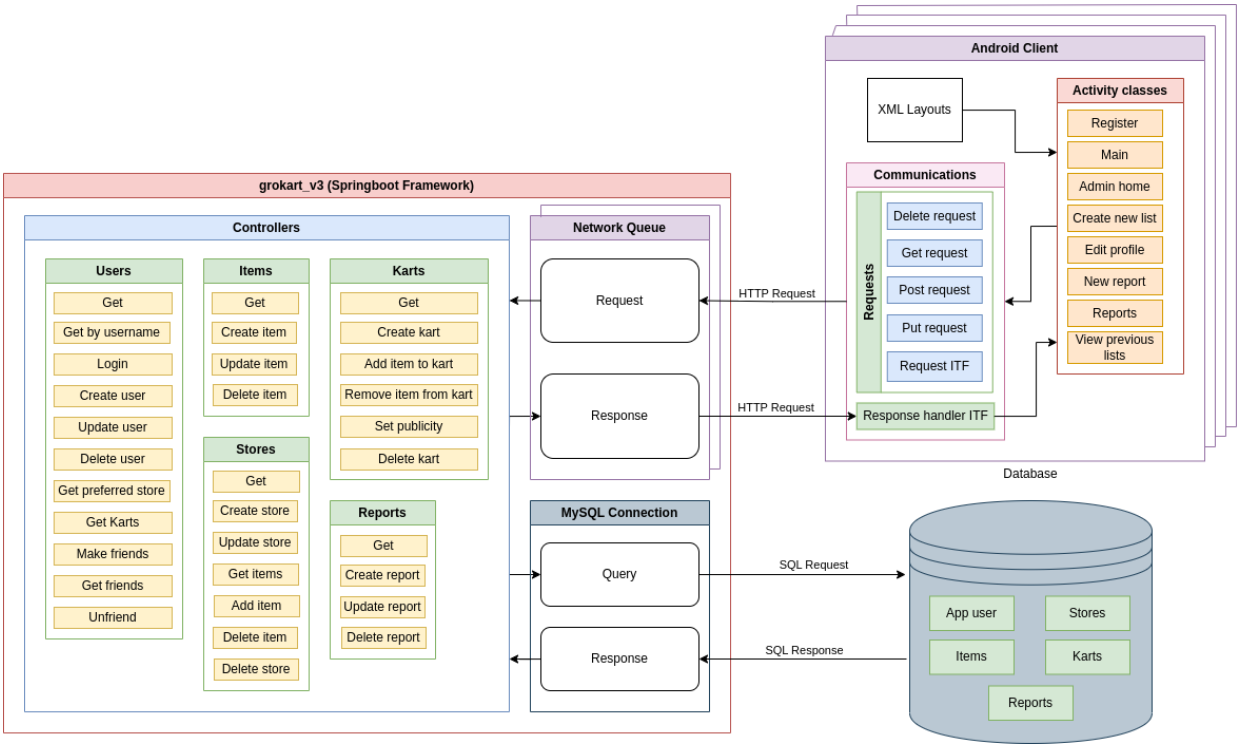
Block Diagram

Group: CW-03

Members: Baganesra Bhaskaran, Charles Dudley, Mattie McGovern, Joao Lira

Project Name: GroKart

Block Diagram



Description of the block diagram

Client Components:

XML Layouts

The XML layout files include a variety of xml files that are responsible for the design of the app

Java Classes

The Java classes hold the code that displays the design elements included in the XML layout files and dictates what happens when a user interacts with those elements. This includes when a user types into a textbox or when they click on a button. The java classes use the communication classes to connect to the backend.

Communication/Network Classes

Request Classes: These are helper classes meant to facilitate and modularize code that makes any type of request to the backend.

Response Handling: This is a helper interface, meant to be used as a parameter of a method. This interface has one method through which the user can specify how he wants to handle the response.

Server Components:

Controllers

The controllers are responsible for creating and editing objects such as items and karts. This includes set and get methods for different types of data regarding each object.

Network Queue

This is used to communicate with the front-end.

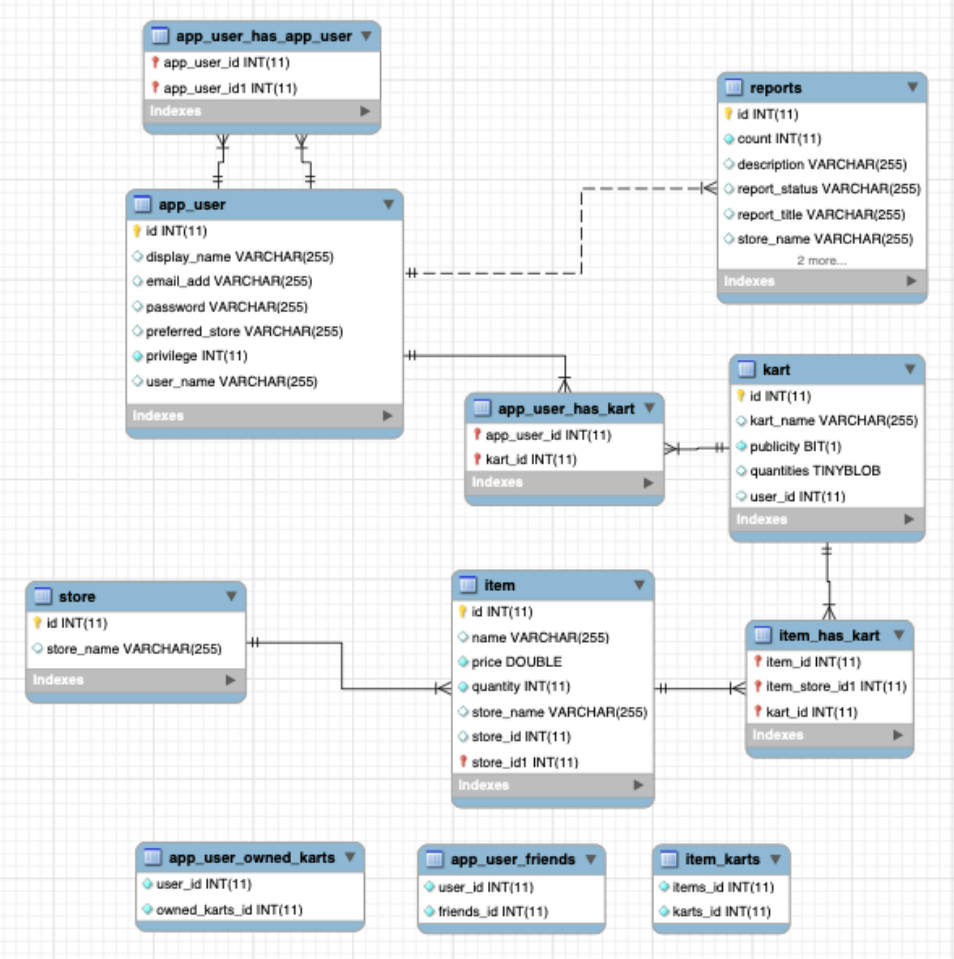
MySQL Connection

This is used to communicate with the database.

Database:

The database stores data that is necessary for the app to function, such as user information, stores, items. This data is stored in tables that have relationships between data points using MySQL.

Tables and Relationships



- App User has Many-to-Many relations with other users since there are friendship relations between each one of them when they follow each other.
- App User has 1-to-Many relation with the report as each user would be able to launch/have multiple reports under their account
- App User has Many-to-Many relation with Karts, where each user would be able to own multiple karts and due to the friendship feature multiple users can be assigned to a kart if they follow it
- Store has 1-to-Many relation with Items, since each store has multiple items listed
- Items have Many-to-Many relation with Karts, since each Item can be in multiple karts, and each kart can hold multiple items in it
- The 4 distinctive tables map the relations between the specific user to others, user to their owned karts, items to karts and store to items, respectively.