Design Document for Marketbuzz

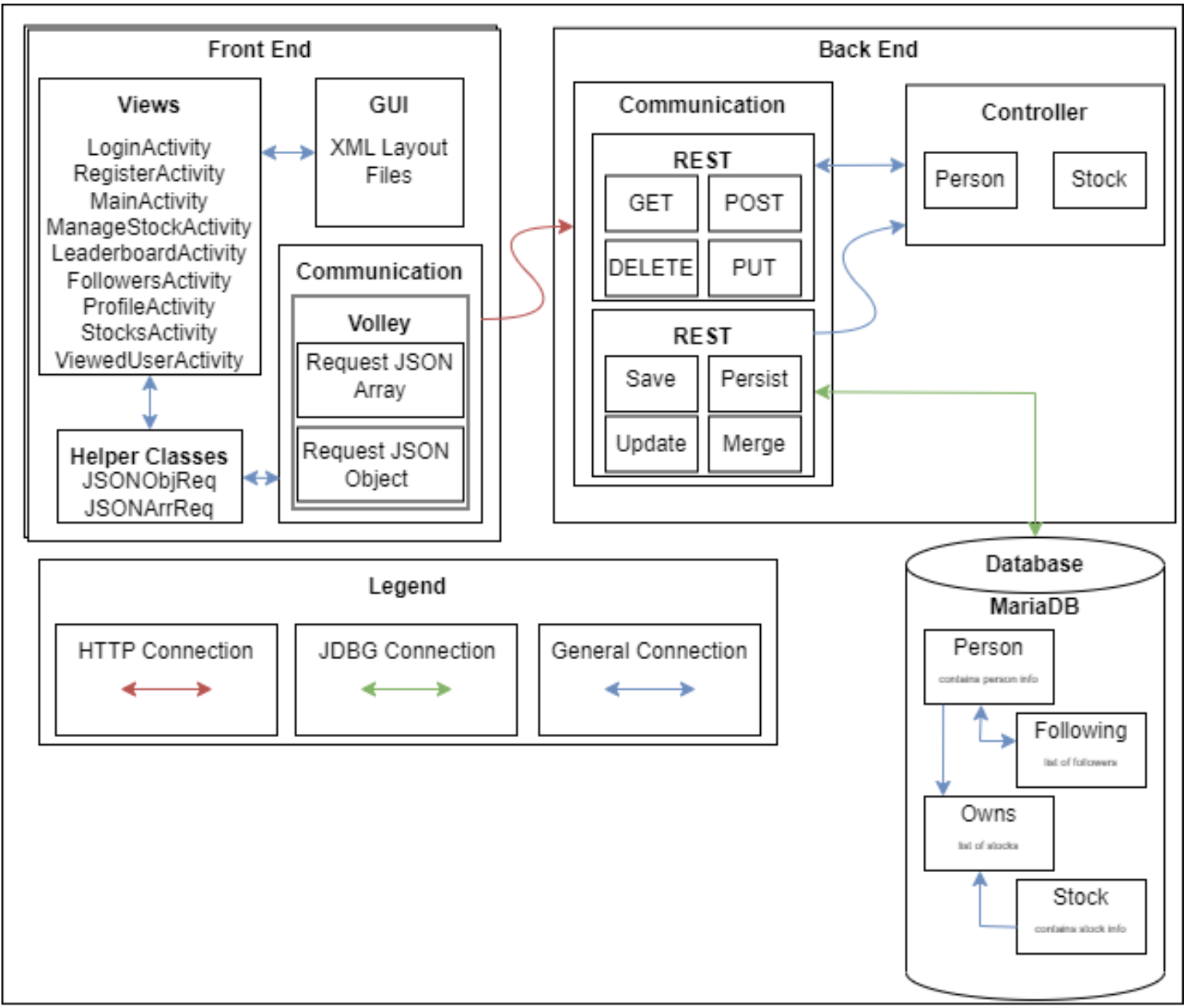
Group 1_cw_6

Ben Podjenski: 33.3% contribution

Joseph Jennings: 33.3% contribution

Sam Husted: 33.3% contribution

Block Diagram



Complex Design

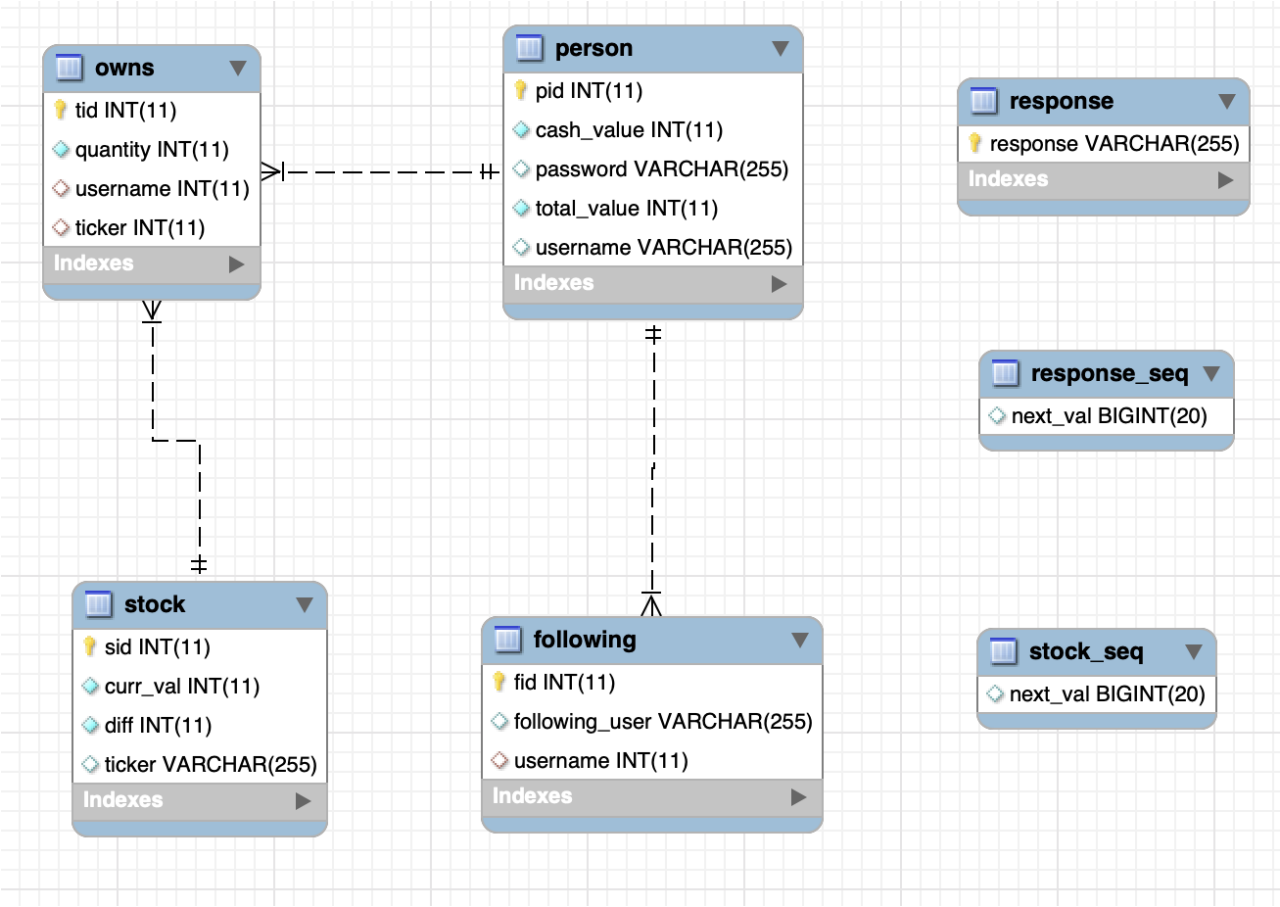
Frontend:

- RecyclerViews: There are many RecyclerViews throughout each activity. These allow us to scroll through a large defined list of objects that are pulled from the Backend. For example, on StocksActivity, there is a list of 20 stocks that is scrolled through. It would show the first 10 stocks and allow you to scroll further down to see the lower 10 stocks. These are gathered from a GET call from Backend to give us all the stocks. For each RecyclerView, a Model for one of the rows had to be made along with its corresponding adapter. For some of the RecyclerViews, you are allowed to interact with the model to go to another page. This happens because of the RecyclerView interface.
- CustomRequest: This class allows us to do Custom HTTP Requests. For normal requests, we can either send a JSONObjectRequest and return a JSONObject or send a JSONArrayRequest and return JSONArray. These custom requests allow us to send one type of JSON and it returns the other type. So, if I send a JSONObject, it would return a JSONArray, and vice versa for JSONArray. This was done because we had several requests we wanted to do that wouldn't return what we wanted it to.

Backend:

- Server: The server on the backend uses Spring to manage Requests and Calls. MariaDB is used to store objects and data relating to objects.
- Communication: The backend uses a stock controller and a person controller to communicate with the front end using GET and POST mappings to handle the following schemas.
- Schemas:
    - Person: Creates a Person object which has the following attributes:
        - pid - id value unique to each person entity.
        - cashValue - the value of money the person has that they are able to buy stocks with.
        - totalValue - the total valuations of a person's portfolio, including cash value.
        - username - the name the person is identified by to other users.
        - password
        - folloingList - a list of all people that a person is following.
        - ownsList - a list of all stocks that a person owns.
    - Following: Creates a Following object which has the following attributes:
        - followingUser - a string of the username of the person that is following.
        - username - a person entity of the person that is being followed.
    - Owns: Creates a Owns object which has the following attributes:
        - quantity - the number of shares that a person owns of said stock.
        - owner - a person entity of the owner of said stock.
        - stock - a stock entity of the stock that is owned.
        - stockTicker - a string of the stock ticker of the owned stock.
        - ownerUsername - a string of the username of the owner.
    - Stock: Creates a Stock object which has the following attributes:
        - sid - id value unique to each stock entity
        - ticker - a string of the ticker identifier for the stock
        - currVal - the current market value of one share of the stock.
        - diff - a int holding if the stock has increased, decreased or stayed the same.

Table Diagram



API Docks Link

http://coms-309-019.class.las.iastate.edu:8080/swagger-ui/index.html