**Design Document for Reline**

Group Reline

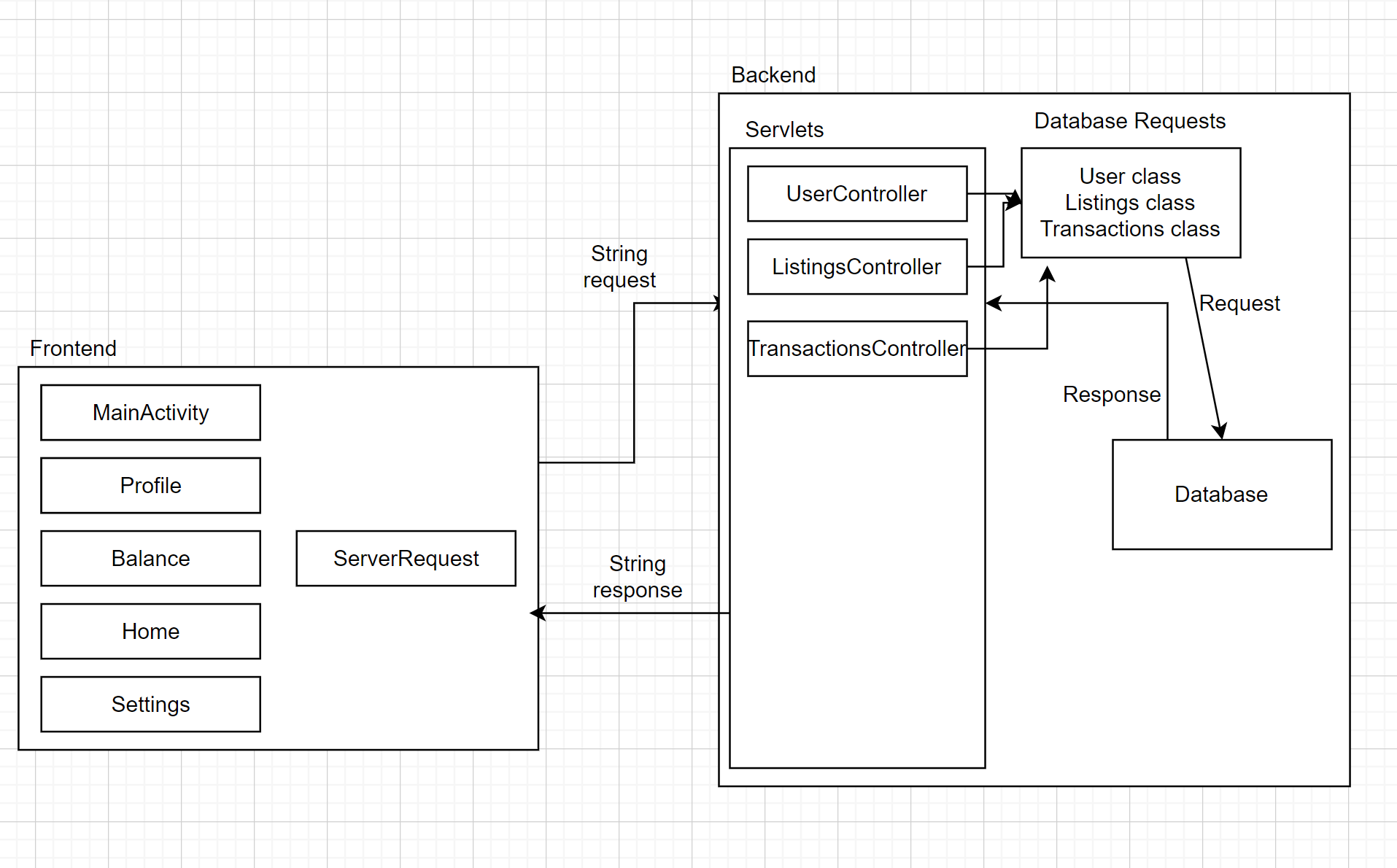
Riley Lawson: Riley % contribution

Alexander Haack: Alexander % contribution

Isaac Plambeck: Isaac % contribution

Wei Hern Lim: Noah % contribution

PUT THE BLOCK DIAGRAM PICTURE ON THIS PAGE! (Create the picture using pencil or drawIO)



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

Design Description

Model

Our app is designed to use database storage (stored on a custom remote server with gitlab pipelining) from a mysql server to store images, account information, and much more. We used lots of controller endpoints to communicate between the backend and the frontend. It allows the frontend to be able to pull and store proper information to and from the server. This includes both a listings and users table. With the use of spring boot, there are multiple simultaneous threads that allow the use of running background tasks without interrupting the user interface when communicating over a network (unless refreshing the whole page) while most of the project is done in Java. In the backend we have mock tests that insert false data to test out our design.

Controller

Once a user changes pages or refreshes the page, it calls a controller that handles the dataflow and calls depending on what calls the user decides to make. The controller file uses http methods to call the database to retrieve data to be sent to the UI classes and alter information in the database. There is also a controller class that encodes passwords so that the passwords are not kept in plain text. This is helpful in case a user reuses the password across different apps and our databases were to be compromised.

View

Our front end consists of only XML files and Java classes to control the functionality of those XML files. The views mostly use a simple combination of textboxes, buttons, image views to create a simple and clean UI. The activities make a call to a java file which then calls onto the server to retrieve data so that images, balances and other resources can be displayed accordingly. The calls are made whenever a refresh button or a link to another activity is clicked.

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

