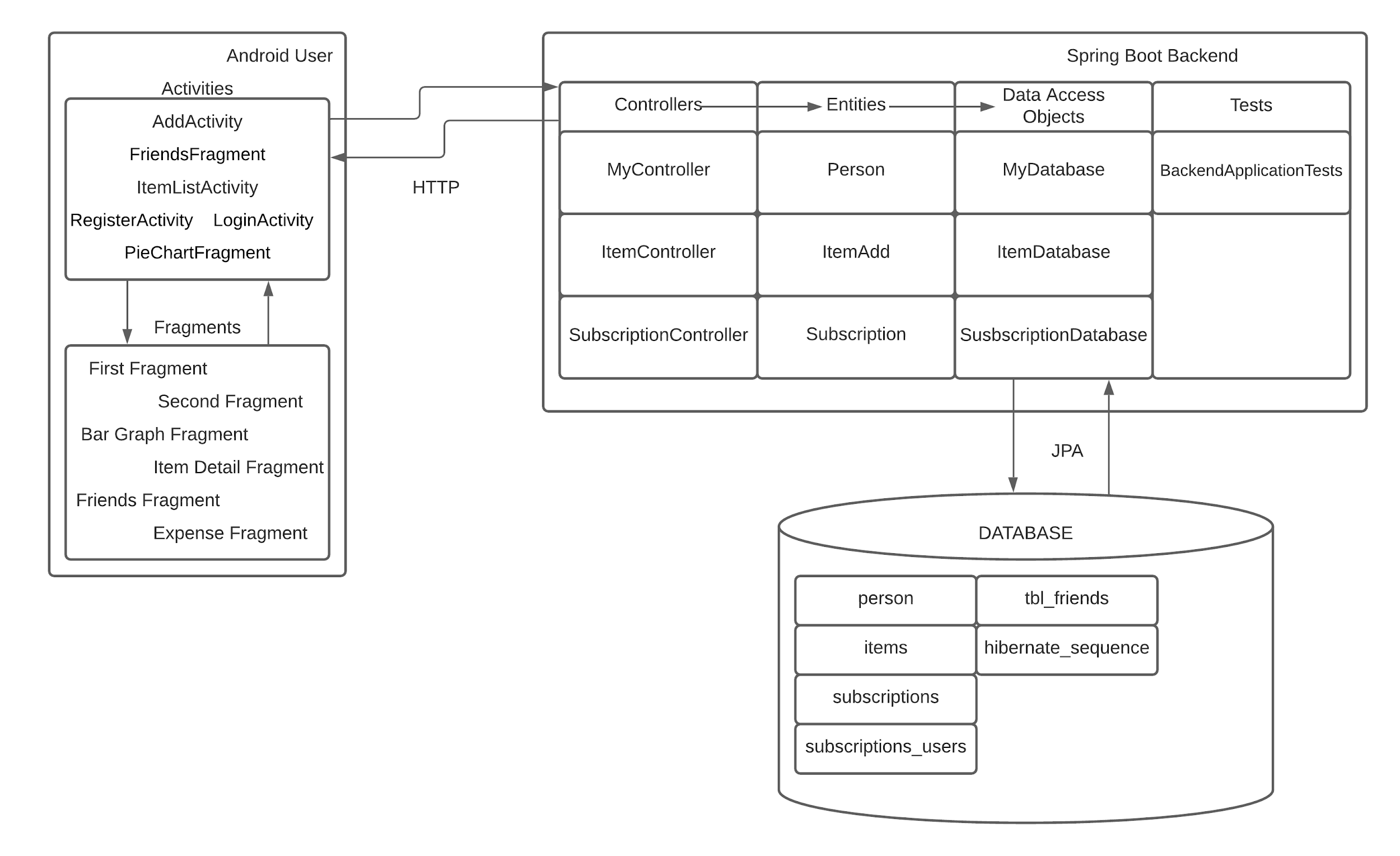
**Pocket Savings**

Block Diagram

UG\_02

Michael, Kateryna, Polina, Jacob

**Block Diagram**



**Design Description**

Code Set Up

Pocket Savings is created with android studio, java, and spring boot. The application has also been designed with user experience in mind. The app is extremely navigable and easy to use. The code has internal documentation to make the project more readable for other developers. Each component of the app has been sorted into specific packages for easy access and organization. Most methods are labeled strategically for easy understanding of their functionality.

User Experience Design:

Each page that the user can see on the android app has been simplified for user experience. Each page has three buttons the user can press at the bottom of the page. This will make the app extremely navigable and allow the user to have an extremely enjoyable experience. Every button is labeled based on its function, and they are easily accessed on the screens. The charts given to the user will allow them to visualize and analyze the data that they have put into the app. From here the user can make decisions about their fiscal health and spending patterns. Any information related to the user, such as the summary and friends lists, are directly displayed on their given screens.

Main Android Design:

The app connects to the server and is able to send and collect information about users. The information is sent and collected through json files. Variables are created to store information that the user inputs directly from the app and then are organized into json objects, which in turn are sent to the server. In order to display the objects, we store them in an array and reformat them to display the objects in the app.

Main Spring Boot Design:

The code is set up so that there is a controller and model for the users, subscriptions, and the items. The users model also has a many to many relation to itself. This creates the friends table. The users to items is a one to many relation and the users to subscriptions is a many to many relationship. This is what allows the user the upload and access their items and subscriptions. There is also a database file for each of the entities described above. This allows the server to create and maintain a database of user and user information.

**Tables and Fields**

