**Design Document for ISU Groups**

Group 3\_MC\_3

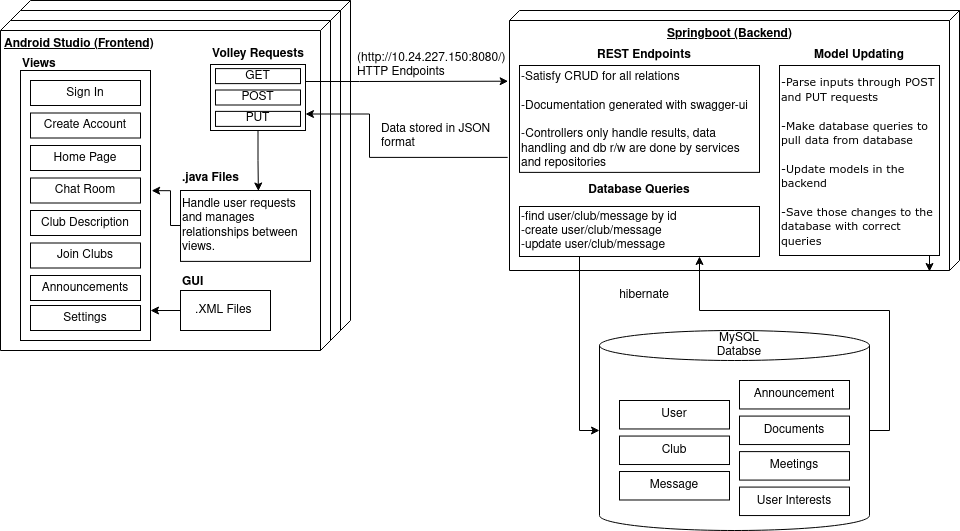
Andrew Adams

Ryan Derrick

Gavin Andres

Cody Prochaska

**Block Diagram:**

****

**Data Storage**

Our data is stored in MySQL, managed by JPA/Hibernate using Springboot.

All distinct entities are represented as Java objects, with foreign keys making up the relations between them. In our current design, Clubs and Users have been implemented, as well as a many to many relationship between them for enrollment. Each User object contains a hash table with all the clubs they are enrolled in, which are foreign keys from the Clubs table. This is more straightforward, takes less storage, and is faster to query than a separate entity relation table containing the same information. In addition, a Chat table has been implemented, but is still undergoing work to use foreign keys. Once finished, it will use a foreign studentId to represent the message sender, and a foreign clubId to represent the chat in which it was sent. While this design only accounts for one chat per club at the moment, this is easily fixed by creating a subchat instance variable for messages and clubs.

At the moment most calculations are handled by controller classes, but are being moved into service classes to improve modularity and testing feasibility.

**Interfacing and Manipulating Data**

All data manipulation at the moment is done through a REST api that satisfies all points of CRUD. Server endpoints exist to create all needed objects, modify any data that is allowed to change, and delete objects with controls in place to prevent wrongful deletions.

The actual manipulation of server classes utilized the CRUDRepository interface provided by Springboot, which automatically implements methods to manage data that are then used by our service classes. The underlying library used to access our MySQL server is Hibernate, which supports the autowiring of objects in Springboot.

**Frontend Design**

The frontend is made up of .java and .xml files in android studio. The xml files define the Graphical User Interface (GUI). The java files help handle the volley requests, which send a request to our backend endpoints over HTTP and receive data from the server in JSON form. They also define how each individual view interacts with each other.

**Current MySQL Table Diagram**

