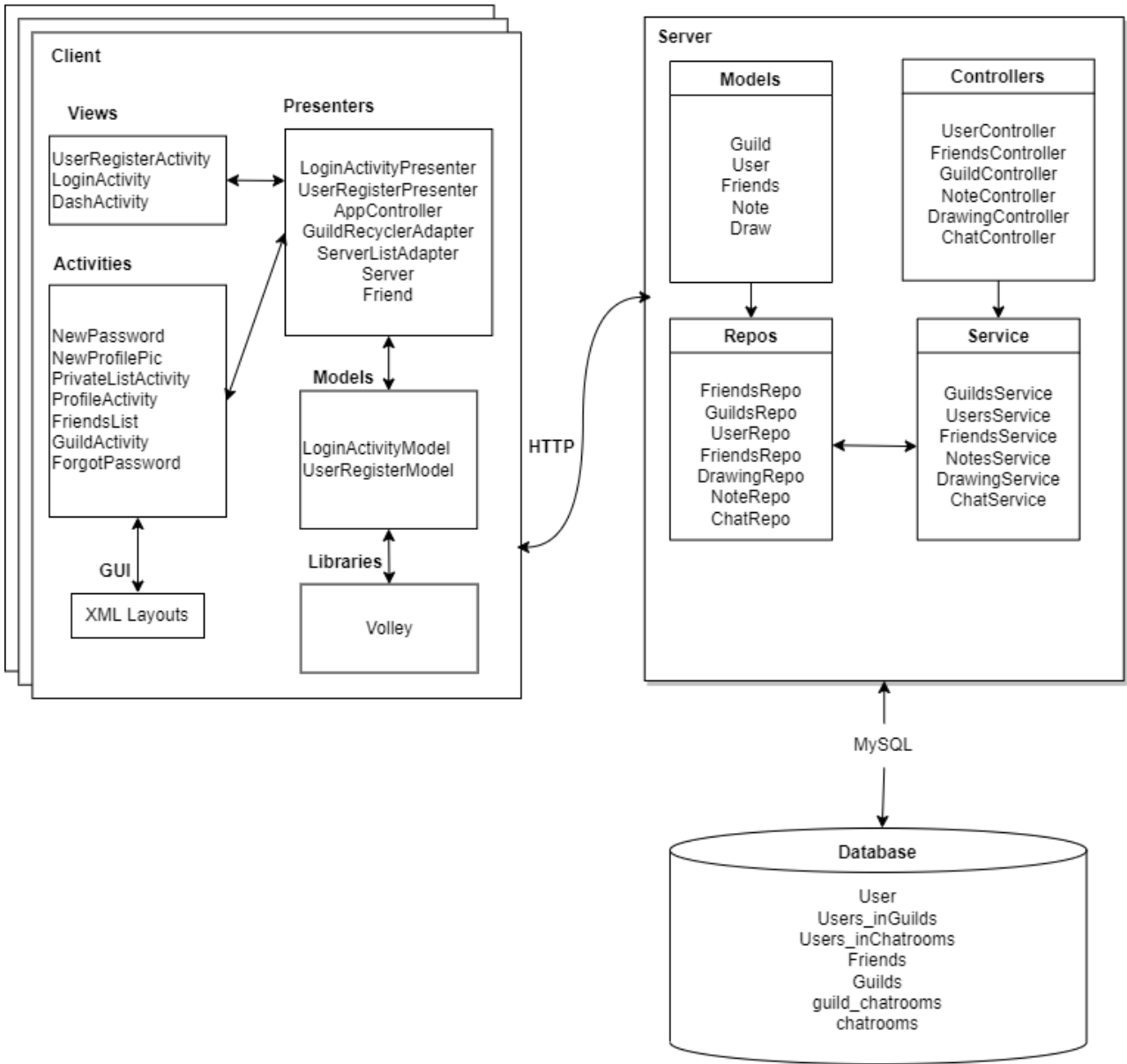

Design Document for Infinity

Group: 1_MC_2

Members: Aaron Case, James Root, Cale Rosenbalm, & Leo Stoyanov

Block Diagram

Project: Infinity
Team: 1_MC_2
Members: Aaron Case, James Root, Cale Rosenbalm, & Leo Stoyanov



Frontend

Views:

The view classes establish the xml files and allow the user to interact with that specific view. The views connect to the presenters, which connect to the models.

Activities:

These activities are primarily god classes, not organized in the MVP pattern. Given their status, they perform various actions. For example, the GuildActivity class sets up the basic information (including data received from other classes and objects in the GUI) for each guild once a guild is selected by the user and opens up the guild home page.

Models:

The models make all of the HTTP requests, using GET, POST, and PUT. For example, the UserRegisterModel class makes a get request in order to check if the user exists (i.e., if the email is already registered, then an account exists), and it makes a post request, sending the user's entered display name, email, and password, along with a generated username that combines the user's display name and password (i.e., John#23).

Backend

Springboot Controllers:

The controllers allow for the frontend to communicate to the backend. Depending on how the controller is implemented, the frontend can send data or request it.

Springboot Models:

The models are the “wrapper” of what exactly we are trying to implement. For instance one of our models is a user. Within the model, it requires certain information that will be needed for future implementation, or information that is required to distinguish between other users.

Springboot Repositories:

The repositories give the backend the ability to add to the server's database in which we can Get, Post, Put, and Delete data as we need.

API Documentation

