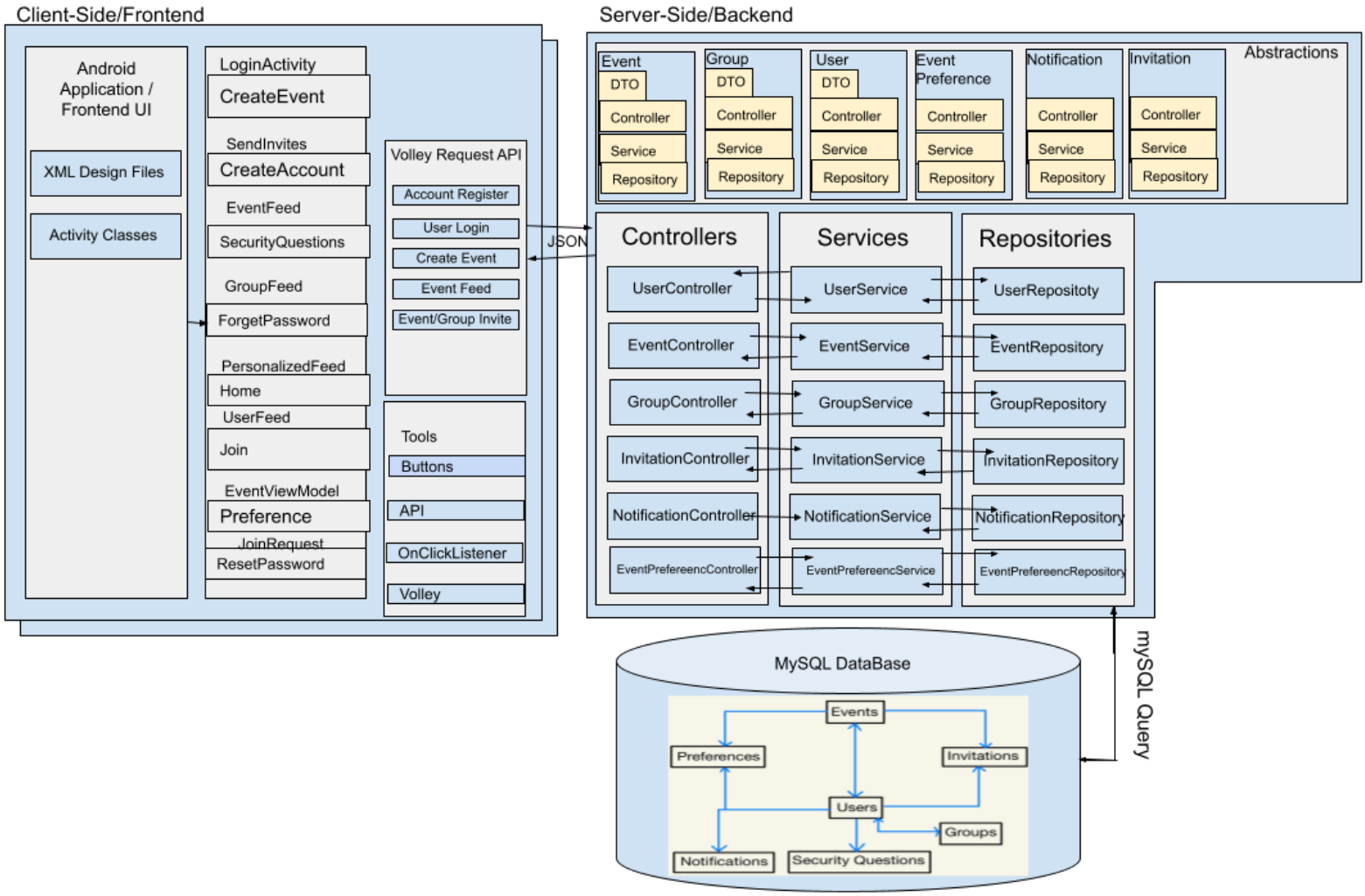
Block Diagram

Group 1_JK_6:
Chris Tague
Nikhil Kuricheti
Austin Krekula
Trevor Friedl

Event Planning Application

Application Block Diagram



Application Design Description

Android Application/Frontend: The front end portion of the app is focused around the xml files and the activity files. Each goes together to create the screen for what the client will be seeing. As the user goes into the app for the first time, the user will be taken through the create account screen where the user will enter the necessary information needed to make an account and then the user will set up security questions and preferences. Once that is done, they will be taken to the login. In the case where a user forgets their login the user will be taken into the forget password section to answer security questions and reset the login. After login the user will have the opportunity to create or join events and have access to a calendar to see if there are any conflicts between two or more events.

Tool: The tools that are used throughout the project are the buttons along with the API and the OnClickListener while connecting with the backend using Volley. The button allows the user to move from page to page while the OnClickListener makes it possible to integrate the button onto the program. The API allows the use of an interface onto the app. Currently the API is being used for the calendar. Finally, Volley allows the Frontend to communicate to the Backend through a series of requests, for example, GET, PUT, PUSH and more.

Volley: The volley portion is the main controller responsible for communication between the frontend and the backend. Different examples of requests are provided in the block diagram and all volley communication is made through JSON responses, either in the form of JSON object request or JSON array requests.

Server/Backend: The function of the back end of our application is to act as an interface between the client and the MySQL database, as well as to perform certain logical operations on data. We are using three primary layers to control data exchange between the client and the database and maintain modularity. The first layer is the control layer. The controllers interface directly with the clients by defining endpoints that they can use to access or modify data. The controller layer calls functions in the service layer. The service layer's job is to manipulate data and feed it back to the controller. It does this by interacting with the repository layer. The server layer will make a call to the repository, perform some simple data manipulation, then send that data on to the controller, or send a status to the controller. Alternatively the service layer can be used to make new entries into the repositories. The final layer is the repository layer. This layer directly interacts with the database. The repository is simply a modularized way for the service layer to access certain parts of the database. The repository automatically preforms mySQL calls to retrieve and modify the database, then it passes that data along to the service layer.

Database: The database holds many relations between different objects. These include users, events, groups, notifications, invitations, security questions. Some notable relations are user to events, users pending in events, user admins, groups admins, event admins. The relations among the data are all managed by spring, and all the relation tables are created by spring.

List and Tables - DataBase Schema