

---

## Design Document for **Schedulity**

---

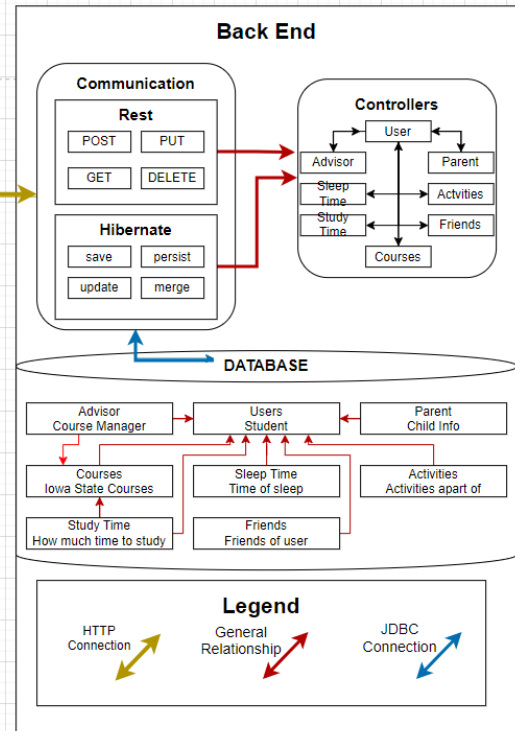
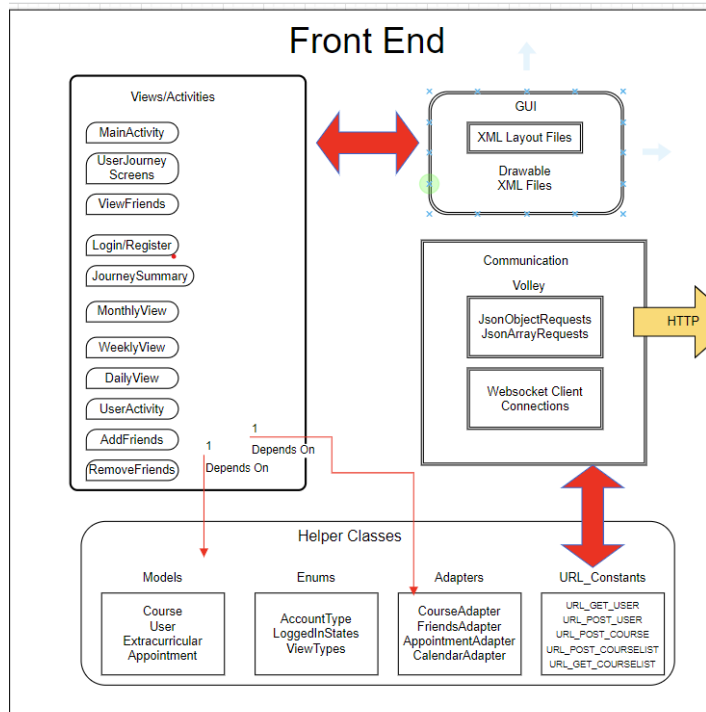
Group **1\_AN\_2**

Wesley Jansen: 25% contribution

Elizabeth Schmitt: 25% contribution

Bobby Stickler: 25% contribution

Bean Towle: 25% contribution



### **Backend Communication:**

Different mappings are used to work alongside CRUDL to Create Receive Update Delete List different items in the database.

**Controllers:** Uses URLs so then the frontend can access the database and gives functionality

**User:** User is the user type that this app is made for and has many relationships: One-to-One relationships with parent and sleep time. One-to-Many with activities, Many-to-One with advisor and study time and Many-to-Many with Courses and friends which is a self-relation as well. The User has their calendar and everything they are doing and can add and edit whatever they want besides the courses they are in.

**Parent:** The Parent type can add a user to be their child and then view their child's schedule. The parent cannot make any edits to the database or add any events.

**Advisor:** The advisor type can have many students which the advisor can edit those students class schedule. The advisor is the only one with the ability to drop a class for a student.

**Sleep Time:** A user needs to sleep at some point during the week. This sleep time adds a start and end time to the users sleep schedule which they cannot have activities during.

**Courses:** The courses that Iowa State has to offer. These have a Many-to-Many relationship with users. One user can have many courses and one course can have many users.

**Study Time:** Time for each student to study for each of their courses. This is indicated by the user and how important they find each course they are taking to be.

**Friends:** Users can view friends schedules and message them. This is a Many-to-Many relationship.

**Activities:** A user can add an activity to their schedule as long as it does not overlap with a course they are taking or their sleep time.

### **Frontend Design and Complex Aspects**

**GUI:** Our Android application consists of 25 different activities that the student user can click through. 13 of which are dedicated to prompting the user input different details about their weekly schedule in terms of classes, extracurricular activities, etc. Each activity has its respective xml layout file dictating the overall design of each screen.

**Helpers:** Our frontend code contains several helper classes and methods. Our models are some of the most used and most important classes in our design as they contain and save information related to the current user of the application as well as the user's list of courses, appointments, extracurricular activities, and friends. Much of the code within each activity is dependent upon these model classes. Many of the scrollable lists displayed in our activities need Adapter classes in order for our android code to know what to display to the user. These adapters are used to display lists of courses, friends, and extracurricular activities. We directly call the Adapter classes when we create recycler views (scrollable list views). In addition to the Models and Adapters, our android design also has a URL class that holds our URL string constants. These are used to make Volley HTTP requests to either send or receive information to the backend.

**Communication:** Our android application communicates primarily with the backend spring boot application through http requests in JSON format using the android volley library. These requests are used to send all information back and forth between users, courses, study times, etc. Each request can be made as a POST or GET request depending on which direction the information should be sent. After each request, the backend will also send a response that is to be processed by

the frontend developers. In addition to Volley requests, the android application can also communicate with the backend using websockets. Websockets are used to support a chat feature in which the user can message with whoever they are friends with.

