Design Document for Course View

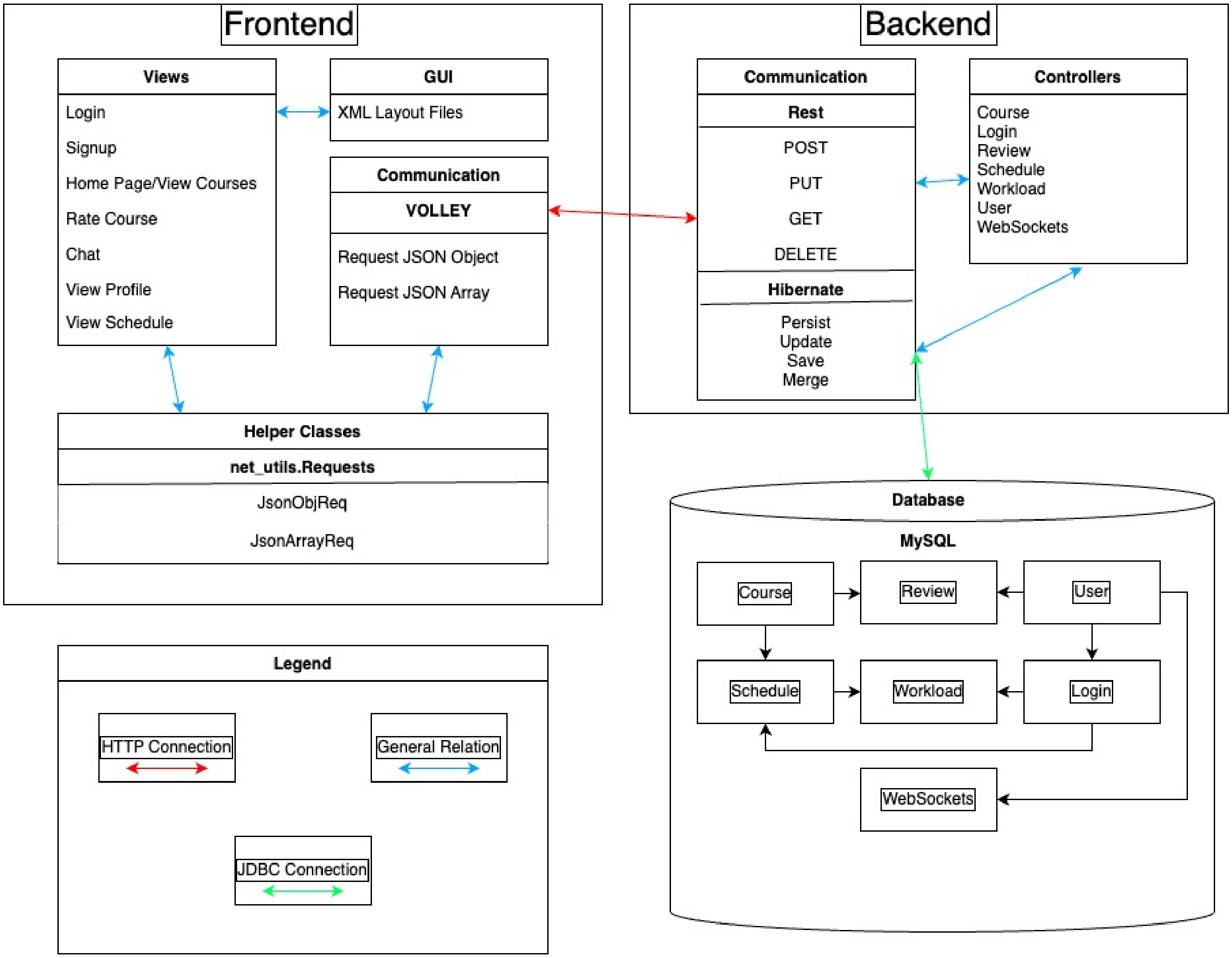
Group LG_113

Devin Alamsya: 25 % contribution

Kalem Schrock: 25% contribution

Manjul Balayar: 25% contribution

Benjamin Muslic: 25% contribution


Frontend
Views
Login
Signup
Home Page/View Courses
Rate Course
Chat
View Profile
View Schedule
GUI
XML Layout Files
Communication
VOLLEY
Request JSON Object
Request JSON Array
Helper Classes
net_utils.Requests
JsonObjReq
JsonArrayReq
Backend
Communication
Rest
POST
PUT
GET
DELETE
Hibernate
Persist
Update
Save
Merge
Controllers
Course
Login
Review
Schedule
Workload
User
WebSockets
Database
MySQL
Course
Review
User
Schedule
Workload
Login
WebSockets
Legend
HTTP Connection
General Relation
JDBC Connection

Frontend

SignUp (User)

- A page is generated with the elements listed below:
    - EditText: Username
    - EditText: First Name
    - EditText: Last Name
    - EditText: Email
    - EditText: Password
    - Spinner: User Role
    - Button: Sign Up
- When the sign up button is clicked, the values in the username, first name, last name, email, password, and user role fields will be sent as a POST request to the server.

Login (User)

- A page is generated with the elements listed below:
    - EditText: Username
    - EditText: Password
    - Button: Login
- When the login button is clicked, it checks the values of username and password and compares it to the database. Sends user to homescreen if the user is valid.

LoadCourses (Course)

- Load courses generates a page with the elements listed below
    - Spinner: Courses
    - TextView: Course Description
    - TextView: Selected Course
- When the home page is loaded, a GET request is completed from the server to populate the page

Backend

Communication

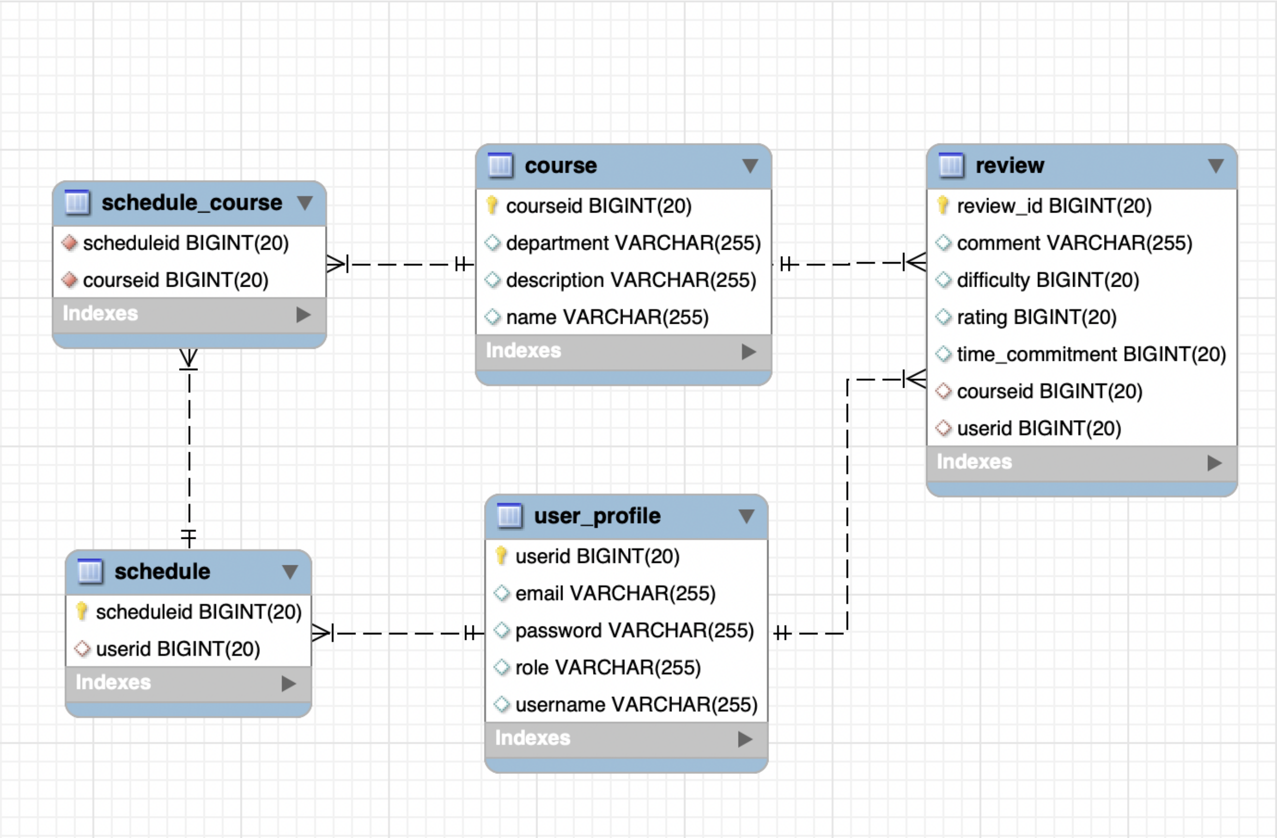
Our backend uses various different URL mappings to update the information in our database:

1. Post: Adding information or an item to the database
2. Get: Request information and sometimes uses an identifier to request something specific.
3. Put: Updating something specific item in the database.
4. Delete: Using an identifier to delete a specific item from the database

Controllers

Our controllers make use of these mappings to communicate with the frontend and the database.

1. Course: Contains the mappings from above and has a one-to-many relationship with Review and a many-to-many relationship with Schedule.
2. Login: Contains one post and one get mapping that let's you post a user to the list of users while also helping you find a specific user based on their username to login.
3. Review: Contains the mappings from above and has a many-to-one relationship with both User and Course.
4. Schedule: Contains the mappings from above and has a many-to-many relationship with Course and a one-to-one relationship with User.
5. User: Contains the mappings from above and has a one-to-many relationship with Review and a one-to-one relationship with Schedule.
6. Workload: Contains two get mappings with either user ID or course ID to get the workload of a user's schedule (which is filled with different courses).



schedule_course

- scheduleid BIGINT(20)
- courseid BIGINT(20)

Indexes

course

- courseid BIGINT(20)
- department VARCHAR(255)
- description VARCHAR(255)
- name VARCHAR(255)

Indexes

review

- review_id BIGINT(20)
- comment VARCHAR(255)
- difficulty BIGINT(20)
- rating BIGINT(20)
- time_commitment BIGINT(20)
- courseid BIGINT(20)
- userid BIGINT(20)

Indexes

schedule

- scheduleid BIGINT(20)
- userid BIGINT(20)

Indexes

user_profile

- userid BIGINT(20)
- email VARCHAR(255)
- password VARCHAR(255)
- role VARCHAR(255)
- username VARCHAR(255)

Indexes