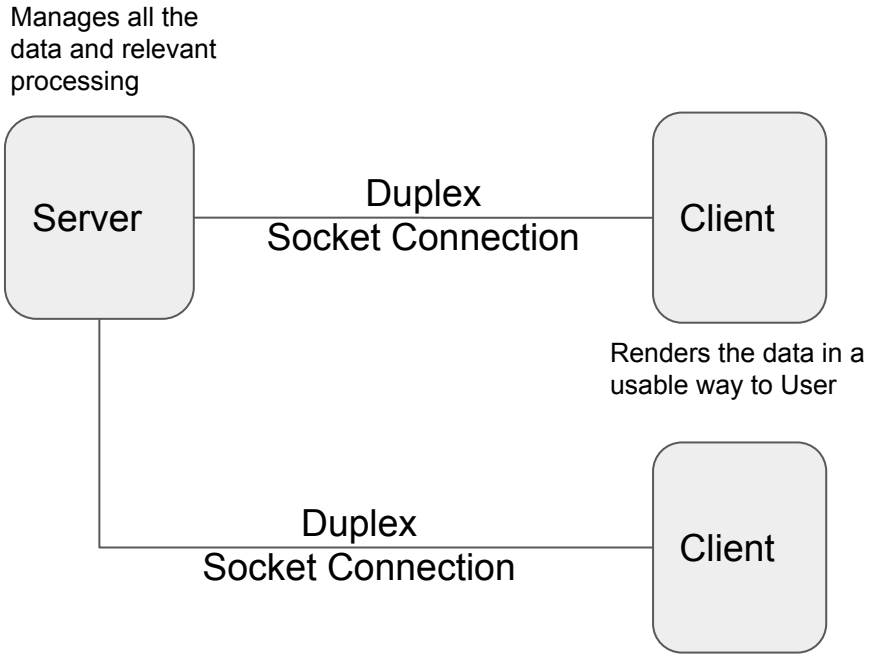
Project 1 - AP (Classroom Booking System)

Implementation Details

Design Patterns Used:

- Singleton for Server and Timetable
- Flyweight for Database of Users, Courses and Room Bookings
- Prototype for Communication protocol over Sockets
- Facade for Connection handling with clients



Problems faced during the project:

- Database Scheme - We decided to use HashMaps and Array Lists and serialise them into a file for the database for convenience of integration with Java Project.
- Parsing CSV and creating a database - We had to build a CSV file manually since the excel sheet with timetable was very irregular, This CSV needed to be converted and stored in the scheme designed above, we used the open source library OpenCSV in order to do so.
- Managing Multiple Clients - In order for our application to serve to multiple clients, we segregated it into two parts, Client and Server. Server taking up the responsibility of secured and safe data access.
- Communication between Server and Client - To solve this, we devised our own Protocol for any exchange happening between Client and Server, using our own RequestObj Class.
- Data Concurrent access issue - We used locks and appropriate thread safe techniques on Server side in order to ensure consistent usage over several instances.

Individual Contributions

Member 1
Viresh Gupta
Roll No - 2016118

- Class implementations
- Communication over Socket
- Thread Safety

Member 2
Baani Leen Kaur Jolly
Roll No - 2016234

- GUI designing
- UML Designing
- Class structuring

Along with lots of tasks in between that were done in team spirit :D
JavaDoc available at https://virresh.github.io/Schedulr/
Code hosted on GitHub at https://github.com/virresh/Schedulr (Private Repository Currently)

Extra Features

- Multithreading
- Socket Programming
- Ability for Admin to edit any request that it receives from students to change the venues (and only venues).
- Ability for Faculty to register for courses