## Techniques, Architecture, and Data Structures

The Distributed Room Reservation System (DRRS) is implemented on five servers: DVL, KKL, WST, Central Repository, and Client.

### Client

Here are the most important files: Client.java, StudentClient.java, and AdminClient.java.

StudentClient.java and AdminClient.java are interfaces implemented on DVL, KKL, and WST (see AdminServer.java and StudentServer.java in DVL, KKL, and WST). Client.java needs to be run, and during the run time, the user, through his inputs, will remotely call the appropriate methods through StudentClient.java and AdminClient.java.

There is also a folder named "logs". When Client.java is running, it will create a log file with all the requests and responses for each user. Client.java will delete every log after the run is complete such that the number of logs matches the number of users currently running the program.

### DVL, KKL, and WST

Each of these servers shares the same set of files. The most important ones are Server.java, RoomRecords.java, StudentClient.java, AdminClient.java, StudentServer.java, and AdminServer.java. There is also a file named "log.txt".

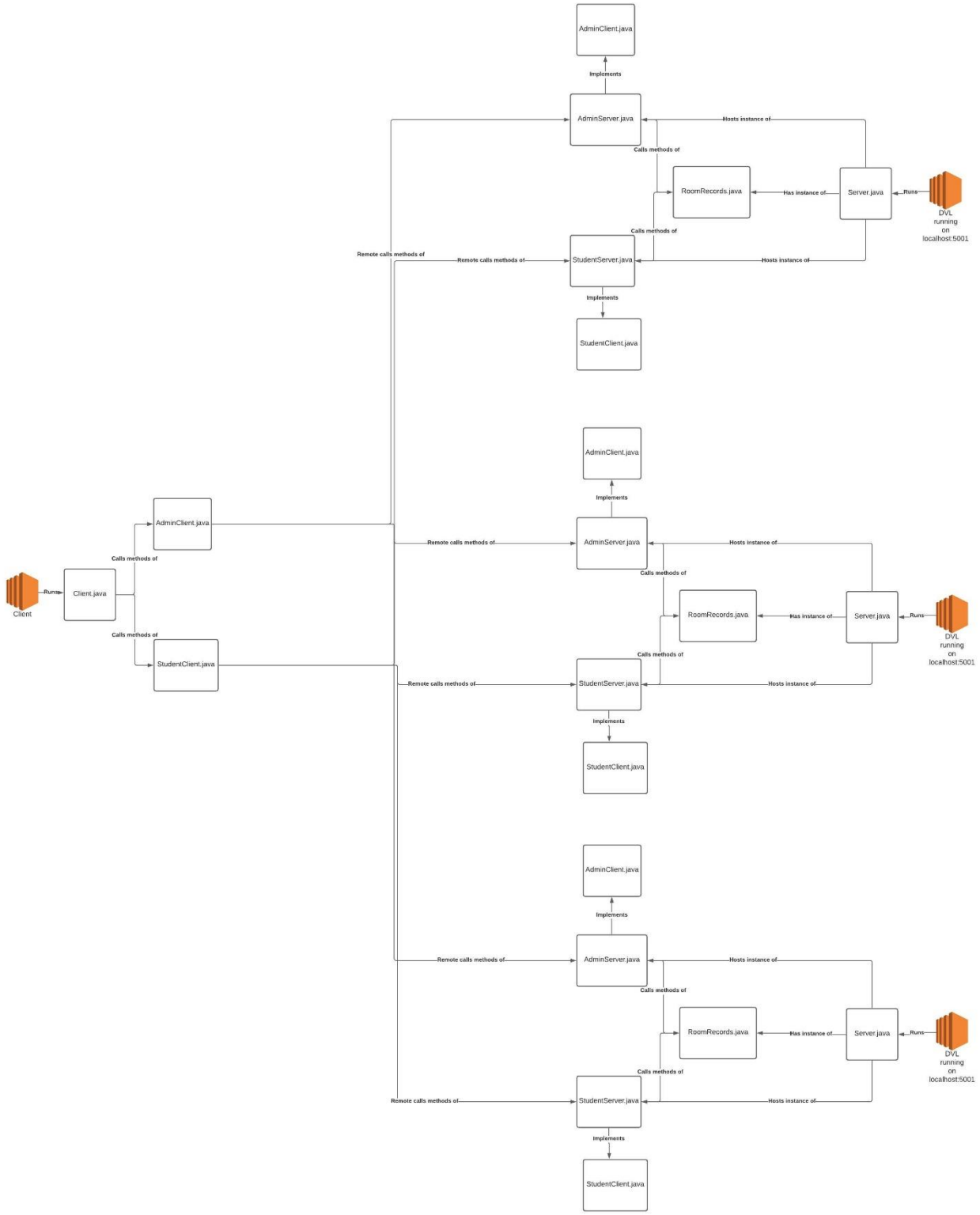
Server.java needs to be run. During the run time, it will allow StudentServer.java and AdminServer.java (which implement StudentClient.java and AdminClient.java) to interact with RoomRecords.java (instantiated as a static attribute of Server Class).

RoomRecords.java has an attribute "date2RoomNumber2TimeSlots2StudentID," a HashMaps nested with more HashMaps where its name describes its chain of key-value pairs. RoomRecords provides the synchronized methods createRoom, deleteRoom, bookRoom, and cancelRoom. It provides all the functionality asked in the assignment instructions. The key StudentID maps to either null if unreserved or the id of the student who reserved it.

DVL runs on localhost:5001, KKL on localhost:5002, and WST on localhost:5003.

There is also a file named "log.txt" that logs each request received and responses sent. It clears the file on each run of Server.java.

# Systems Design Diagram



## Testing

| Testing description | Result |
|---|---|
| Can you run multiple clients at the same time without undergoing any resource conflict when invoking RoomRecords methods? | Yes. |
| Are the logs in the Client and the Server correct? | Yes. |

## The most challenging part of this assignment

Implementing the nested hashmaps was the most challenging. Using a relational database would have been easier, efficient, and concurrency would automatically be taken care of.