To test the Project

1. We started with socket programming and to test that our server and client are working fine we just send a message from client to server and got a reply back from the server.

   We were converting the sent message to the upper case in the server and sending the upper cased message back to the client.

2. Then we started testing our server-client model from the different systems but on the same network. We had to un-block the firewall and give the server IP address in the client system.

3. Third step was testing the multi-thread program which we created in order to communicate between 1 server and multiple clients. We tested this using the same method mentioned in step 1.

4. Next step was creating a datacenter, which consisted of a server and database installed on the local host. This was tested using few demo tables and we integrated data base with the server program using jdbc.

5. The next step was hosting the server on the web. For that we used Tomcat. And the results of the query was published using the json format, which is easy to transmit using http protocol.

6. Once our data center model and the clients were working fine we moved to the android application step. The android application was made using android studio. Several activities were created and tested using the software itself. Finally the android studio also helped us in testing our final application.

7. The final step was integrating all the modules together and testing the final functionality of our system.

For this we are having multiple clients connected to the server. And each server represents 1 company. If the load in the each server is going above the specified limit then the server is getting overloaded and it will switch. In our project we have specified the maximum numbers of clients connected to 3. Further, all these can be monitored in the android app which is there with the manager. He can see how many clients are connected to the each server (company).

Further we planned to apply some firewall to the servers. If an unauthorized person logs in from the android app or the clients are unauthorized then the security mechanism will detect those clients and block it. We tried to implement this but because of shortage of time we were unable to do this.