**Distributed Room Reservation System (DRRS) using CORBA**

**Source Code Design :**

DRRS application consists of two clients and three servers with an interface that contains the methods which will be implemented in server programs and can be used by the client programs

Two client programs are developed for Client and Admin who can perform operations on rooms available in three campus such as Dorval-Campus (DVL), Kirkland-Campus (KKL) and Westmount-Campus (WST) for your implementation. Each campus has its own server and database. In this application, Each server has its own hashmap to store, retrieve and remove data.

Below files are included with this documentation

* AdminClient.java
* StudentClient.java
* KklBookingServer.java
* DvlBookingServer.java
* WstBookingServer.java
* RoomBooking.idl
* DvlBookingServerImpl.java
* KklBookingServerImpl.java
* WstBookingServerImpl.java

**AdminClient :**

AdminClient program allows administrators to create and delete rooms in each campus. Each campus has its own set of administrative users only who can access particular server’s database.

Any user of the system can perform the below basic operations:

* Create a room with its time slots.
* Delete a room with its time slots

**StudentClient:**

StudentClient program allows the users\students to get the list of available rooms from all three campus, book a room in a campus or other campus, cancel a booking that was made in the same campus or other campus that is achieved by calling below methods available in server implementation.

Any user of the system can perform the below operation

* Get the list of available rooms from all servers
* Book a room
* Cancel booking
* Change the reservation

**KklBookingServer,DvlBookingServer,WstBookingServer :**

All three servers have methods for all the above listed operations which is being accessed though remote object.

**RoomBooking.idl:**

This interface specifies all the methods which will be implemented and accessed by all client and server programs

* createRoom
* deleteRoom
* bookRoom
* cancelBooking
* getAvailableTimeSlot
* changeReservation

**Test Cases :**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Test Case Description** | **Expected** |
| 1 | Create a room with multiple time slots | Pass |
| 2 | Add few more time slots to an existing room | Pass |
| 3 | Delete a time slot | Pass |
| 4 | Book a room in same campus | Pass |
| 5 | Book a room in other campus | Pass |
| 6 | Change reservation that was made in same campus | Pass |
| 7 | Change reservation that was made in another campus | Pass |
| 8 | Cancel a booked room in same campus | Pass |
| 9 | Cancel a booking in other campus | Pass |
| 10 | Get available timeslots across the servers | Pass |
| 11 | Book 4 rooms in a week in same campus | Fail |
| 12 | Book 4 rooms in a week in different campus | Fail |
| 13 | Multiple clients access server operations at the same time | Pass |