**COMP6231 Assignment3**

**Name: Kaichen Zhang ID: 40000160**

**Techniques:**

In this assignment:

The work flow is:

cd documents/workspace/Zhang/bin

/bin

wsgen -verbose -cp . servers.MTLServer -wsdl

wsgen -verbose -cp . servers.WDCServer -wsdl

wsgen -verbose -cp . servers.NDLServer -wsdl

cd ..

/

wsimport -keep -d src/ -p client.MTL http://localhost:1050/mtlservice?wsdl

wsimport -keep -d src/ -p client.WDC http://localhost:1050/wstservice?wsdl

wsimport -keep -d src/ -p client.NDL http://localhost:1050/delservice?wsdl

First generate end point files using wsgen commend, then import wsdl files using wsimport command.

I used **UDP** to implement the communication between servers.

I used **HashMap** to store the passenger records and flight record.

I used **multithreading** technique to implement that multiple clients can act simultaneously.

I used **synchronization** technique to keep the integrity of data while modifying it, so the server can maximize the concurrency.

**Design architecture:**

clients package:

Client class defines the human machine interaction, by simulate both passenger client and manager client.

MultiThreadTest class uses multithreading to test multi clients interaction. Which can check the performance of synchronization.

Records package:

Defining the flight record data structure, passenger record data structure. Both are using hash map to store.

servers package:

Communications are made through ORB.

Containing the FlightServer class, providing remote method invocation by extends ServerInterfacePOA, implemented by MTL,WDC and NDL servers.

FlightServer class overrides the bookFlight method, but put a new passenger into the hash map using his first character of last name as key, and passenger records as value.

FlightServer class overrides the getBookedFlightCount method by sending UDP messages to other servers, and receive UDP counter reply messages from others.

FlightServer class overrides editRecord method by editing expected flight record field, giving record ID, filed name and new value.

FlightServer class overrides transferReservation method, by sending passenger record using UPD message from original city to other city. And making reservation at the other city, canceling passenger record at the original server automatically.

**Scenarios:**

Firstly we should run the server of three different cities.

**Human machine interaction scenario:**

To simulate the passenger, we run the client class and choose to perform as passenger, and we can do the operations:

Entering valid information and booking a ticket.

To simulate the manager, we run the client class and choose to perform as manager, and we can do the operations like:

Entering valid manager ID.

1. Manage flight information:

1) return all available flights at the current server 2) edit flight information 3) back

2. Get booked flights counts by sending UDP

3. Transfer reservation by entering existing passenger ID and other city name.

4. Exit

**Important part/ Difficulty:**

First I tried many times to generate the end points files in my windows environment, but it always failed. After I checked many documentation I changed to use Linux environment to generate and it passed immediately. I still don’t know why is that.

And the import step should be done after publish the servers, which caused me confusing for a long time.