

Appendices R

System Testing

As explained in the report, due to the graphical user interface (GUI) not being implemented, I was unable to carry out system testing as I had planned, and was unable to test all of the functionality implemented such as the Observer pattern due to the GUI not being implemented, which is the observer within the observer pattern.

To carry out the system testing due to not having implemented the GUI I decided to create a test class for the Client, so firstly I had to run the Server class, to allow the Client to be able to connect to it and the output of the Server class is below:

run:

Started Registry

ServerTEST

Daily Tasks

Server started

Daily Tasks

Daily Tasks

Daily Tasks

Daily Tasks

Daily Tasks

Daily Tasks

BUILD STOPPED (total time: 7 minutes 14 seconds)

As you can see from the above Server output, the DailyTasks are being output, this is because for the TaskGenerator class within my Server class (Task Scheduling functionality), I have amended the time in which tasks are generated (as they are usually generated every 24 hours, which is $1000\text{milliseconds} * 60 * 60 * 24$, however because I have amended it to $1000\text{milliseconds} * 60$), it is now running every 60 seconds, as you can see from the BUILD TIME of 7 minutes 14 seconds, it has produced 7 `System.out.println("Daily Tasks")`. This proves that although I have not been able to show that the methods to create Rent and Lease Transactions every 24 hours, or create Salary Transactions and generate the monthly report every month (on the last day of the month), this proves that the task generator class is invoking the methods within set within the Task Generator class.

To test the client I then had to run the Client Test class, where I invoked a number of create methods to create elements within the database, and create an address and person, I also updated an element, tried to delete that updated element (which did not delete because it had been amended, and . The first time I run the client test, I got the below output:

```
*****Running Client Test*****
```

Policy found

file:/C:/Users/Dwayne/Documents/University%20Work/Masters/MSc%20Dissertation/project/MSc%20Properties%20Client/build/classes/RMISecurity.policy

Creating security policy

Environment : ServerTEST

Trying host : 127.0.0.1

Server found!

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

true

This shows that the client is able to invoke methods on the Server remotely and create objects which is then stored remotely on the server. I then decided to amend the client class to create an application, and also invoke the methods to try and create the elements that was previously created, which should return a false, because they are already created in the system. The output for the second client test is below:

run:

*****Running Client Test*****

Policy found

file:/C:/Users/Dwayne/Documents/University%20Work/Masters/MSc%20Dissertation/project/MSc%20Properties%20Client/build/classes/RMISecurity.policy

Creating security policy

Environment : ServerTEST

Trying host : 127.0.0.1

Server found!

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

false

true

true

true

true

true

true

false

As you can see from the output, the Elements that was already created (excluding the element I deleted from the previous run of ClientTest) it has returned false, as it has not created them elements, but has still updated the element, deleted the element again (because it was created again), and created the address and person, however it has not created the Application due to a bug in the server class which I have been unable to locate and fix due to time constraints.