

Appendices Q

Integration Testing

As explained during the report, due to time constraints I was unable to carry out integration testing at the end of each iteration so I decided to use the Big Bang Integration testing version, and I decided to use the Database class, as this class interacts with all of the Objects within my class, when creating, updating and deleting records from the MySQL database.

Unfortunately, though, due to time constraints, I was not able to test all of the database class methods, but have tested a large portion of the create, read, update and delete methods and is mainly the search methods that I have been unable to test.

For this integration test, as explained in the report, I used white box testing, and decided to write a test class, which creates a Database element, and allows me to invoke methods on this database object to test that the methods work. However due to having an external MySQL database which is updated using the database Create, Read, Update and Delete (CRUD) functionality, I have also provided some print screens of my database contents (due to the large number of classes I was unable to provide print screens of all of the updates.

Below is an output from the Database Test class:

run:

Running database test

Job Role Code: MNGR

Modifications: 3

Job Role Notes: 1

Job Role Requirements: 1

Job Role Benefits: 1

System Job Roles: 1

Address Ref: 1

Address: 5 Brook Crescent, Edmonton, London, England, N9 0DJ

Modifications: 1

System Addresses: 3

Person Ref: 1

Modifications: 1

Person Notes: 1

Person Contacts: 1

Person Documents: 1

Person Addresses: 1

System People: 3

Document previous versions: 0

Office Code: EDM

Modifications: 1

Office Contacts: 1

Office Address: 5 Brook Crescent, Edmonton, London, England, N9 0DJ

System Offices: 1

Landlord Ref: 1

Modifications: 1

Landlord Leases: 1(no leases added yet)

System Landlords: 2

Modifications: 3

Employee Contracts: 1(no contracts added yet)

System Employees: 1

Property Ref: 1

Modifications: 6

Property Rent: 300.0

Property Charges: 99.99

Property Landlords: 1

Property Documents: 1

PropertyElements: 3

System Properties: 3

System Property Elements: 3

Property Element 1: Property Element Ref: 1

stringValue: 3 BED - AMENDED

doubleValue: null

startDate: 2016-01-11

endDate: null

Charge: false

Property Element 2: Property Element Ref: 2

stringValue: null

doubleValue: 99.99

startDate: 2016-01-11

endDate: null

Charge: true

System Property Element Values: 3

Application Ref: 1

Application Correspondence Name: Dwayne Leroy Edwards

Modifications: 0

Application Addresses: 1

Application Household: 1

Application Properties Interested In: 0

System Applications: 1

Involved Party Ref: 2

Modifications: 1

Involved Party End Reason: null

System Involved Parties: 3

Tenancy Ref: 1

Modifications: 0

Tenancy App Ref: 1

Tenancy Charges: 99.99

Tenancy Rent: 300.0

Tenancy Prop Ref: 1

System Tenancies: 1

Rent Acc Ref: 1

Modifications: 0

Rent Acc Rent: 399.99

Tenancy Ref: 1

System Rent Accounts: 1

Lease Ref: 1

Modifications: 1

Lease Prop Ref: 1

Lease Expenditure: 1000.0

Lease Landlords: 1

System Leases: 1

Lease Acc Ref: 1

Modifications: 0

Lease Acc Expenditure: 1000.0

Lease Ref: 1

System Lease Accounts: 1

Contract Ref: 1

Lease Employee Ref: 1

Lease Job Role: MNGR

System Contracts: 1

Employee Acc Ref: 1

Employee Acc Salary: 30000.0

Contract Ref: 1

System Employee Accounts: 1

As you can see from the above print out, I am able to create all of the system objects, and load these at system start up, although with some of the loading of the modifications, and loading of previous documents, it seems that these are not all working correctly, but unfortunately due to time constraints I have been unable to track and fix these bugs.