**UNIVERSITY OF HERTFORDSHIRE**

**Faculty of Science Technology and the Creative Arts**

**Modular MSc Honours in Computer Science**

**7WCM0031 Software Engineering MSc Project**

**Interim Project Report**

**September 2015**

**Development of a distributed system for ‘MSc Properties’**

**Mr D L Edwards**

**Table of Contents**

Table of Contents……………………………………………………………….…………………………………………………i

1 Completed work……….…………………….…………………………………………………………………………………1

2 Work to be completed……………………………………………………………………………………………………….3

3 Gantt chart…………………………….......…………………………………………………………………………………….5

4 What makes a good MSc project ………………………………………………………………………………………9

5 Bibliography…………………………………………………………………………………………………………………….10

**Completed Work**

Since the submission of the Extended Project Proposal, I have completed the below work for my MSc dissertation:

* Completed literature search and review to identify methods and methodology that will be used to achieve the project aim and objectives.
* Completed mapping of system requirements and documented in requirements specification.
* Completed mapping of business data flow and documented in data flow diagrams.
* Completed mapping of system graphical user interface and documented in storyboard.
* Completed mapping of interaction between classes in order to carry out system functions and documented in sequence diagrams.
* Normalized system data and mapped my class diagram to an entity relationship model.
* Produced dummy data for the system.
* Written 80% of the unit test scripts for system testing.
* Created the MySQL database which will store system data.
* Using the divide and conquer design principle, I have broken the system into a series of smaller things to be completed.
* Continually updating Project Gantt Chart.
* Continually tracking updates to project through version concurrent system
* Carried out coding of the following classes for the server side package;
  + AccountImpl implements Account
  + AddressImpl implements Address
  + AddressUsageImpl Implements AddressUsage
  + AgreementImpl implements Agreement
  + ApplicationImpl implements Application
  + ContactImpl implements Contact
  + ContractImpl extends AgreementImpl implements Contract
  + ElementImpl implements Element
  + EmployeeImpl implements Employee
  + EmployeeAccountImpl extends AccountImpl implements EmployeeAccount
  + InvolvedPartyImpl implements InvolvedParty
  + JobRoleImpl implements JobRole
  + JobRoleBenefitImpl implements JobRoleBenefit
  + LandlordImpl implements Landlord
  + LeaseImpl extends AgreementImpl implements Lease
  + LeaseAccountImpl extends AccountImpl implements LeaseAccount
  + ModifiedByImpl implements ModifiedBy
  + OfficeImpl implements Office
  + PersonImpl implements Person
  + PropertyImpl implements Property
  + PropertyElementImpl implements PropertyElement
  + RentAccountImpl extends AccountImpl implements RentAccount
  + TenancyImpl extends AgreementImpl implements Tenancy
  + TransactionImpl implements Transaction
  + UserImpl implements User
* Further to the creation of the above server side classes to carry out the business logic, which is the model in the model-view-controller (MVC) concept, I have also created a Database class which stores a number of lists of objects to replicate the business data, and have implemented a connection to the MySQL database using JDBC but am yet to implement the select, update and create statements.
* I have also created a ServerImpl class which is the controller in the MVC concept, and although I have not implemented all of the controlling code, the class creates a ServerImpl object known as a server stub and uses RMI to register this server stub as an RMI object along with the IP address in a remote object registry to allow it to be assessed by client objects.
* Carried out coding of the following interfaces for the common package;
  + Account
  + Address
  + AddressUsage
  + Agreement
  + Application
  + Client
  + Contact
  + Contract extends Agreement
  + Element
  + EmployeeAccount extends Account
  + Employee
  + InvolvedParty
  + JobRoleBenefit
  + JobRole
  + Landlord
  + LeaseAccount extends Account
  + Lease extends Agreement
  + ModifiedBy
  + Office
  + Person
  + Property
  + PropertyElement
  + RentAccount extends Account
  + Server
  + Tenancy extends Agreement
  + Transaction
  + User
* Carried out coding of the following classes for client side package;
  + HomeForm extends JFrame
  + CreatePersonForm extends JFrame
  + CreateContactForm extends JFrame
  + CreateAddressForm extends JFrame
  + ClientLogin extends JFrame
* Further to the above client side classes, which is the view in the MVC concept, I have created a ClientImpl class which uses the singleton pattern to return a unique stub for this client instance and registers that client instance with the server stub stored on the remote object registry, this will allow the client user to connect with the server to invoke remote methods.
* Lastly I have coded test classes to carry out unit testing of the following classes;
  + AccountImpl
  + AddressImpl
  + AddressUsageImpl
  + AgreementImpl
  + ContactImpl
  + TenancyImpl

**Word Count = 624**

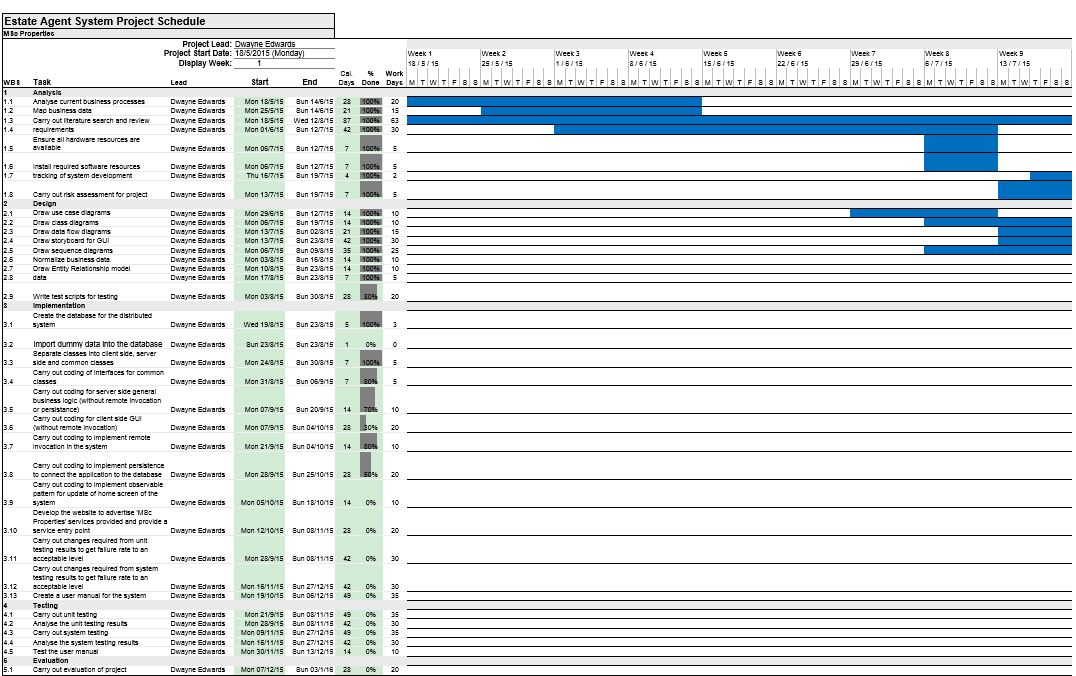
**Work to be completed**

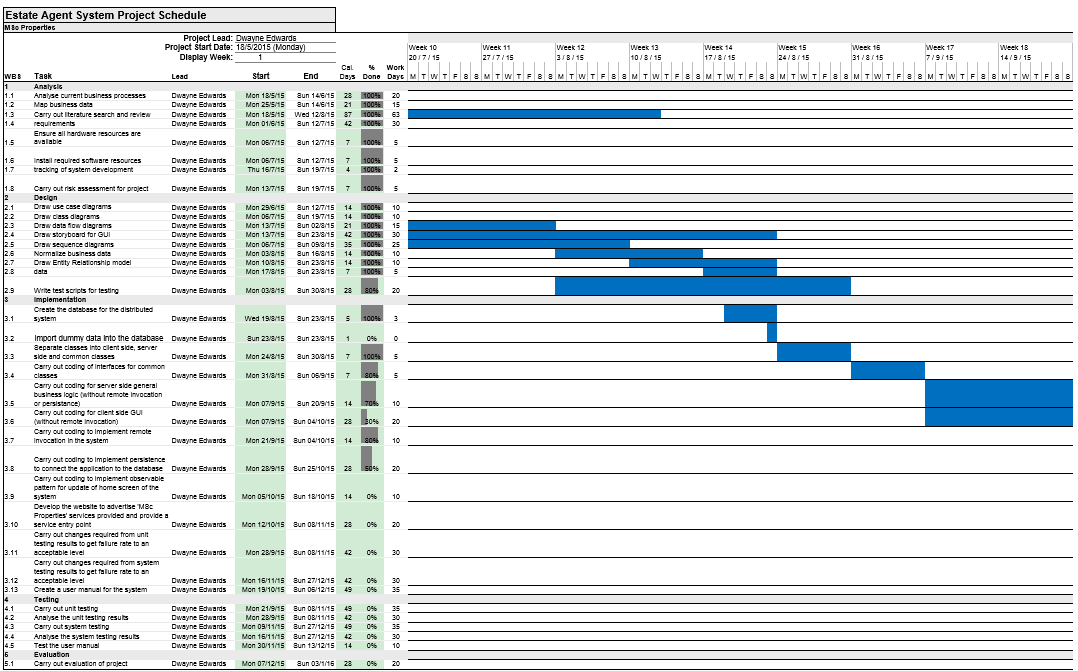
Before the submission of the Final Project Report, I still have to complete the below work for my MSc dissertation:

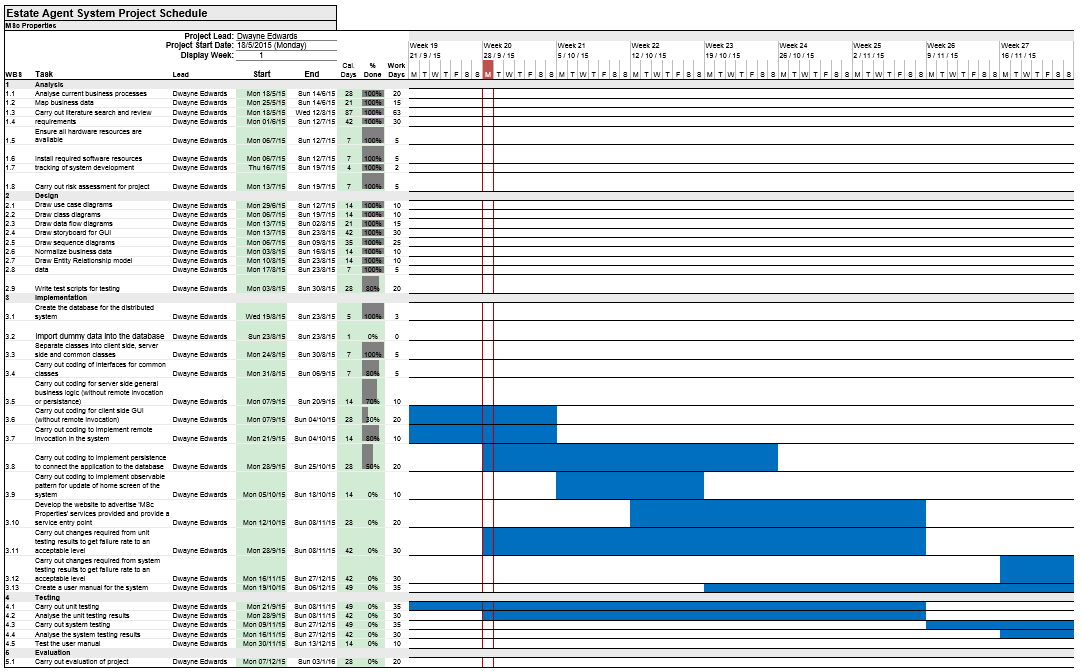
* Complete the writing of the remaining 20% of the unit test scripts.
* Complete the coding for the Database class, to allow for the system to select data using SQL select queries, update data using SQL update statements, and create data using SQL insert statements for the MySQL database to carry out system functionality.
* Complete the coding for the ServerImpl class, to allow for the ServerImpl class to receive method calls from the client side package, and control this communication between the view (graphical user interface – client package) and the model (classes which deal with the business logic).
* Complete the coding of test classes for unit testing of the following classes;
  + ApplicationImpl
  + ContractImpl
  + ElementImpl
  + EmployeeImpl
  + EmployeeAccountImpl
  + InvolvedPartyImpl
  + JobRoleImpl
  + JobRoleBenefitImpl
  + LandlordImpl
  + LeaseImpl
  + LeaseAccountImpl
  + ModifiedByImpl
  + OfficeImpl
  + PersonImpl
  + PropertyImpl
  + PropertyElementImpl
  + RentAccountImpl
  + TransactionImpl
  + UserImpl
  + ClientImpl
  + All GUI classes for client package
* Write the system test scripts.
* Continually update the project Gantt chart.
* Continually tracking updates to project through version concurrent system.
* Import dummy data into the MySQL database.
* Implement the observable pattern to allow for changes to the MSc Properties Agreements to update the office watch lists (MSc Properties agreements due to expire), which will allow for all observers (clients) to be notified of changes to these watch lists and can then update the view of the system (graphical user interface – client package) for each client.
* Develop the website to advertise MSc Properties’ services such as Properties available to let, Property management services and Job vacancies, and provide an online entry point into the MSc Properties services through the website.
* Implement the document management functionality using the JFileChooser class on the client side package, and storing the File into the Database class and uploading that to the MySQL database into a BLOB field.
* Carry out unit testing.
* Analyse the unit testing results.
* Make changes to the system off the back of the analysis of the unit testing results.
* Carry out system testing.
* Analyse the system testing results.
* Make changes to the system off the back of the analysis of the unit testing results.
* Create a user manual to assist clients in using the MSc Properties system.
* Test the user manual.
* Make any changes to the user manual off the back of the testing.
* Carry out evaluation of the project as a whole, outlining what went well and what went wrong, and what would have been done better?
* Write up the final project report which will contain;
  + A statement of the problem and its context.
  + Formulation and refinement of the question in the problem domain.
  + Objectives of the project.
  + Discussion of the methodology used to explore or address the issue, selection of approach, selection of techniques, and selection of evaluation approach.
  + Application of the methodology and evaluation of the results.
  + Discussion: conclusion, evaluation of the whole project and any future work.

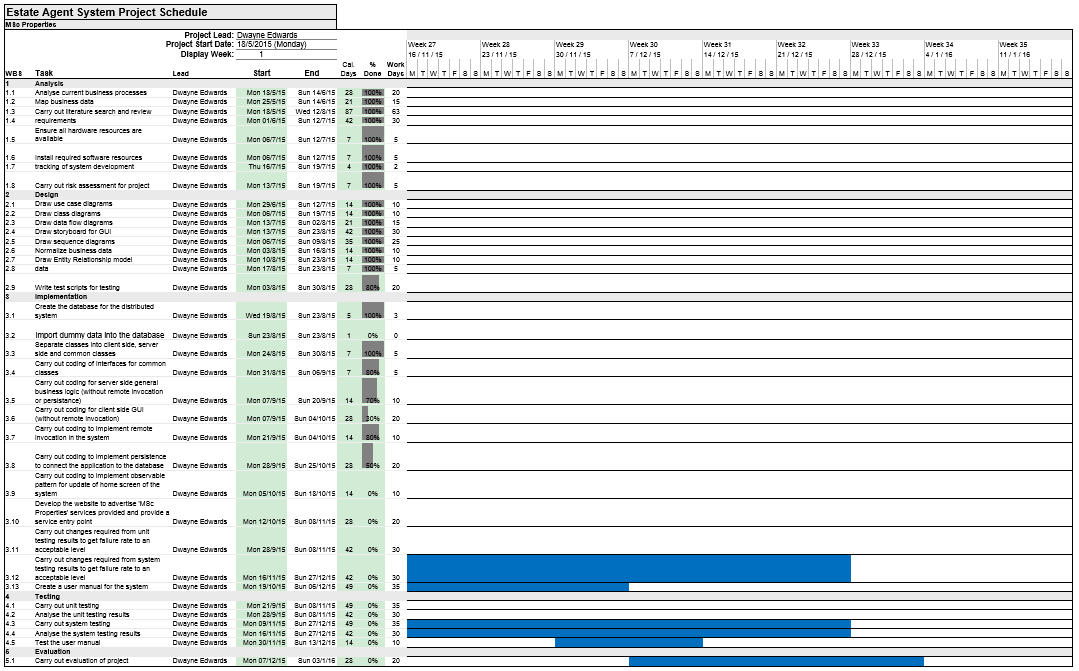
All of the work completed and work to still be completed is outlined in my updated project Gantt chart below.

**Word Count = 536**









**What makes a good MSc project?**

**Summary of audio file “What make a good MSc project”?**

Bodo Scholz describes a good MSc project as delivering some new or incremental knowledge, and in doing so combining background research and work that delivers a tangible result.

Bruce Christianson describes it as three key areas, demonstrating technical competency, engaging with a real problem, and producing a dissertation that tells a story.

James Malcom describe it as not one that just develops software or reads lots of books, but by answering a question of interest through some piece of practical or experimental work.

Paul Wernick describes it as something that is not too small and can be expanded upon. Also something that the student can own and make a success of.

I believe the audio file in essence describes a successful MSc project as a piece of background research combined with an overall objective, that then directs the creation of a practical piece of work or an experiment to produce an outcome, that has the possibility of expansion.

**How the issues are likely to affect my project?**

I believe that the issues discussed in the audio file “What makes a good MSc project” are likely to affect my project work because although I am not creating a new software engineering method/methodology, I am researching existing technology available to solve a real business problem, and in doing so I hope to demonstrate a competent skill level in a number of different areas within software engineering by producing a distributed system.

Furthermore, in the selection of my MSc project, I have selected a project that I have a long connection with, having worked in the Housing sector for a number of years and which I have a deep passion for, which means that I will stay interested in the topic area. I also hope to have selected an area that has a wide scope to enable me to express a wide range of skills, in the hope that I do not limit my project and in turn stop me from showing my full skill set.

**Word Count = 349**

**Bibliography**

1. Scholz, B, Christianson, B, Malcolm, J and Wernick, P. (2015). *What make a good MSc project?.* Available: http://www.studynet2.herts.ac.uk/web/crs/14/7WCM0003-0602/resources/question01.html. Last accessed 28th Sep 2015.
2. Sommerville, I. (2011). Introduction. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P1-26.
3. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Charecterization of Distributed Systems. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P17-52.
4. Sommerville, I. (2011). Distributed software engineering. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P479-507.
5. Sommerville, I. (2011). Software processes. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P27-55.
6. Cockburn, A. and Highsmith, J. and Bohem, B. (2001). Agile Software Development: The Business of Innovation. *Computer*. 1 (1), p131-133
7. Sommerville, I. (2011). Requirements engineering. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P82-117.
8. Sommerville, I. (2011). System modeling. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P118-146.
9. Connolly, T. and Begg, C. (2005). Normalization. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P387-414.
10. Connolly, T. and Begg, C. (2005). Entity-Relationship Modeling. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P387-414.
11. Sommerville, I. (2011). Design and implementation. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P176-204.
12. Reges, S. and Stepp, M. (2011). Graphical User Interface. In: Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Building Java Programs A Back to Basics Approach.* 2nd ed. Boston: Pearson. P846-909.
13. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Remote Invocation. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P201-246.
14. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Distributed Objects and Components. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P351-396.
15. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Transactions and Concurrency Control. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P691-742.
16. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Distributed Transactions. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P743-780.
17. Connolly, T. and Begg, C. (2005). SQL: Data Manipulation. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P112-156.
18. Connolly, T. and Begg, C. (2005). Security. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P541-571.
19. Connolly, T. and Begg, C. (2005). Transaction Management. In: McGettrick, A. *Database Systems A Practical Approach to Design, Implementation and Management.* 4th ed. United States of America: Pearson. P572-629.
20. Coulouris, G. and Dollimore, J. and Kindberg, T. and Blair, G. (2012). Security. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Distributed Systems Concepts and Design.* 5th ed. United States of America: Pearson. P479-536.
21. Sommerville, I. (2011). Software testing. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P205-233.
22. Sommerville, I. (2011). Project management. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P593-617.
23. Sommerville, I. (2011). Project planning. In: Horton, M. and Hirsch, M. and Goldstein, M. and Bell, C. and Holcomb, J. *Software Engineering*. 9th ed. Boston: Pearson. P618-650.
24. Citizens Advice. (2015). *The benefit cap - what you need to know.* Available: https://www.citizensadvice.org.uk/benefits/the-benefit-cap/the-benefit-cap-what-you-need-to-know/. Last accessed 20th Jun 2015.
25. Enfield Council. (2013). *Enfield's Homelessness Strategy 2013-2018.* Available: http://www.enfield.gov.uk/download/downloads/id/8004/enfields\_homelessness\_strategy\_2013-2018. Last accessed 20th Jun 2015.
26. Hunt, B. (2015). *UH Ethics Approval.* Available: http://www.studynet2.herts.ac.uk/ptl/common/ethics.nsf/Homepage?ReadForm. Last accessed 19th Jul 2015.
27. Waldo, J. (1998). Remote procedure calls and Java Remote Method Invocation. *Concurrency, IEEE*. 6 (3), P5-7.
28. Guan, H. and Ip, H. and Zhang, Y. (1998). Java-based approaches for accessing databases on the Internet and a JDBC-ODBC implementation. *Computing & Control Engineering Journal*. 9 (2), P71-78.