**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 What makes a good MSc project?.…………………………………………………………………………………….5

4 Bibliography…………………………………………………………………………………………………………………….6

**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 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.
* 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 created test classes to carry out unit testing of the following classes;
  + AccountImpl
  + AddressImpl
  + AddressUsageImpl
  + AgreementImpl
  + ContactImpl
  + TenancyImpl

**Work to be completed**

**What makes a good MSc project?**

**Bibliography**

1. 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.
2. 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.
3. 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.
4. 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.
5. Cockburn, A. and Highsmith, J. and Bohem, B. (2001). Agile Software Development: The Business of Innovation. *Computer*. 1 (1), p131-133
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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.
24. 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.
25. Hunt, B. (2015). *UH Ethics Approval.* Available: http://www.studynet2.herts.ac.uk/ptl/common/ethics.nsf/Homepage?ReadForm. Last accessed 19th Jul 2015.
26. Waldo, J. (1998). Remote procedure calls and Java Remote Method Invocation. *Concurrency, IEEE*. 6 (3), P5-7.
27. 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.