**UNIVERSITY OF HERTFORDSHIRE**

**Faculty of Engineering and Information Sciences**

**Modular MSc Honours in Computer Science (Software Engineering)**

**7WCM0031 Software Engineering MSc Project (Online)**

**Final Project Report**

**January 2016**

***Development of a distributed data and document management system for ‘MSc Properties’***

**Mr D L Edwards**

**SRN: 07156987**

**Supervised by: Thiago Matos**

Abstract

Acknowledgements

Table of Contents

1. Introduction
   1. Introduction to the project

For my MSc Computer Science Dissertation (Software Engineering), I decided to solve the problem of data and document management through the implementation of a distributed system for a fictional estate agent called ‘MSc Properties’.

The reason for me undertaking this project is that there is an ever increasing demand for private rented accommodation due to changes in the social housing market, social benefits offered, and unemployment rates, resulting in less social housing being available and reduced benefits for people on a low income, who would normally need to make use of social housing and/or benefits offered by the government, meaning families are unable to afford the private rents of their current area they live in and need to move to a different area they can afford to privately rent.

* + 1. **Report Structure**

During this report I will document the aim and objectives of this project and explain the problem background in more detail, looking at the different factors that contribute to the problem identified.

I will then document the research and literature review I carried out in order for me to identify the different software engineering techniques and technologies available to solve the problem.

I will then go on to document the design of the distributed system, outlining the different software engineering techniques chosen, and why these were chosen. I will then explain how these were used to enable me to structure and manage the project, and just as importantly allow me to produce diagrams that virtualized the structure, behaviour and interaction of the distributed system.

Furthermore, I will document the implementation and testing of the distributed system, explaining which techniques and technologies I chose, and why they were the best solution for this project, given the available resources. I will then go on to provide an analysis of the test results to draw some conclusions on the validity of the software produced.

Lastly I will document my evaluation of the project as a whole, outlining what went well, what didn’t go so well, and what I would do different if I was to do the project again. I will then go on to stating whether or not I have successfully achieved the project aim and objectives.

* 1. Project Aim

The aim of this project is to tackle the issues of data and document sharing across the Internet by developing a distributed data and document management system for a fictional estate agent called ‘MSc Properties’.

The distributed system should allow ‘MSc Properties’ to share business data and documents across the Internet, whilst providing data security and integrity. ‘MSc Properties’ requires the distributed system to be maintainable, dependable and usable, which means I will explore the different techniques that support program specification, design, validation and evolution of software.

* 1. Project Objectives
     1. Core Objectives

Analyse ‘MSc Properties’ current business processes by week 5.

Complete literature searches and review of existing data management systems, identifying the software engineering models, methodologies, tools and metrics used in the development process by week 12.

Set out functional and non-functional requirements for the development within the requirements document by week 9.

Ensure required resources are available for the entire project by week 9.

Carry out risk assessment by week 10.

Set out the distribution mechanism I am going to employ for the distributed system by week 13.

Develop a suitable data management system model that meets the requirements defined by week 15.

Write test scripts to test the implementation of the system outlined in the development model by week 15.

Develop a suitable database to handle the business data and import dummy data into the database by week 16.

Develop a suitable application to handle the business processes and connect to the database to store the business data by week 24.

Develop a suitable search facility so users can search for information stored in the database, and should be implemented by week 24.

Develop reporting functionality so certain users can report on business performance indicators by week 24.

Develop a log in facility for users, allowing for restricted access, and to prevent unauthorised access and should be implemented by week 26.

Test the system using the test scripts created, ensuring the test results are above the acceptable failure rate defined in the requirements by week 33.

Develop and test a user manual by week 31.

Evaluate the project in a report to detail the entire development and outline what went well and what could have been done better by week 34.

* + 1. Advanced Objectives

Develop a website to advertise services offered to potential customers/suppliers. Customers/Suppliers will be able to register and submit a service request through the website and should be implemented by week 26.

Develop document management facility that allows for documents to be stored electronically, and should be implemented by week 26.

Develop a home screen which provides a live feed of the tenancies and leases due to expire by week 26.

Develop a reset password facility, so users are able to reset their password if they have forgotten it allowing users to establish access to the system. This should be implemented by week 26.

* 1. Project Background

‘MSc Properties’ is a fictional estate agent with a number of sites nationwide across England. Due to the current unemployment rates, and recent legislation changes resulting in local councils being able to house homeless families outside of the local borough [25] and benefit caps [23] meaning families have to move out of their local borough due to not being able to afford local rents [25], ‘MSc Properties’ require the need to be able to transfer customers between sites, meaning the transfer of data and documents across sites that could be 100’s of miles apart.

‘MSc Properties’ require me to develop a distributed system to create and manage their property portfolio, as well as creating and managing customer/supplier/employee accounts. They require this so that data can be stored on a server or locally and all the officers of different ‘MSC Properties’ sites will be able to access this data. The system will have a login facility to provide restricted access for users, and will also allow managers of ‘MSc Properties’ stores to manage their employee accounts. The system will also allow ‘MSc Properties’ managers to report on business data.

‘MSc Properties’ have allocated an office manager to the project to assist with analysis of current business processes, map system requirements, and communicate back to ‘MSc Properties’ with the work that is occurring in project meetings, and relaying any project document back to the business for sign off. From the project meetings the below business analysis and system requirements were compiled.

* + 1. **Business Analysis**

‘MSc Properties’ currently have an electronic filing system, where customers, landlords of properties and potential employees fill out an application form to request a service from ‘MSc Properties’. Anyone applying for a service will have to provide their personal information. Customers will also need to provide the properties they have an interest in letting. Landlords will have to provide the property information for the property they would like managed. Potential employees will have to provide the job role they are applying for.

Part of the landlord/customer/employee sign up process is to bring in a number of documents confirming their information, and once customers have agreed on a property to let, once ‘MSc Properties’ have agreed to manage a property for a landlord, or once ‘MSc Properties’ have agreed to take on an employee, contracts will be drawn up and signed by both ‘MSc Properties’ office manager and the client, and is called the agreement.

Each agreement will result in a transaction account being created and documented in an Excel spreadsheet, which will record all transaction information for the different agreements, for example, rent payments from tenants, salary payments to employees, or lease payments to landlords. All of this information captured in the form along with any documents and spreadsheets will be stored electronically the computer of the officer managing the case, except for the employee information which is stored on the office manager’s computer.

‘MSc Properties’ currently advertise their property portfolio and job role vacancies through pictures in their store windows. Office managers carry out a revenue monitoring exercise each month, going through each transaction account, and carrying out budget monitoring and cash flow forecasting, along with reports for the month’s business activity. Also there is a staff hierarchy within ‘MSc Properties’ where specific tasks and responsibilities are assigned to certain staff roles meaning that not all staff can carry out each task due to privileges.

* + 1. **System Requirements**

NEED TO DO SYSTEM REQUIREMENTS

* 1. Literature Review

As previously outlined, I will now document the literature reviews I undertook to allow me to understand the different options available to me to enable me to successfully achieve the project aim and objectives.

* + 1. **Design Methodologies**

The problem I am trying to solve by undertaking this project, requires a piece of software to be developed for me to successfully achieve the project objectives surrounding data and document sharing. But as well as trying to solve the business problem surrounding data and document sharing, there is also the problem of producing software that is maintainable, dependable and usable, which will require me to undertake literature reviews into different design decisions I will need to make when designing the system.

* + - 1. **Software Development Approach**

The first design principle I am going to discuss is the software development approach. This is because as this is a large project, it requires the project to go through a structured development process to give the project the highest possibility of successfully achieving the project aim and objectives.

The piece of literature I am going to review for the software development approach is the article called *“Software Quality & Agile Methods”* written by *M. Huo, J. Verner, L Zhu and M.A. Babar.* This article looks at the quality of software produced when comparing the Waterfall Model and Agile methods, and specifically how agile methods can achieve high quality software even if the process is not linear and a complete requirements specification has not be developed prior to the design and implementation stage of the development.

The article then goes on to conclude that “agile methods do have practices that have Quality Assurance abilities, some of them are inside the development phase and some others can be separated out as supporting practices. The frequency with which these agile Quality Assurance practices occur is higher than in a waterfall development and lastly, agile Quality Assurance practices are available in very early process stages due to the agile process characteristics”. The below diagrams show the different methods and Quality Assurance techniques undertaken within the Waterfall Model and Agile Methods.

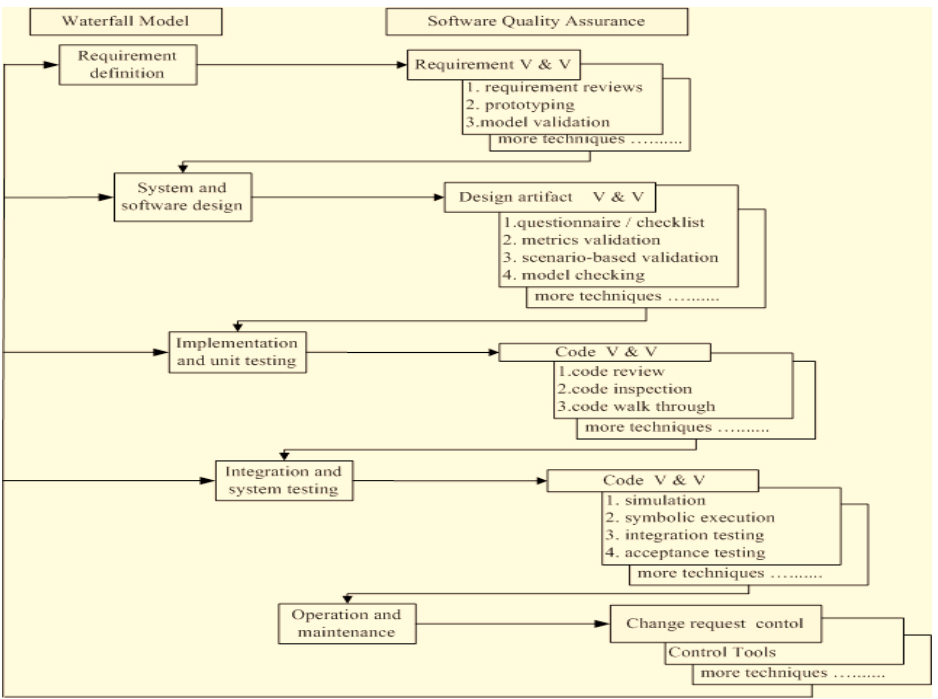


Fig 1 – Waterfall Process Model with Quality Assurance Techniques

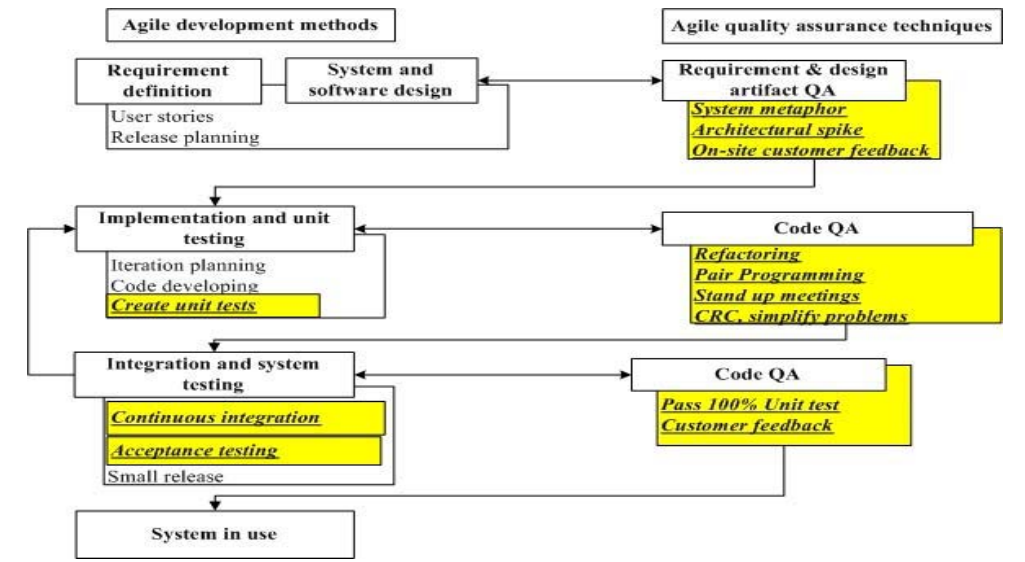


Fig 2 – Agile Development Methods with Quality Assurance Techniques

* + - 1. **Modelling System Behaviour**

The next design principle I am going to discuss is modelling system behaviour. This is because this project centres on software development, and part of the aim is to produce software that is maintainable, dependable and usable, so it will require me to produce high quality software, and by modelling system behaviour and understanding how the elements of the system will interact with each other.

The piece of literature I am going to review for the modelling of system behaviour is the article called *“Designing Concurrent, Distributed, and Real-Time Applications with UML”* written by *H. Gomaa.* This article looks at two areas, the software design method called Concurrent Object Modelling and Architectural Design Method (COMET), which is an example of a Unified Modelling Language (UML) based method, and the different modelling required for concurrent, distributed and real time applications using UML.

The article explains that “In the requirements model, the system is considered as a black box and the use case is developed… In the analysis model, the emphasis is on understanding the problem, hence the emphasis is on identifying the problem domain objects and the information passed between them… In the design model, the solution domain is considered, so the analysis model is mapped to a concurrent model”. These different models are what I will have to consider when going through the software development process for this project.

Below is a list of different techniques to model system behaviour which I have come across during the research for this project:

* Use Case Diagrams – A representation of a user’s interaction with the system, showing the relationship between the user and the use cases they are involved in [Wiki].
* Data Flow Diagrams – A graphical representation of the “flow” of data through an information system [Wiki].
* Class Diagram - A static structure diagram that describes the structure of a system showing the system classes, their attributes, methods, and the relationships amongst objects [Wiki].
* Entity Relationship Diagram – A data model for describing the data or information aspects of business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as relational database [Wiki].
* Class Responsibility Collaborator (CRC) Models – A brainstorming tool used in the design of object-oriented software, documenting the dynamics of object interaction and collaboration [Wiki].
* Sequence Diagrams – An interaction diagram that shows how processes operate with one another and in what order, showing object interaction arranged in time sequence [Wiki].
* Storyboards – A graphic organizer in the form of illustrations or images displayed in sequence for the purpose of pre-visualising a motion picture, animation, motion graphic or interactive media sequence [Wiki].
  + - 1. **Design Patterns**

The last design principle I am going to discuss is design patterns, and most importantly the design patterns I can employ within the software I am going to produce. Again as with modelling system behaviour previously, this project centres on software development, and part of the aim is to produce software that is maintainable, dependable and usable, so it will require me to produce high quality software, and by implementing design patterns, it will allow me to produce software with high cohesion, low coupling, encapsulation, and other metrics of software development which indicate high quality software.

Ian Sommerville explains design patterns as a description of accumulated wisdom and experience, a well-tried solution to a common problem, and the Hillside Group puts it as “Patterns and Pattern Languages are ways to describe best practices, good designs, and capture experience in a way that it is possible for others to reuse this experience” [].

Below are some design patterns I have come across during my research for this project:

* Observer pattern – A software design pattern in which an object, called the subject, maintains a list of dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods [Wiki].
* Singleton pattern – A software design pattern that restricts the instantiation of a class to one object [Wiki].
* static modifier –
* Strategy pattern – A software design pattern that enables an algorithm’s behaviour to be selected at runtime [Wiki].
* Creational pattern – A software design pattern that deals with object creation mechanisms, trying to create objects in a manner suitable to the situation [Wiki].
* Iterator pattern – A software design pattern in which an iterator is used to traverse a container and access the container’s elements [Wiki].
* Composite pattern – A partitioning software design pattern, which describes that a group of objects is to be treated in the same way as a single instance of an object, allowing clients to treat individual objects and compositions uniformly.
* Inheritance – Is when an object or class is based on another object or class, using the implementation to maintain the same behaviour, and is a mechanism for code reuse and in programming languages that support inheritance, produce an “is a” relationship between sub classes and its parent class. [Wiki].
* Object Composition – Is a way to combine simple objects or data types into more complex ones, and are a critical building block of many data structures. Composition can be regarded as a relationship between types: an object of a composite type “has an” object of a simpler type [Wiki].
* Object Relational Mapping – Is a programming technique for converting data between incompatible type systems in object-oriented programming languages, and in effect creates a “virtual object database” that can be used within the programming language.
  + 1. **Development Methodologies**

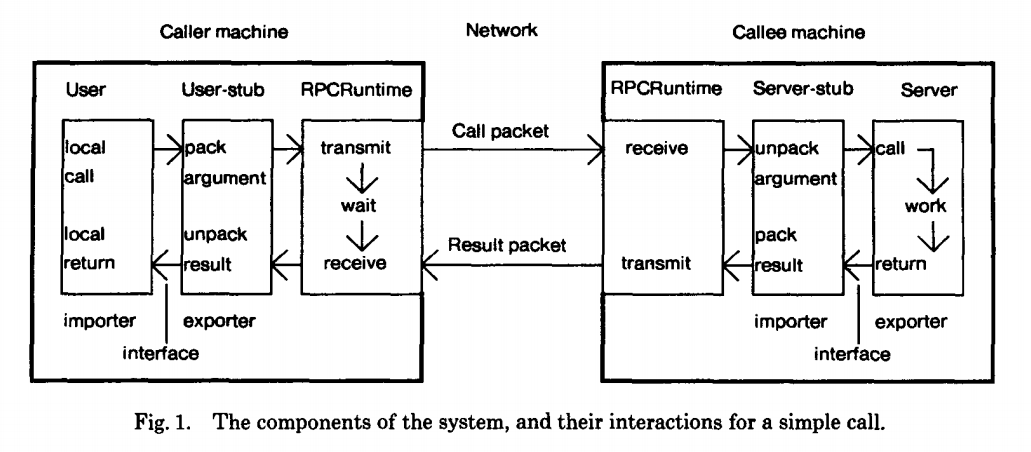
I am now going to discuss the different development methodologies I uncovered during the course of the project, whilst undertaking the literature review.

* + - 1. **Networking**

One of the major problems I am trying to solve by undertaking this project, is to be able to produce a system that allows ‘MSc Properties’ to share data and documents across the Internet, whilst ensuring that the validity of the data and documents being shared are maintained. For me to do this, it has meant that I have had to undertake a literature review into the different networking technologies that will enable me to successfully achieve the project aim and objectives related to networking.

The first piece of literature I am going to review for Networking is the article called *“Implementing Remote Procedure Calls”* written by A.D. Birrell and B.J Nelson. This article talks about the options that face the designer implementing remote procedural call (RPC) functionality and the considerations that need to be made when making decisions on this type of system.

The article states “when making a remote call, five pieces of program are involved: the *user*, the *user-stub*, the RPC communications package (the RPCRuntime), the *server-stub*, and the *server*… When the user wishes to make a remote call, it actually makes a perfectly normal local call, which invokes a corresponding procedure in the user-stub. The user-stub is responsible for placing a specification of the target procedure and the arguments into one or more packets and asking the RPCRuntime to transmit these reliably to the callee machine. On receipt of these packets, the RPCRuntime in the callee machine passes them to the server-stub. The server-stub unpacks them and again makes a perfectly normal local call, which invokes the appropriate procedure in the server. Meanwhile, the calling process in the caller machine issues pended awaiting a result packet. When calling the server completes, it returns to the server stub and the results are passed back to the suspended process in the caller machine. There they are unpacked and the user-stub returns them to the user.” This process is represented in the below figure.



The Remote Procedural Call functionality highlights the type of technology I could implement to successfully achieve the project aim and objectives of data sharing amongst ‘MSc Properties’ hosts that are at different locations.

The second piece of literature I am going to review for Networking is the article called “*Push vs. Pull in Web-based Network Management”* written by *J.P. Martin-Flatin*. This article talks about two models of network management, which are “The Pull Model” and “The Push Model”, which represent two well-known approaches to exchanging data between two hosts with a distance between them, which is one of the major problems I am trying to solve during this project.

The article states “The pull model is based on the request/response paradigm, the client sends a request to the server, then the server answers, either synchronously or asynchronously. This is functionally equivalent to the client “pulling” the data off the server. In this approach, the data transfer is always initiated by the client, i.e. the manager. The push model, conversely, is based on the publish/subscribe/distribute paradigm. In this model, agents first advertise what Management Information Bases they support, and what Simple Network Management Protocol notifications they can generate; the administrator then subscribes the manager (the Network Management Station) to the data he/she is interested in, specifies how often the manager should receive this data, and disconnects. Later on, each agent individually takes the initiative to “push” data to the manager, either on a regular basis via a scheduler (e.g., for network monitoring) or asynchronously.” The article then goes on to state that “the pull model, well suited to ad hoc management, and the push model, well adapted to regular management”.

* + - 1. **Document Management**

The next development area I am going to discuss is document management, which is one of the biggest areas of ‘MSc Properties’, as the business deals with numerous documents that need to be stored, and available for access by any host at different locations.

The piece of literature I am going to review for Document Management is the article called *“Electronic Document Management: Challenges and Opportunities for Information Systems”* written by *R.H. Sprague, Jr.* This article talks about a number of benefits gained from implementing a document management system, such as improving the publication process, supporting organisational processes and communication amongst people and groups, improving access to external information, creating and maintaining documents, maintaining corporate records and lastly promoting training education.

The article states “Documents are stored electronically, shipped over a network and printed when and where they are needed, resulting in reduction in obsolescence, elimination of warehouse costs, and reduction or elimination of delivery time.”, the article later goes on to explain “The benefits of Electronic Document Management for these applications are, quicker access to the documents, more efficiency in the search process, simultaneous access by several people to the most current version of the document, and reduced cost of printing and distributing documents”.

Below are some document management frameworks I have come across during my research for this project:

* Apache JackRabbit – Is a content repository which implements the Content Repository for Java (JCR), with support for structured and unstructured content full text search, versioning, transactions, observation and more [].
* Modeshape – Is a distributed, hierarchial, transactional, and consistent data store with support for queries, full-text search, events, versioning, refrences, and flexible and dynamic schemas, which implements the Contant Repository for Java (JCR) [].
  + - 1. **Task Scheduling**

The next development area I am going to discuss is task scheduling, and in particular task scheduling in real time systems, this is because the distributed system that will be developed for ‘MSc Properties’ during this project will be a real time system, dealing with the processing of scheduled tasks.

The piece of literature I am going to review is the article called *“Application of Real-Time Monitoring to Scheduling Tasks with Random Execution Times”* written by D. Haban and K.G. Shin. This article talks about the calculation of execution time for posteriori tasks (calculation of execution time requires experience with the given task) scheduled within real time systems, and that the worst-case execution time is usually used to ensure that enough time has been allocated for the task to be completed, and discusses the drawbacks with this sort of approach and alternatives to this method.

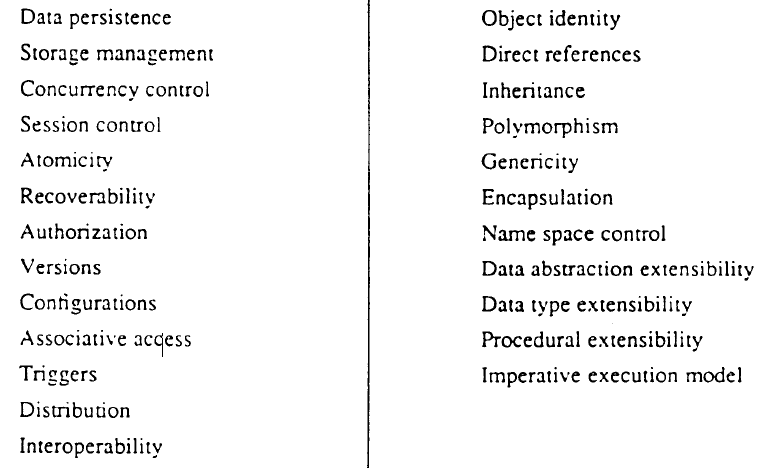
The article states “real-time tasks are usually scheduled based on their worst-case execution time, and since the worst-case execution time can be several orders of magnitude larger than the true execution time, scheduling tasks based on the worst-case execution time can lead to severe underutilization of CPU cycles and/or incorrect decision on the schedulability of tasks i.e., some tasks are declared to be un-schedulable even if they can be completed in time”.

Below are some task scheduling frameworks I have come across during my research for this project:

* Quartz project -
* Haban & Shin project
  + - 1. **Database Management Systems**

The next development area I am going to discuss is database management systems, and in particular data models based on object-orientated concepts. This is because I have decided to develop a distributed system using object-orientated concepts, such as objects, inheritance, Encapsulation etc., which means I will need to adopt a database management model based on these object-orientated concepts.

The piece of literature I am going to review is the article called *“An Introduction to Object-Oriented Database and Database Systems”* written by *M.L. Horowitz.* This article looks at the possibilities of combining most of the desirable features of database systems with desirable features of the object orientated model of computation and the below diagram outlines these features.



The article later goes on to explain the issues that can arise when combining the database and object-oriented model concepts, by stating that “First integration should occur without impedance mismatch. In particular, language support for object-oriented database services should be orthogonal and transparent. Second, integration should not lose any advantages of existing data models. For instance, object-oriented programming does not support data independence inherently, so features such as relationship support and query joins should be provided. Finally, integration presents an opportunity for introducing new desirable features”.

Below are some database management systems I have come across during my research for this project:

* Microsoft Access – Is a Database Management System (DBMS) from Microsoft that combines the relational Microsoft Jet Database Engine with graphical user interface and software development tools [Wiki].
* MySQL – Is an open-source relational database management system (RDBMS) and the most widely used open-source client-server model RDBMS [Wiki].
  + - 1. **Web Server**

The next development area I am going to discuss is web servers, and in particular the different web server software that can be implemented to receive and manage the Hypertext Transfer Protocol requests sent by a user of the website that will be developed to advertise ‘MSc Properties’ services to potential customers, and also manage communication between the web server and the database management system discussed previously.

The piece of literature I am going to review is the article called *“Specification and Implementation of Dynamic Web Site Benchmarks”* written by *C. Amza, A. Chanda, A.L. Cox, S. Elnikety, R. Gil, K. Rajamani and W. Zwaenepoel.* This article looks at the movement from web content being static HTML or image files, to web content becoming dynamic through the combination of a front end web server (web browser such as Internet Explorer), an application server (software such as Apache, along with server side scripting in PHP and SQL), and a back-end database (software such as Microsoft Access) and in particular identifies benchmarks for dynamic web sites by comparing 3 different dynamic web sites, looking at the bottleneck characterizations for these web sites.

This combination of technology would allow me to develop a dynamic website that can advertise ‘MSc Properties’ services to potential customers, and be updated by staff when they perform actions in the system which result in updates to the website, for example, a property being rented to a customer meaning the property is no longer available to rent and therefore should no longer be advertised on the website as available to rent.

The article explains that “We have used our implementations to carry out a bottleneck characterization of the benchmarks. Different benchmarks show different bottlenecks: the database CPU for the online bookstore, and the Web server CPU for the auction site and the bulletin board. Complex queries cause the database CPU to the bottleneck for the online bookstore. In contrast, the queries for the other applications are simpler.

Below is some web server software I have come across during my research for this project:

* Apache HTTP Server – Is the worlds most used web server software, and played a key role in the growth of the World Wide Web quickly becoming the most dominant HTTP server [].
* Nginx – Is a web server with a strong focus on high concurrency, performance, and low memory usage [].
* Cherokee – Is an open source cross-platform web server that runs on Linux, BSD, variants, Solaris, Mac OS X, and Microsoft Windows [].
  + - 1. **Graphical User Interface**

The next development area I am going to discuss is graphical user interface (GUI), and in particular the different frameworks that can be adopted with my chosen programming language to develop an interface for the ‘MSc Properties’ staff to interact with the system I am going to produce during this project.

Below are some Java GUI frameworks I have come across during my research for this project:

* Abstract Window Toolkit (AWT) – Is Java’s original platform-dependent windowing, graphics and user-interface widget toolkit, and is part of the Java Foundation Classes (JFC) [].
* Swing – Is a GUI widget toolkit for Java, was developed to provide a more sophisticated set of GUI components than the earlier AWT, and is also part of the JFC [].
  + 1. **Testing Methodologies**

As previously explained, this project will centre around developing a system to tackle the major problem of data and document management and sharing, and whilst meeting these functionality requirements of ‘MSc Properties’, the system also needs to be maintainable, dependable and usable. This means that I will need to carry out testing to ensure that the system I have developed for ‘MSc Properties’ is compliant with the specified requirements and that the system has no faults or errors at runtime of the system, or the system is at an acceptable level for ‘MSc Properties’.

* + - 1. **Unit Testing**

The first testing principle I am going to discuss is the unit testing approach to testing, and will be fundamental to ensuring each of the elements of the system work independently of each other (where elements are not coupled).

The piece of literature I am going to review for unit testing is the article called *“A Simple and Practical Approach to Unit Testing: The JML and JUnit Way”* written by *Y. Cheon and G.T. Leavens*. This article looks at ways programmers can reduce the writing of labour-intensive code for unit testing, by writing formal specifications (for example, pre and post-conditions of methods).

The article goes on to explain that “writing formal specifications instead of test code makes the programmer’s task easier, because specifications are more concise and abstract than the equivalent test code, hence more readable and maintainable. Furthermore, by using specifications in testing, specification errors are quickly discovered, so the specifications are more likely to provide useful documentation and inputs to other tools”.

Below is some unit testing framework I have come across during my research for this project:

* JUnit Testing -
* Java Modelling Language (JML) -
  + - 1. **System Testing**

The next testing principle I am going to discuss is the system testing approach to testing, and will be fundamental to ensuring each of the elements of the system work together as they should.

The piece of literature I am going to review for system testing is the article called *“A UML-Based approach to System Testing”* written by *L. Briand and Y. Labiche.* This article looks at system test cases being derived from the analysis stage documents such as use case diagrams and also looks at the functional system test methodology called Testing Object-orienTed systEms with the unified Modelling language (TOTEM).

The article goes on to explain that “Deriving test requirements from early artefacts produced at the end of the analysis development stage, namely use case diagram, use case description, interaction diagram associated with each use case (sequence or collaboration), and class diagram (composed of application domain classes and their contracts). This early use of analysis artefacts is very important as it helps devising a system test plan, size the system test task, and plan appropriate resources early in the life cycle. Once the low level design is complete, when detailed information is available regarding both application domain and solution domain classes, then test requirements can be used to derive test cases, test oracles and test drivers”.

Below are some testing technologies I have come across during my research for this project that will assist in bug tracking:

* Bugzilla
* The Bug Genie
  1. Project Plan

As this is a large project, it is very important that I planned, monitored and managed the project smoothly from start to finish. I have used a Gantt chart, which provides a graphical illustration of the schedule of the project, broken down by project objectives, with completion dates for each objective, which has helped me track the activities in the project and make changes to work being carried out if necessary. This tool has been used to manage my time and allow me to stay on schedule best as possible, as there was a lot of tasks that needed to be completed in a limited time frame. This process of project planning is outlined in a software management article [5], where the article identifies “a recent update of the Chaos Report from the Standish Group, outlines a recipe for success that includes 10 items. The first three items are executive support, user involvement, and experienced project management.”, so project management is one of the 3 key factors to successful projects.

My project Gantt chart is below:

ADD GANTT CHART

I am now going to explain how the project objectives have been successfully completed by the project deadline date of 11 Jan 2015.

1. I wrote a project document outlining the details of the project, defining project objectives, scope, risks and approaches. I can constantly refer to this document to ensure the project progresses in the correct direction.
2. I wrote a work plan outlining the project objectives, with deadlines for each objective.
3. I defined relevant resources for the project, outlining decisions made on technology, equipment and software applications to use, ensuring that I have tested equipment and software applications, and am competent with the use of the selected technologies, prior to the start of the development.
4. I kept an eye on the project plan ensuring that objectives do not overrun past their completion date (where possible).
5. I stayed vigilant and alert to early warning signs of problems occurring in the project that may have resulted in the project being delayed and not meeting project deadlines.
6. I safeguarded against my project creeping outside of scope, so as new requirements were introduced during the development process, I had to ensure these are all still within available resources and overall aim and objectives of the project.
7. I managed risks as the project progressed, and as new risks were discovered, I had to evaluate them to ensure they do not cause a major problem to the project.
8. I tried to keep my project supervisor informed of any major problems occurring during the project, and did at times, seek advice where necessary, to resolve major problems as early as possible.
   1. Relevance to target award

Software Engineering is defined by Ian Sommerville as an engineering discipline concerned with all aspects of software production (specification, development, validation and evolution), and goes on to say it is concerned with the practicalities of developing and delivering useful software [1].

My project aim is to develop a distributed data and document management system, and to do this I had to explore the different software engineering techniques and decide which are best suited to tackling the software engineering task, and then develop and implement a piece of software that successfully meets the aim and objectives of the project.

This means the work I carried out during this project fits in with my target award MSc Computer Science (Software Engineering), because I applied software engineering models I have studied during my course such as agile to my software development. I also applied software engineering methodology I have studied during my course such as Inheritance and Encapsulation to my software development. I also applied the software engineering tools I have studied during my course such as unified modelling language (UML) to my software development. Lastly applied metrics such as cohesion, coupling, bugs etc. to my software development. By me exploring and applying these different software engineering techniques it allowed me to deliver useful software to ‘MSc Properties’ which in essence is Software Engineering.

* 1. Required Resources and Skills
     1. **Hardware**
* Operating System – Windows, Solaris, Linux or OS X;
* Processor – Intel® Core™ i5-4288U CPU @ 2.60GHz (or similar);
* Memory – 8.00 GB (or similar)
  + 1. **Software**
* Platform – Windows XP or higher (or similar);
* A JDK for Java 5 or later
* An Integrated Development Environment (NetBeans or similar)
* A concurrent version system (Git or similar)
* A bug tracking and testing tool (Bugzilla or similar)
  + 1. **Access**

I will require access to the following:

* MySQL database
  + 1. **Skills**
* Research skills
* Project management skills
* Report writing skills
* Ability to use Unified Modelling Language to model the distributed system
* Ability to write code in Java, HTML, Java Script, PHP and SQL.
* Ability to implement design patterns such as Observer
* Ability to use frameworks and API’s such as Spring and JRC respectively

I met these project resource and skill requirements, by ensuring I had the required hardware in place before development work began, I will then download the required software resources and test these to ensure they work appropriately. Once I have carried out a literature review of the required skills, methods and methodologies I can employ to meet the project aims and objectives, I will then undertake exercises to ensure that I have understood these methods and methodologies before development work begins and if any problems arise I will seek assistance from my project supervisor to overcome these issues.

* 1. Project Deliverables

NEED TO DO PROJECT DELIVERABLES

* 1. Ethics Approval

Ethics Approval is when a committee of University of Hertfordshire staff approve “any student undertaking a study involving the use of human participants which is undertaken as part of a programme of work for which the University of Hertfordshire is responsible for” [25].

My project will not require ethics approval because I did not undertake research that involved collecting data from human participants, and although my system does store business data which includes personal information, I used dummy information which replicates the personal information throughout the development.

1. Design
   1. Introduction

I am now going to discuss the design decisions I made during this project, and how I came to make these decisions.

* 1. Software Lifecycle

Considering the project aim and objectives, and also the project background, I believe an agile method is the best software process model to choose from, this is because in a fast moving business environment, software needs to be ready and available as quick as possible, and as original software requirements can quickly become out of date, it makes software developed useless very quick.

This means the software process model chosen needed to provide rapid development and delivery of software, and with the conventional plan driven software process models it can be difficult to do this because of the amount of documentation that needs to be created and signed off, and the lack of interleaving development stages makes it difficult to cope with quickly evolving requirements.

Also as there will be an office manager from ‘MSc Properties’ working on the project assisting the development, it would make sense in being able to deliver software quickly to allow for this to be evaluated and confirm the project is moving in the correct direction at each iteration of the agile lifecycle, as it can be difficult to gather exact system requirements from clients without going over the development process and having something to evaluate and add to or remove from to develop a system that successfully meets the project aim and objectives.

The agile method I have chosen is METHOD, and although I decided to use the agile method METHOD, as explained previously in the project background, a business analysis and system requirements exercise was carried out with an officer from ‘MSc Properties’, which allowed me to have a fairly strong idea of what functionality ‘MSc Properties’ required from the system to be developed.

The information gathered put me in a position where I was able to carry out a fair amount of design work for the system prior to the first development iteration, however I still used the divide and conquer technique which allowed me to break the development into smaller pieces, and accomplish one or a number of the smaller problems with each iteration of the development process. The breaking down of the development into smaller pieces enabled me to tackle each smaller task on its own and then combine the solutions to the smaller problems to provide a solution to the original problem, which meant providing a solution to the original problem was easier and more manageable.

* 1. Modelling System Behaviour

I used a number of design techniques to assist in the modelling of the system behaviour and am now going to explain what design techniques I used and why I decided to use these for the development.

However due to the size of the project and due to a level of resources available to the project, I decided that for me to successfully achieve as much of the project aim and objectives as possible, I would not be able to develop diagrams to model the entire system behaviour, and instead I selected a sub set of the system functionality to model.

* + 1. Use Case Diagrams

For this development I decided to develop a number of use cases which have been documented under Appendices A. By developing use case diagrams, this enabled me to identify the relationships between actors (roles within the system, for example a user of the ‘MSc Properties’ system, or the database which will hold all the system information) and use cases (functions within the system, for example creating a property).

The use case diagrams developed during the project were not only used to model the system behaviour, but I also used them to create test scripts for the system testing, as the use case diagrams outlined the different functions that should occur within the system, and therefore can be used to carry out the black-box system testing, which will be explained in further detail in the implementation section of this report.

* + 1. Data Flow Diagrams
    2. Class Diagram
    3. Enhanced Entity Relationship Diagram
    4. Sequence Diagrams
    5. Storyboard

1. Implementation
   1. Introduction
   2. Testing

Although