

**UNIVERSITY OF SURREY**

**Faculty of Engineering and Physical Sciences**

**Department of Computing**

BSc Undergraduate Programmes in Computing

Module COM3001: 45 credits

**COM3001: Professional Project**

2014-15

FHEQ Level 6 (Year 3)

Student: **Chara Katiri**

Email: [ck00113@surrey.ac.uk](mailto:ck00113@surrey.ac.uk)

URN: 6166668

**Document Control**

| Version History | | | |
| --- | --- | --- | --- |
| Version | Date | Reason for issue / Change Request | Issued By |
| 0.1 | 29.10.2014 | Document setup & contents creation | CK |
| 0.2 | 30.10.2014 – 10.11.2014 | Requirements gathering | CK |
| 0.3 | 13.02.2015 | Draft table of contents | CK |
| 0.4 | 16.2.2015 -21.02.2015 | Update of table of contents and creation of introduction, literature review and system analysis. | CK |
| 0.5 | 22.02.2015 | Creation of system design. | CK |

| Document Reviewers | | | |
| --- | --- | --- | --- |
| Action | Name | Role | Date Comments Received |
| Reviewer | Dr Steve Wesemeyer | Supervisor |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

| Document Approval | | | | |
| --- | --- | --- | --- | --- |
| Action | Name | Role | Signature | Date |
| Approver | Dr Steve Wesemeyer | Supervisor |  |  |
| Approver | Dr Anna Lisa Ferrara | Examiner |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |
| --- | --- |
| Project Owners | |
| Role | Name |
| Student | Chara Katiri |
| Supervisor | Dr Steve Wesemeyer |
|  |  |

|  |  |  |
| --- | --- | --- |
| Contact Details | | |
| All queries in relation to this document should be addressed in the first instance to: | Name | Chara Katiri |
| Department | Computing, University of Surrey |
| e-mail | ck00113@surrey.ac.uk |
| Address | University of Surrey, Guildford, Surrey, GU2 7XH |

|  |  |  |  |
| --- | --- | --- | --- |
| Document Name: COM3001 Professional Project. Project Requirements Document | Security Level: Private | Version Date: 04/11/2014 | Version: 0.4 |

**If printed this is an uncontrolled document**

**Acknowledgements**

**Table of contents**

1. Introduction 6

1.1 Motivation 6

1.2 Project idea and description 6

1.3 Project objectives and goal statement 6

1.4 Project benefits 6

1.5 ROI statement 6

1.6 Project stages 6

2. Literature review 6

2.1 Background research 6

2.2 Tackling the problem 6

2.3 Usability and acceptance concerns 6

3. System analysis 6

3.1 Current process 6

3.2 System requirements and analysis 6

3.2.1 Functional and non-functional requirements 6

3.2.2 Feasibility analysis 6

3.3 Summary of deliverables 6

3.4 Stakeholders 6

4. System design 6

4.1 Design method 6

4.2 Design plan 6

4.3 Technology options 6

4.4 Benefits of Spring MVC framework 6

4.5 Features of Spring MVC framework 6

4.6 Architecture and programming language 7

4.7 Operating system 7

4.8 Development platform 7

4.9 Database (SSH, selection, creation) 7

4.10 Security levels 7

4.11 System Challenges 7

4.12 System risks and issues 7

4.12.1 System risks 7

4.12.2 System issues 7

4.13 System constraints, dependencies and timescales 7

4.13.1 System constraints 7

4.13.2 System dependencies 7

4.13.3 System timescales 7

5. System implementation 7

5.1 Implementation plan 7

5.2 Build tools and libraries 7

5.3 Database (remote connection) 7

5.4 Integration (database connectivity, university database) 7

5.5 Success criteria 7

6. Diagrams 7

6.1 Class diagram 7

6.2 Sequence diagram 7

6.2.1 Login 7

6.2.2 Add item 7

6.2.3 Delete item 7

7. Test and evaluation 7

7.1 Requirements testing 7

7.2 Security testing 7

7.3 UAT testing 7

8. Conclusion 8

8.1 Achievements, review of objectives 8

8.2 Issues and shortcomings 8

8.3 Future work (system as application) 8

8.4 Final evaluation and conclusions 8

8.5 Summary 8

9. References 8

##### Acknowledgements

##### Table of contents

##### Table of figures

##### Acronyms

# Introduction

## Motivation

## Project idea and description

## Project objectives and goal statement

## Project benefits

## ROI statement

## Project stages

# Literature review

## Background research

## Tackling the problem

## Usability and acceptance concerns

# System analysis

## Current process

## System requirements and analysis

### Functional and non-functional requirements

### Feasibility analysis

## Summary of deliverables

## Stakeholders

# System design

## Design method

## Design plan

## Technology options

## Benefits of Spring MVC framework

## Features of Spring MVC framework

## Architecture and programming language

## Operating system

## Development platform

## Database (SSH, selection, creation)

## Security levels

## System Challenges

## System risks and issues

### System risks

### System issues

## System constraints, dependencies and timescales

### System constraints

### System dependencies

### System timescales

# System implementation

## Implementation plan

## Build tools and libraries

## Database (remote connection)

## Integration (database connectivity, university database)

## Success criteria

# Diagrams

## Class diagram

## Sequence diagram

### Login

### Add item

### Delete item

# Test and evaluation

## Requirements testing

## Security testing

## UAT testing

# Conclusion

## Achievements, review of objectives

## Issues and shortcomings

## Future work (system as application)

## Final evaluation and conclusions

## Summary

# References

**Table of figures**

Figure 1: Overview of the Spring Framework 20

Figure 2: Request processing workflow in Spring MVC 21

Figure 3: Context hierarchy in Spring Web MVC 22

**Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| AD | Active Directory |
| BRD | Business Requirement Document |
| HTTPS | Hyper Text Protocol Secure |
| IST | Integrated Systems Testing |
| MoSCow | Must or Should Could or Won’t |
| MVC | Model View Controller |
| RAD | Rapid Application Development |
| SDLC | Systems Development Life-Cycle |
| UAT | User Acceptance Testing |
| UBE | Unsolicited Bulk Email |

1. **Introduction**

The purpose of this document is the detailed description of the project’s plan to design, develop and deliver the Pinboard website as a method to minimize the number of junk emails sent and received daily in the University of Surrey. The project was developed as part of ‘COM3001 Professional Project’ module, taught by the University of Surrey, Department of Computing.

* 1. **Motivation**

Nowadays, electronic mail is the predominant digital communication platform. According to Reuter’s survey [1], 85% of human population is connected online and communicate through email despite the number of social channels available. The Internet is overflowing with 100 million emails send and received daily [2], of which 97.4 millions are spam/junk [3].

The question is what leads to the creation of so many “junk emails” within the university environment.

* 1. **Project idea and description**

The lack of an organised way to advertise second hand books or requests for house sharing, leads the students to the creation and distribution of unsolicited bulk email (UBE) emails. These emails are categorised as junk emails.

The creation of a Pinboard solution in the form of website is developed as a method to minimize the number of “junk emails” send and received daily within the University of Surrey. Junk emails have a negative impact on the size of users’ mailbox and the management of incoming emails.

* 1. **Project objectives and goal statement**

The objective of the project was the creation of a dynamic website as a university-specific outlet. Pinboard website offers an organised way for students to sell second hand books. It also provides functionalities that allow students to advertise available rooms and make house hunting more efficient.

The use of the services offered by the website have the potential to significantly reduce the number of “junk” emails sent daily send between students regarding second-hand books and room swaps. Additional functionalities can enrich the search user experience by offering a number of browsing categories like unwanted tickets for events, sports equipment, electronics etc.

The objective for personal development was the successful completion of the full development lifecycle from general management of the project, solution design, development and delivery. Introduction and hands on experience using Spring MVC Framework technologies was another important personal objective.

* 1. **Project benefits, ROI statement**

The project benefit was the introduction of a new communication channel that allows students to sell second-hand books and search for housemates/swap rooms on campus. Therefore, the use of the website can help minimize the creation of “junk emails” sent for the purpose of second-hand books and room swaps. The deployment of the website on the University’s intranet can introduce the first steps towards the replacement of the physical Pin-boards on campus, reduce paper copies and promote environmental friendly behaviour. The website can also help reduce network traffic and disk space used by “junk” emails and therefore minimize the storage cost.

* 1. **ROI statement**

The return on investment (ROI) is that the Pinboard solution will enhance the quality of student experience with the use of multimedia and make students’ requests/hunt for houses and books easier.

*Enable*:

* *Providing, a high quality solution to meet students’ needs* and support the University of Surrey strategy for continual improvement. The creation of Pinboard site can be seen as a new communication channel.

*Defend*

* *Improve student relationships by* providing a common platform as a mean to control and minimize the number of spam emails sent internally.

*Grow*

* *Taking a strategic approach by introducing* a new communication channel that allows the students to sell second-hand books and search for housemates/swap rooms on campus. Pinboard solution enhances the quality of student experience with the use of multimedia and makes their requests/hunt for houses and books more pleasant*.* 
  1. **Project stages**

The management of this project is based on the Systems Development Life-Cycle (SDLC). The first step was the investigation of an existing problem that can be improved.

* *Preliminary analysis and Idea*: To help minimise the creation of junk emails for second hand books and room swaps, the development of the Pinboard website was suggested.
* *Literature review:* Background research helped identify websites, which offer similar services and then create functional requirements that help measure Pinboard’s improvement over existing websites.
* *System analysis, requirements definition:* Functional and non-functional requirements were created for the creation of a Business Requirement Document (BRD).
* *System design:* To support the BRD, features of the website were detailed and screen layouts were created along with process diagrams.
* *System Development*: In preparation for the development of the website, a request was made to the IT Services team for access to a Tomcat server and a MySQL database backend. Following the creation of Pinboard’s home page, the login functionality for the user was created first. Appropriate measures were taken to ensure that input text submitted by users is secure and the website is protected from attacks.
* *Test and evaluation*: Following the completion development of code the system is tested to ensure that each requirement is met. Tests included functional requirements, security concerns and User Acceptance Testing (UAT) to ensure that early adopters are happy with the service and functionalities offered by the website.

1. **Literature review**
   1. **Background research**

Background research took place to identify other solutions that offer similar functionalities and understand how the website was developed. Identification of their strengths and weakness helped in the decision of the must-have features of the Pinboard website. This includes the target audience and functionalities that the users would like to see on the website. Comparison between a number of open source e-commence solutions such as Gumtree, eBay, Etsy and Amazon was made. The strengths and weakness of the services are shown below:

|  |  |
| --- | --- |
| **Ope source e-commerce solutions like Gumtree, eBay, Esty, Amazon.** | **Pinboard** |
| Open and available to everyone. | Targets students and is available only to the University of Surrey students and stuff. |
| Basket functionality. | Bookmark functionality. |
| Integration with Pay-Pal | Students are located on campus and payments are made in person. |
| Delivery services. | Students are located on campus and deliveries are be made in person (meeting between student seller and student buyer). |

* 1. **Tackling the problem**

The development of Pinboard website will provide alternative processes to email distribution for students that wish to sell their second-hand books or request room swaps on campus. By providing an organised system, the target audience can navigate to Pinboard website and check numerous options before they make their decision on which book to purchase/room to swap. The organised service helps minimize the number of bulk emails created by student-sellers that wished to advertise books or room swaps.

* 1. **Usability and acceptance concerns**

The Systems Team might not accept the deployment of the application on Surrey’s intranet. In this case, the project will be seen as practice and an opportunity to manage and complete the lifecycle for a website.

1. **System analysis**
   1. **Current process**

Currently, there’s no organised process for students that wish to advertise second hand books or request room swaps on campus. As a result, the sellers/seekers create and distribute bulk emails (known as UBE) hoping that someone will get back to them. This process generates numerous unwanted emails for students that are not interested, and it’s possible that overloads the network as well as the amount of storage on servers.

* 1. **System requirements and analysis**

The requirements listed below will be used as the basis for the development and testing of the web application, to determine the completion of the project and access its success.

* Requirement Type: Functional or Non-Functional.
* Functional: Requirements are classified as Functional if their aim is to provide information on what the system should do. [6] Examples include: authentication and authorization processes, historical data, audit tracking, legal and regulatory requirements etc. [7]
* Non-Functional: Requirements are classified as non-functional if they cover constraints of the solution, targets and ways to control its mechanisms. Examples include performance, scalability, maintainability etc. [7]
* MoSCoW: ‘Must or Should Could or Won’t’ business analysis and prioritization method. The prioritization shows the importance of each requirement however it does not mean that requirements classified as Musts will be developed first; it means that by the completion of the project they must be delivered. Requirements classified, as ‘Won’t’ should be listed even if they won’t be met during this sprint. The list of ‘Won’t’ requirements can be suggestions for future development and improvements on the system.
  + 1. **Functional and non-functional requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. no.** | **Requirements Description** | **Req. Type** | **MoSCoW** |
| 1 | The solution must automate and simplify the process of selling and buying goods, such as second-hand books, within the University of Surrey environment. | Non-functional | M |
| **Database** | | | |
| 2 | The centralized database that contains the products must update automatically and daily. | Non-functional | M |
| 3 | The solution must update the list of products so that a product becomes unavailable after a user purchases it. | Functional | M |
| **Seller/Buyer requirements** | | | |
| 4 | The solution must ask the users (byers & sellers) to agree with ‘The Sales of Goods Act 1979’ and adhere to it: “*"traders must sell goods that are as described and of satisfactory quality”* [4], [5]. The Terms & conditions must be accessible, meaningful and fair. | Non-functional | M |
| 5 | The solution won’t require the users to register. The login process will be based on the existing Active Directory (AD) groups. New users will be automatically added in the database. | Functional | W |
| 6 | The solution must validate the user information upon registration e.g. text fields that require information and valid email that contains ‘@’, ‘.’. | Functional | M |
| 7 | The solution should make use of cookies to enable fast and effective login. | Functional | S |
| 8 | The solution must allow the users to search products based on categories e.g. books or room swaps. | Functional | M |
| 9 | The solution must provide a search functionality (text-box search) to allow the users enter keywords. | Functional | M |
| 10 | The solution should enable the users to sort the products based on price (Low to High, or High to Low). | Functional | S |
| 11 | The solution must provide information about the product: Title, Description, Price, Category and Seller contact information. | Functional | M |
| 12 | The solution must ask the seller to fill specific fields regarding the product: Title, Description, Price, Category and Seller contact information such as email address and/or mobile phone number. | Functional | M |
| 13 | The solution should provide a ‘My profile’ area for each user. | Functional | S |
| 14 | The solution should allow users to bookmark items. | Functional | C |
| 15 | The bookmark functionality must protect users’ bookmarks and follow data privacy policies. | Non-functional | M |
| 16 | The solution won’t allow items to be purchased online (PayPal services) provided that the users are students that live on campus and they’ll meet for the delivery and payment of the product. | Functional | W |
| 17 | The solution won’t provide delivery charges, stages involved in ordering process and dispatch information. | Functional | W |
| 18 | The solution won’t provide refund functionalities. | Functional | W |

* + 1. **Feasibility analysis**

Feasibility study [9] is conducted as part of the assessment process. The analysis of the project helped identify problems and outline alternative options. This helped identify issues and potential risks in the early stages of the software development system. For example there was a risk that the Service Team might not accept the deployment of the website to the University of Surrey intranet. After the discussion of the project with the Service Team, the feasibility of the project was also discussed with the project supervisor and the decision made was to go ahead.

Even though the website cannot be deployed on the university’s intranet, it’s an opportunity for a final year student to manage a project and go though the full systems development lifecycle. It was also an opportunity to achieve a challenge and develop a project based on a new Framework without previous experience, Spring MVC.

* 1. **Summary of deliverables**

|  |  |
| --- | --- |
| **Deliverable name** | **Description** |
| Project Design | The solution structure and the plan to implement functional and non-functional requirements in order to meet the project objectives |
| Definition of functionalities and services | The services delivered by the project should be clearly defined for the benefit of  i) the customer, to describe how navigate thought and use the website  ii) the IT team that is responsible for the maintenance of the website. |
| Specification of Roles and responsibilities | Clear specification of roles and responsibilities of:  i) the user and the policies that must adhere  ii) the IT Team that is responsible for the maintenance of the website. |
| Project deployment | Details on the steps to be followed as part of the deployment process. |

* 1. **Stakeholders**
* Stakeholders [8] are individuals or organizations with interest in a project. Typically they are internal or external investors, employees, customers and suppliers. Stakeholders include the target audience, which is a subset of students, sellers, of second hand books and seekers of room swaps on campus.

|  |  |
| --- | --- |
| **Stakeholder** | **Support Role** |
| A subset of University of Surrey students /and academics | A group of volunteers that wish to test the application and are interested in the functionalities that the application offers. |
| University of Surrey IT and Service Desk | The support team. |
| Chara Katiri | Computer Science Student, the developer. |
| Dr Steve Wesemeyer | Professional project supervisor and coordinator.. |

1. **System design** 
   1. **Design method**

The chosen development approach is Waterfall (over Agile) in combination with Rapid Application Development (RAD). In the project environment, Waterfall methodology can be easily understood not only by the developer but also the client. Waterfall is easier to manage due to its structure and the way major tasks can be divided in achievable chunks thought numerous phases.

RAD methodology enables the developer to share working software and visual prototypes with the client in early stages of the development cycle. Therefore, the stakeholders engage with the project and can provide useful feedback to the developer.

* 1. **Design plan**

The modular design of the website ensures flexible maintenance and allows for the website to be expanded later in time if required. The RAD methodology allows the developer to offer iterative and incremental delivery of the project, collaboration with the stakeholders and future improvements of the core if necessary. The Waterfall development cycle is described as follows:

1. Planning
2. Requirements analysis
3. Design
4. Development
5. Testing
6. Implementation
7. Maintenance

The main resource that the development plan was based on is the final year student - developer (CK) and the supervisor (SW) that supports and guides the project.

* 1. **Technology options**

The client and web server web application is developed using Java and the Spring MVC Framework. *Spring Framework* is an open source, Aspect Oriented Programming (AOP) framework for Java based applications. *Spring MVC Framework* is an extensible MVC framework and a component of Spring that is used for creating web applications. It’s developed on core Spring functionality that provides technologies for views (JSP) [10].



Figure 1: Overview of the Spring Framework

* 1. **Benefits of Spring MVC framework**

Spring MVC Framework is chosen over other technologies, such as PHP, due to the support features it offers. It’s a popular framework used by a lot of web developers because it offers quality from design to implementation and it follows best practices. The creation of specialised objects like Model View Controller (MVC), DispatcherServlet and handler mapping result in clear separation of roles. The use of existing business objects allows reusability of code and escapes duplication. Therefore system flexibility is increased and maintainability is improved [10].

* 1. **Features of Spring MVC framework**
* *DispatcherServlet*

Spring MVC is a request-driven framework and it makes use of a central Servlet to dispatch requests to controllers [10]. DispatcherServlet is configured in XML and acts as the front controller. The Front controller receives the incoming request from the client and then delegates the request to a Controller. The Controller receives and handles the request, performs business logic and creates a new model. In order to delegate rendering of response, the Controller forwards the model to the Front Controller. The Front Controller selects a View and forwards the model in order to create a render response. Finally the View returns control to the Front Controller and the response is sent to the client.



Figure 2: Request processing workflow in Spring MVC

* *Model, View, Controller (MVC):*

Model View Controller is a well-established software architectural pattern. It’s used in web application development in order to separate the business logic from the user interface [11]. The MVC consists of three modules: Model, View and Controller. Controller is the front servlet and acts as an interface between the View and Model. The client sends requests to the Controller. The Controller then forwards the input to JSP pages. In Spring Web MVC terminology, JSP pages are named Views and are managed by the Controller. In response, the View (JSP) calls the Model. The Model (JavaBean) represents the state (data) and business logic of the application. The Model connects to the database to retrieves/saves data. The View (JSP) generates the response and sends it back to the controller to return it to the client. Note that in complex applications the Controller also manages the Service entity whenever it needs to perform business logic. This helps improve maintainability, as the business logic is not contained directly in the controller [12].

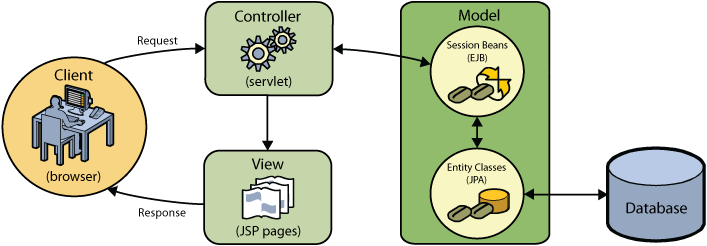


Figure 3: Context hierarchy in Spring Web MVC

* 1. **Architecture and programming languages**

The system architecture is separated in several layers to cater the needs of the website.

* *Presentation layer:* View (JSP pages), presentation of data (HTML5, CSS)
* *Security layer:* Authentication and authorisation in login using the University of Surrey email address to restrict access only to students and staff of the university. Security is also provided by Spring MVC framework. Use of Hyper Text Protocol Secure (HTTPS) as a communication protocol for secure communication over the network.
* *Business logic:* Spring MVC framework controllers and services.
* *Database layer:* Web server, Application server, MySQL.

* 1. Operating system
  2. Development platform
  3. Database (SSH, selection, creation)
  4. Security levels
* *Security defence – SQL injection*:

Students are required to login in order to view the list of available items or add items as sellers. There’s no need for the users to register as they’re using their University email account. During the login process the students provide their username (email) and password. The submission of their personal information generates an SQL query [13]. The query is then executed on the database and the user account is authenticated and authorised to allow access to the website. Text submitted by the user should always be treaded as suspicious thus not only client side but also server side validation should be applied. Lack of validation will allow cybercriminals to submit specifically crafted SQL commands (known as SQL injection) aiming to attack the application and reveal information about the database’s structure such as numbers and names of tables. The knowledge of such information will allow them to manipulate the tables, create admin accounts to manage the database and access users’ private information.

In order to defend such operations from attackers, the SQL query is parameterised to prevent execution of malicious code. This secures the database from common attacks such as the ‘Little Bobby tables’.

* *Form validation:*

The jQuery validator [14] plugin is used to validate the Input submitted by users through the login form. The customisation options offered by the plugin provide easy validation of input content. These include ready-made validation methods such as email validation and delivery of prepared error messages.

Validation is applied both on client and server side. Client side validation offers a smooth user experience and it helps reduce the load on server by identifying user errors. Server side validation checks the data submitted by the user to prevent SQL injection attacks of cybercriminals attempts to submit untrustworthy data.

* 1. **System Challenges**

A number of features make the development of the website a challenging task.

* *Use of Spring Web MVC Framework:* The use of Spring MVC is a steep learning curve.
* *Users user’s profile:* The user’s profile must be kept secure to ensure that no other users can gain access to it and modify sensitive information.
* *Items list:* The list of items must be updated automatically so that new items are shown as available. The list of items should also be updated to exclude the latest purchased/sold items.
* *The nature of products:* Each product is unique (quantity=1)
* *Deployment of the website on the University of Surrey intranet:* If the solution is not accepted by the Service Team/IT Services Team the web application cannot be tested by volunteers.
  1. **System risks and issues**
     1. **System risks**
* If the deployment of the website on the University of Surrey intranet is not approved, the deployment to early adopters cannot be completed and neither feedback can be collected.
* UAT depends on testers being available when required (CS students, volunteers).
* End user expectations unrealistic compared to solution.

*Contingency plan to mitigate risks:*

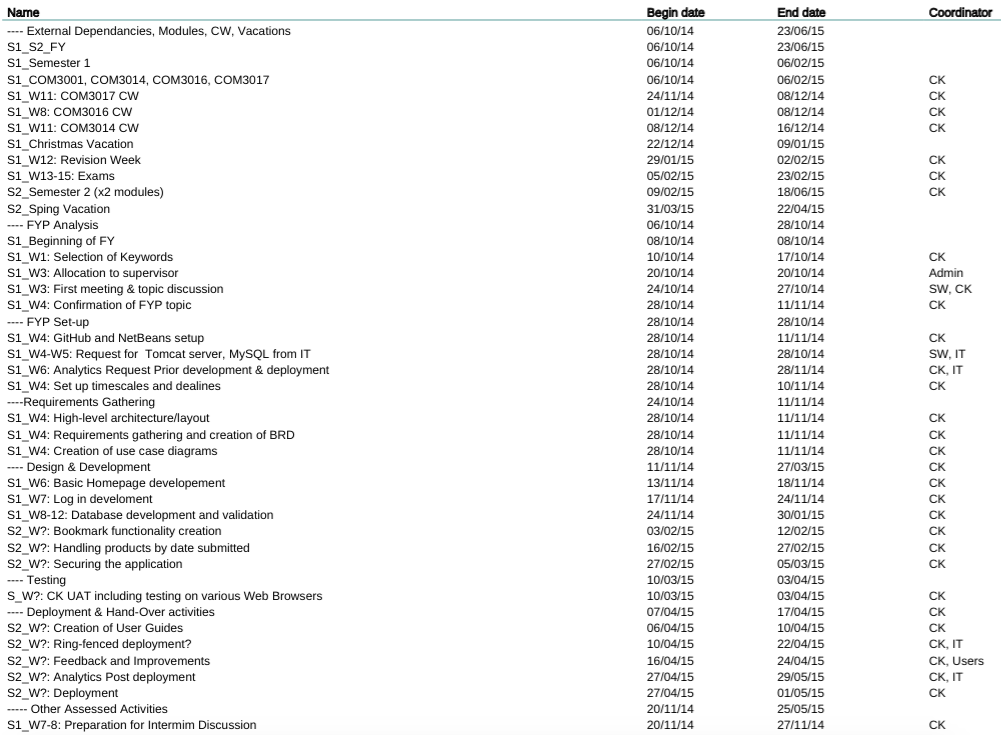
* The development of the website is based on Rapid Application Development (RAD) which allows visual prototypes to be shown to the stakeholders. This keeps stakeholders engaged and feedback can be received in early stages of the development lifecycle.

* + 1. **System issues**
* One of the initial aims of the system was to collect the number of the junk emails sent within the university. Then analyze the number of emails sent before and after the deployment of the website to measure the efficiency of use of the website. A request was submitted to the IT services team to extract the average number of junk emails exchanged on Surrey’s network. In response they explained that there is no way to pull out the junk emails as all Surrey accounts are automatically listed as ‘safe senders’ and they can’t be filtered as junk unless those emails are marked as junk by the receivers.
  1. **System constraints, dependencies and timescales** 
     1. **System constraints**
* Time constraints due to 5 other modules that run in parallel with COM3001. All the modules are assessed based on at least one coursework and an exam.
* Time scales for set up, development and deployment of the website are aggressive in order to meet the demands of COM3001 Professional Project deadlines and deliveries.

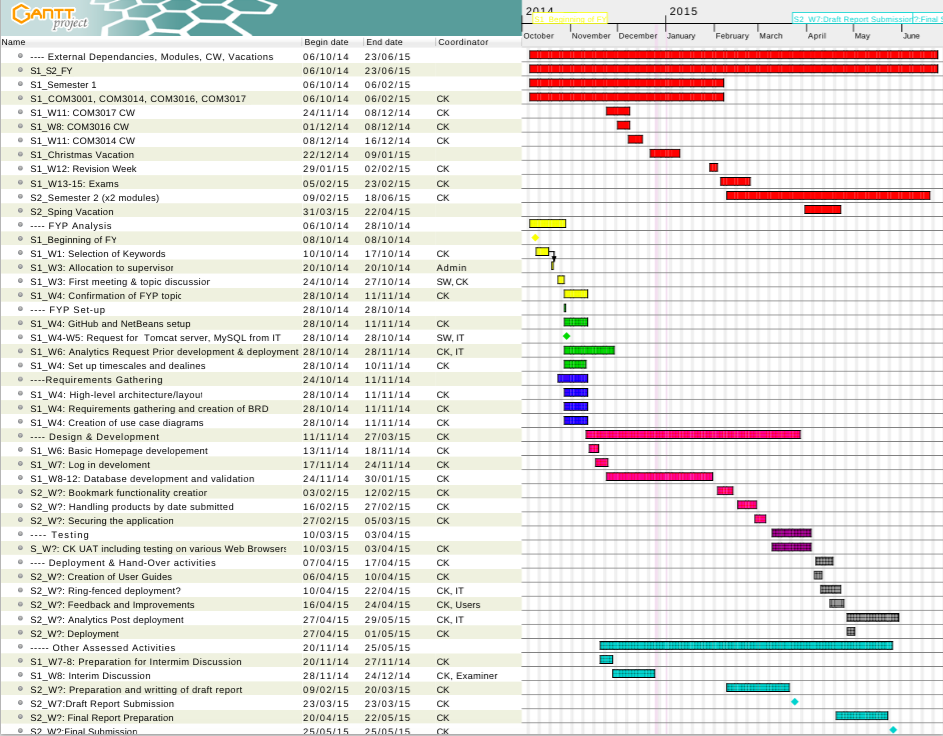
*Contingency plan to mitigate risks:*

* Timescales were set up with the work required by other modules in mind.
* If for any reason the timescales shift, the requirements categorised as ‘Could’ will not be implemented.
  + 1. System dependencies
    2. **System timescales**

The optimal timescales for system’s development is 25% research and design, 45% development and 30% testing.



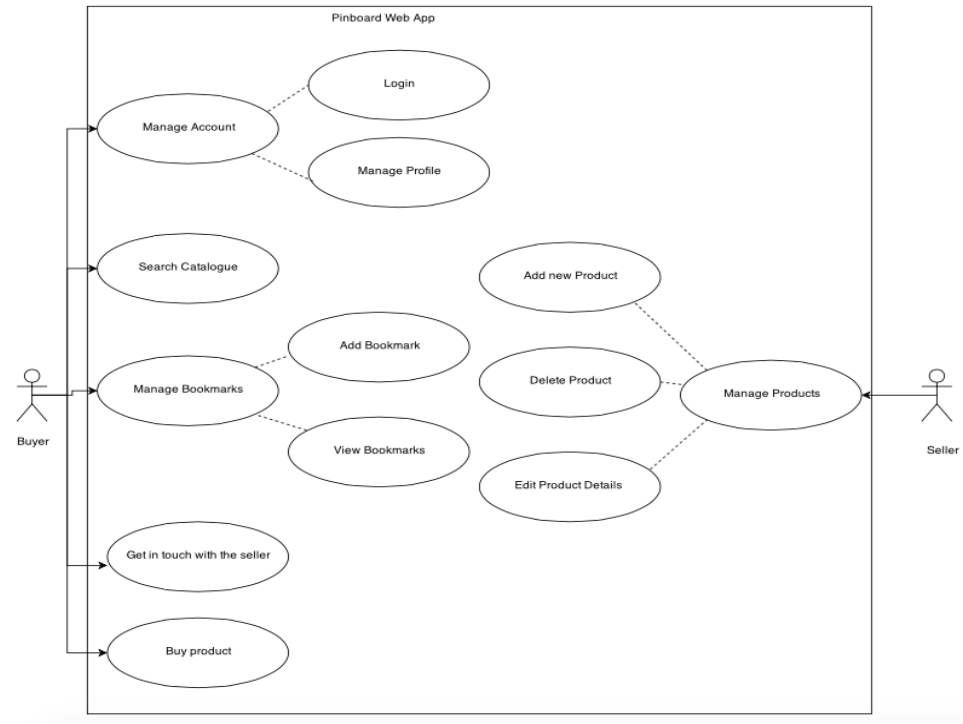




1. System implementation
   1. Implementation plan
   2. Build tools and libraries
   3. Database (remote connection)
   4. Integration (database connectivity, university database)
   5. **Success criteria**

* The deliverables and software development requirements met.
* The objectives are met.
* The website functions.
* Delivery of final project in agreed timescale.
* System testing (requirements and UAT) is fully documented.

1. **Diagrams**
   1. Class diagram
   2. **Sequence diagram**



* + 1. Login
    2. Add item
    3. Delete item

1. Test and evaluation
   1. Requirements testing
   2. Security testing
   3. UAT testing
2. Conclusion
   1. Achievements, review of objectives
   2. Issues and shortcomings
   3. Future work (system as application)
   4. Final evaluation and conclusions
   5. Summary
3. **References**

[1] Reuters, 'Most of world interconnected through email and social media', 2015. [Online]. Available: http://www.reuters.com/article/2012/03/27/uk-socialmedia-online-poll-idUSLNE82Q02120120327. [Accessed: 21- Feb- 2015]. [Accessed 03/11/2014].

[2] The Radical Group, ‘Most of world interconnected through email and social media’, 2015. [Online]. Available: http://www.radicati.com/wp/wp-content/uploads/2013/04/Email-Statistics-Report-2013-2017-Executive-Summary.pdf. [Accessed: 21- Feb- 2015]. [Accessed: 03/11/2014]

[3] Esecurityplanet.com, 'Almost 100 Billion Spam E-mails Sent Daily in Q1 2013 - eSecurity Planet', 2015. [Online]. Available: http://www.esecurityplanet.com/network-security/almost-100-billion-spam-e-mails-sent-daily-in-q1-2013.html. [Accessed 3 Nov. 2014].

### [4] Legislation.gov.uk, 'Sale of Goods Act 1979', 2015. [Online]. Available: http://www.legislation.gov.uk/ukpga/1979/54. [Accessed 10 Nov. 2014].

[5] Glassraven.com, 'Trading Online | Legal Requirements | Ecommerce Websites | Glassraven', 2015. [Online]. Available: http://www.glassraven.com/articles/online-legal-requirements.php. [Accessed: 21- Feb- 2015]. [Accessed 10 Nov. 2014].

[6] Sqa.org.uk, 'Functional and Non-Functional Requirements', 2015. [Online]. Available: http://www.sqa.org.uk/e-learning/SDM03CD/page\_02.htm. [Accessed 10 Nov. 2014].

[7] ReQtest, 'Functional Requirements vs Non Functional Requirements', 2015. [Online]. Available: http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/. [Accessed 10 Nov. 2014].

[8] Dictionary.com, 'the definition of stakeholder', 2015. [Online]. Available: http://dictionary.reference.com/browse/stakeholder. [Accessed 9 Nov. 2014].

[9] Extension.iastate.edu, 'What is a Feasibility Study? | Ag Decision Maker', 2015. [Online]. Available: http://www.extension.iastate.edu/agdm/wholefarm/html/c5-65.html. [Accessed: 21- Feb- 2015].

[10] Spring.io, (2014). spring.io. [online] Available at: <http://spring.io/> [Accessed: 21- Feb- 2015].

[11] T. Coplien, 'The DCI Architecture: A New Vision of Object-Oriented Programming', Artima.com, 2015. [Online]. Available: http://www.artima.com/articles/dci\_vision.html. [Accessed: 22- Feb- 2015].

[12] D. Vrusias, 'COM3014: Advanced Challenges in Web Technologies', 2012. [Online]. Available: https://sharepoint.surrey.ac.uk/exampapers/Papers/Advanced%20Challenges%20in%20Web%20Technologies.pdf. [Accessed: 22- Feb- 2015].

[13] Beyondsecurity.com, (2014). SQL Injection. [online] Available at: http://www.beyondsecurity.com/about-sql-injection.html [Accessed: 22- Feb- 2015].

[14] Jqueryvalidation.org, (2014). jQuery Validation Plugin | Form validation with jQuery. [online] Available at:<http://jqueryvalidation.org/> [Accessed: 22- Feb- 2015].