2.2. Tackling the problem

The development of Pinboard website provides an alternative process 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.

2.3. Acceptance and usability concerns

The key concern is for the IT Services/Systems Team to accept the deployment of the website on Surrey's intranet. If this is resolved, the arising concern is the level of usability of the website by the end users. However, if the deployment of the website is not approved, the project will be seen as an opportunity for hand-on experience, management and completion of the development lifecycle for a website.

2.4. Research on technology

The client and web server website 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 the creation of web applications. It's developed on core Spring functionality that provides technologies for views (JSP) [10].

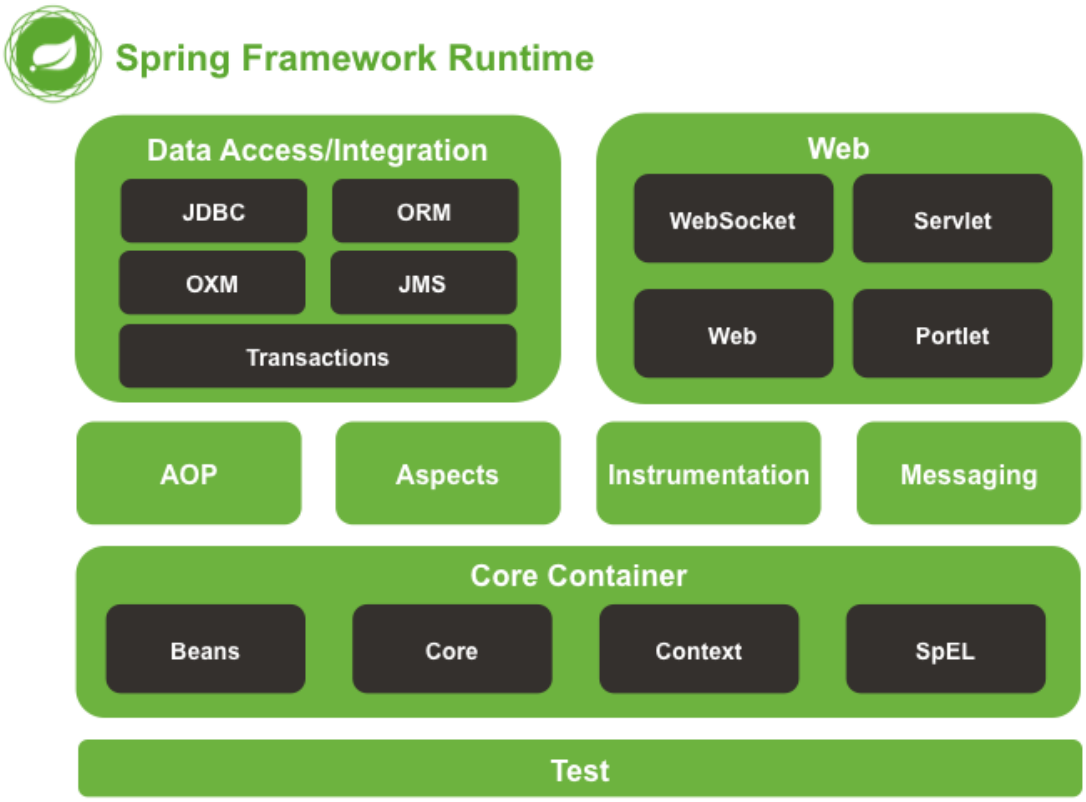

Figure 1: Overview of the Spring Framework

The Spring Framework consists of a number of modules: Data access/integration; Web; AOP; Aspects; Instrumentation, Messaging; the Core container and finally the Test module.

In Data access/integration module provides support to: (i) JDBC, (ii) ORM, (iii) OXM, (iv) JMS and (v) Transaction module.

(i) The Java DataBase Connectivity technology (JDBC) defines how a client accesses the database and provides methods to query and update the database.

(ii) The Object Relational Mapping (ORM) converts data between incompatible types and systems. In particular, this technique is beneficial in modern systems where systems include a number of subsystems.

(iii) The Object/XML mapping (OXM) converts objects to XML and vice-versa.

(iv) The Java Message Service (JMS) sends messages between two or more clients and enables reliable connection between different components.

(v) The Transaction module coordinates the transactions for management APIs and Java objects.

The Aspect Oriented Programming (AOP) manages the application challenges, such as security, logging and management, into aspects and helps create modules that enable reusability. AOP complements OOP (Object Oriented Programming) by creating a new program structure. The difference between OOP and AOP is that is OOP the key element (unit of modularity) is the class where in AOP the key element is the aspect.

The Core container is Spring's key module and it's used to provide BeanFactory and ApplicationContext containers.

The purpose of the Test module is to support the services for unit and integration testing.

2.4.1.Benefits of Spring MVC framework

Spring MVC Framework was chosen over other MVC technologies, such as Struts, due to the support features it offers. Struts is a sophisticated framework, strictly focused on presentation. On the other hand, Spring is a "*Lightweight inversion of control and Aspect Oriented Container Framework*" [10]. It's a popular framework used by a lot of web developers because it offers quality from design to implementation, promotes best practices and it's adaptable. 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].

2.4.2.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.

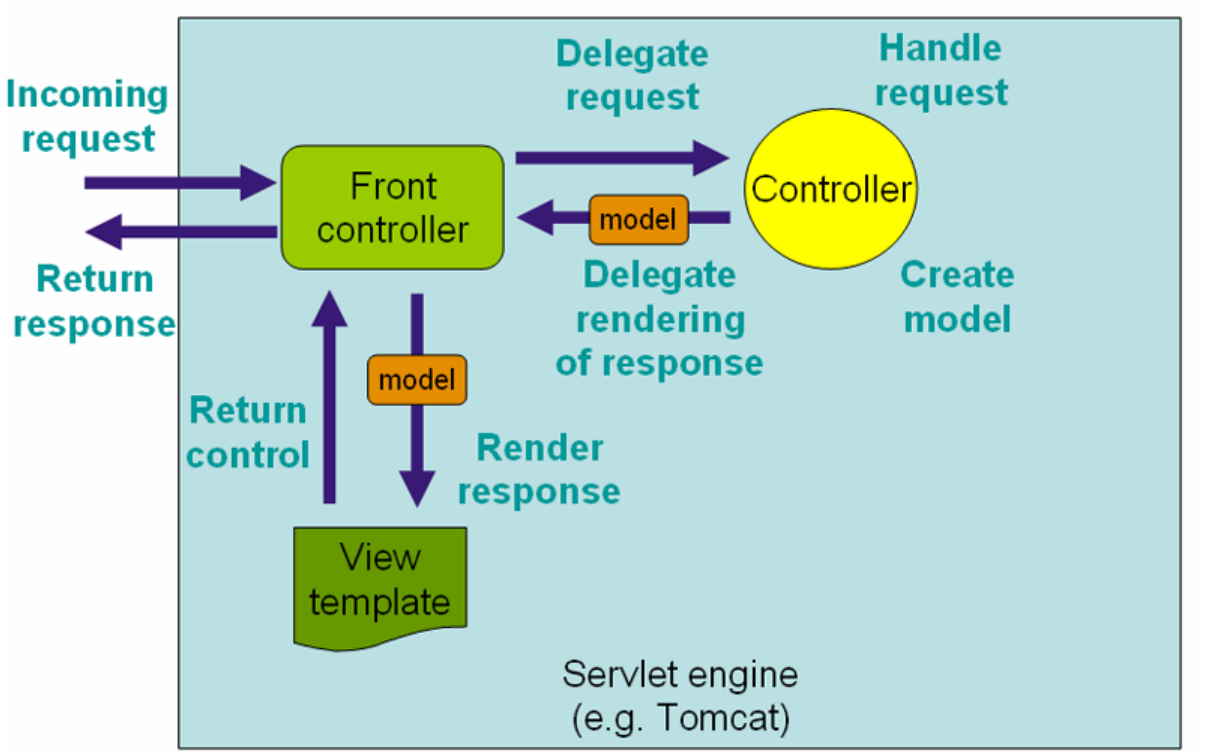

Figure 2: Request processing workflow in Spring MVC

- *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].

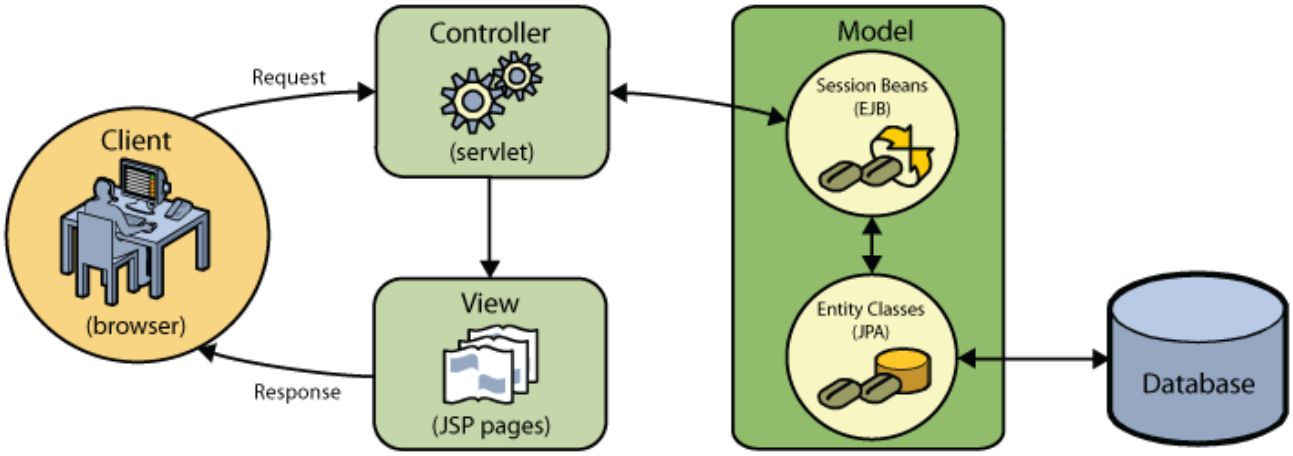

Figure 3: Context hierarchy in Spring Web MVC

3. System analysis

3.1. Current process

Currently, there's no organised process for the University of Surrey community 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.

3.2. 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 assess 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 represents a number of suggestions that Would be implemented as part of future development to improve the system. [25]

3.2.1.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 must allow the users to search products based on categories e.g. books or room swaps.	Functional	M
8	The solution must provide a search functionality (text-box search) to allow the users enter keywords.	Functional	M
9	The solution should enable the users to sort the products based on price (Low to High, or High to Low).	Functional	S
10	The solution must provide information about the product: Title, Description, Price, Category and Seller contact information.	Functional	M
11	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
12	The solution should provide a 'My profile' area for each user.	Functional	S

13	The solution should allow users to bookmark items.	Functional	C
14	The bookmark functionality must protect users' bookmarks and follow data privacy policies.	Non-functional	M
15	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
16	The solution won't provide delivery charges, stages involved in ordering process and dispatch information.	Functional	W
17	The solution won't provide refund functionalities.	Functional	W

3.2.2.Feasibility analysis

Feasibility study [9] is conducted as part of the assessment process. The aim of the project analysis was to identify trade-offs and outline alternative options. The feasibility analysis helped recognise issues and potential risks in the early stages of the software development system. One of the first risks identified was 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 with the development of the digital Pinboard.

Even though the Pinboard website could not be deployed on the University's intranet, it was an opportunity for the final year computing student to plan and deliver the project through the different phases of systems' development lifecycle. It was also an opportunity to achieve a challenge and develop a project based on a new Framework without previous Spring MVC experience.

3.3. Summary of deliverables

I might not include the summary of deliverables as the key deliverable is the full development of the website (including the report and video). Those deliverables are assumed.

Deliverable name	Description
Project Design	The structure of the solution and the plan to implement the 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 will be responsible for the maintenance of the website if the deployment of the solution is accepted.
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 will be responsible for the maintenance of the website if the deployment of the solution is accepted.

3.4. Stakeholders

Stakeholders [8] are individuals or organizations with interest in a project. Typically they are internal or external investors, employees, customers and suppliers. Stakeholders of the digital Pinboard solution include the target audience, which is a subset of students; sellers of second-hand books and seekers of room swap 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 if the deployment of the solution is accepted.
Chara Katiri	Computer Science Student, the developer.
Dr Steve Wesemeyer	Professional project supervisor and coordinator.

4. System design

4.1. 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 through numerous phases. In contrast to Agile methodology, Waterfall follows a linear approach in software development where the complete scope of requirements is known in advance.

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.

4.2. 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.

4.3. Technology options

4.3.1. Architecture and programming language

The system architecture is separated in several layers to achieve efficiency and maintainability of the website.

- *Presentation layer:* is responsible for the (i) View and (ii) the presentation of data.

(i) The View, which is also known as JSP pages in Web MVC terminology, generates responses and forwards them to the controller. The controller than sends the response to the client.

(ii) The presentation of data is archived by HTML5 and CSS. In order to ease maintenance and achieve maximum efficiency, the Style, Behaviour and Structure are clearly separated. The Style includes links of imported style sheets or external CSS files. The Behaviour includes links to local script elements or imported JavaScript files. Finally, the Structure includes the HTML structural elements.

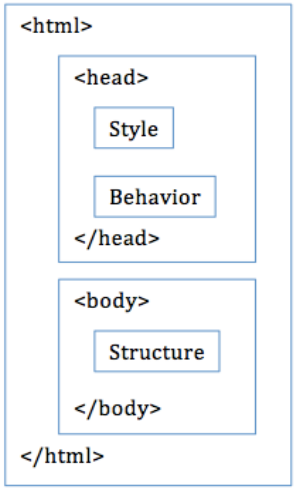

Figure 4: Separation of Style, Behavior, Structure

- *Security layer:* it deals with security concerns and it's implemented in layers using a number of technological controls. First, the use of the University of Surrey email address restricts access only to students and staff of the University. Then, upon login, authentication of the user and authorization to access the website is applied. Spring MVC framework libraries also provide security. Finally, Hyper Text Protocol Secure (HTTPS) is used as a communication protocol for secure communication over the network.
- *Business logic:* it consists of Spring MVC framework controllers and services. The business logic receives the incoming information, performs specific operations and defines the content that needs to be returned to the user ensuring that the required functionalities are always available.
- *Database layer:* it consists of the Web server, the Application server, and the MySQL database.

  Unsure if it's better to separate the layers in a different way: Application layer (business logic), DAO (database services), ORM (data mapping), Data source (connection pool), database (data).

4.3.2.Security levels

- *Security defence – SQL injection*: Login is required for buyers in order to view the list of available items, and for sellers in order to add new items. 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's

account is authenticated and authorised to allow access to the website. Text submitted by the user is always treaded as suspicious thus not only client side but also server side validation was 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' [28].

- *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.

4.4. 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 with no previous experience is a steep learning curve.
- *SHH Tunnel:* Connectivity issues due to SHH Tunnel problems from local workstation to remote server.
- *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), thus a unique reference to each item must be created.
- *Risk of non-deployment of the website on the University of Surrey intranet:* If the Service Team/IT Services Team does not accept the deployment of the website, volunteers cannot test the web application or provide feedback for improvements.

4.5. System risks and issues

4.5.1. 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.

4.5.2. System issues

Collection of junk emails: One of the initial aims of the system was to collect the number of the junk emails sent within the University. Then, the plan was to analyse the number of emails sent before and after the deployment of the website in order to measure the efficiency of the website though the functionalities it offers. A request was submitted to the IT services team to extract the average number of junk emails exchanged on Surrey's network. In response the team 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. (The email trail can be found in section 10. Appendix).

4.6. System constraints, dependencies and timescales

4.6.1. 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.

4.6.2.System dependencies

I could only think of dependencies as the issues described in section '4.5.1 System Risks'. Can I remove the system dependencies section?

4.6.3.System timescales

Following the gathering and definition of functional responsibilities a project plan is developed. Time limits for scheduling were created to ensure that the required time is allocated for the completion of each task. Milestones and deliverables were defined to keep track of the project progress and help get closer to the finished product. Then the Critical Path was defined. Activities included in it are critical and any delay in delivering those will have a negative impact on the rest of the project and will result delay in the delivery of the project [29].

The aim of the project plan was to ensure balance between timescales and project quality. Details in the R&D phase were important but it was equally important to avoid destructions and remain focused on the project goal [30].

The detailed project plan including Milestones and resources allocated for each task is shown in the Gantt chart below. The optimal timescales for system's development is 25% Research and Design, 45% Development and 30% Testing.