STUDENT INDUSTRIAL PROJECT(SIP) REPORT

DURATION OF INTERNSHIP:

JANUARY 2022 – AUGUST 2022

HOST COMPANY:

CYBER VILLAGE SDN. BHD.

STUDENT NAME: LIM WUAN LU

STUDENT ID: 19000420

PROGRAMME: BACHELOR OF INFORMATION TECHNOLOGY WITH HONOURS

VERIFICATION STATEMENT

I hereby verify that this report was written by LIM WUAN LU and all information regarding this company and the projects involved are NOT CONFIDENTIAL / ~~CONFIDENTIAL~~ (strikethrough not relevant).

Host Company Supervisor's Signature & Stamp	
Name:	Lee Fee Lan
Designation:	Project Manager
Host Company's:	Cyber Village Sdn. Bhd.
Date:	10 Aug 2022

ACKNOWLEDGEMENT

First and foremost, I would like to take this chance to express my gratitude to Universiti Teknologi PETRONAS(UTP) for developing such a great internship program for their undergraduate students to develop their knowledge and skills in a real-world setting and to demonstrate their commitment and enthusiasm for their work. Additionally, the Centre for Student Development(CDO) did an outstanding job of ensuring all students receive adequate and first-hand information about internship matters. They act as a link between undergraduate students and reputable companies, sending us notification emails frequently to keep us updated on our status and circumstances.

Sincere appreciation to my host company, Cyber Village Sdn. Bhd. for accepting my application as an industrial trainee, allowing me to complete my internship in a well-established company for the full 7 months. Also, I would like to thanks my reporting manager and seniors for guiding me during this internship period. I am deeply thankful for the effort that everyone in this team has put out to make me understand and learn about the company and knowledge regarding my tasks.

Last but not least, my heartfelt thanks go to my family, my deepest gratitude, and my deepest respect. Your encouragement has kept me going until now. I would also like to thank all my friends and colleagues who have supported me and inspired me to pursue my goals.

I see this opportunity as a milestone in my career. I will do my best to make use of the skills and knowledge I have acquired, and I will continue to strive to improve in order to achieve my career goals.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 ABSTRACT

Java is Object Oriented Programming language which is more popular among the developers and both small or large standalone, web-based, and mobile applications can be developed. However, to develop a application is time consuming and human errors might occurred during configuration and setting up the project. Hence, frameworks are created to be reused again so that developers can create programmes without having to manually create everything from scratch. This project is aimed to develop an error-free system with minimum effort and time by utilizing proper Java framework. Hence, Spring Boot Framework is being utilized in this project to develop a Document Management System. The methodology that has been implemented in this project is Agile Methodology as it can allow developer to be always on track throughout whole implementation of project. Last but not least, the results and discussions will be showed by leveraging on diagrams and tables in order to have a better visions of the outcomes.

2.0 INTRODUCTION

2.1 BACKGROUND OF STUDY

Java as a language is becoming more and more important as the world moves more and more to mobile apps and web applications. It is one of the most powerful languages we see and has been ranked as the third most popular language among recruiters in the last two years. Hence, frameworks are created to be used over and over, allowing developers to code their applications without hassle of building everything from scratch. Frameworks often determine the structure of an application. In fact, some frameworks provide so much code that developers don't have to do much to write an application. This can be good or bad, depending on ease of use.

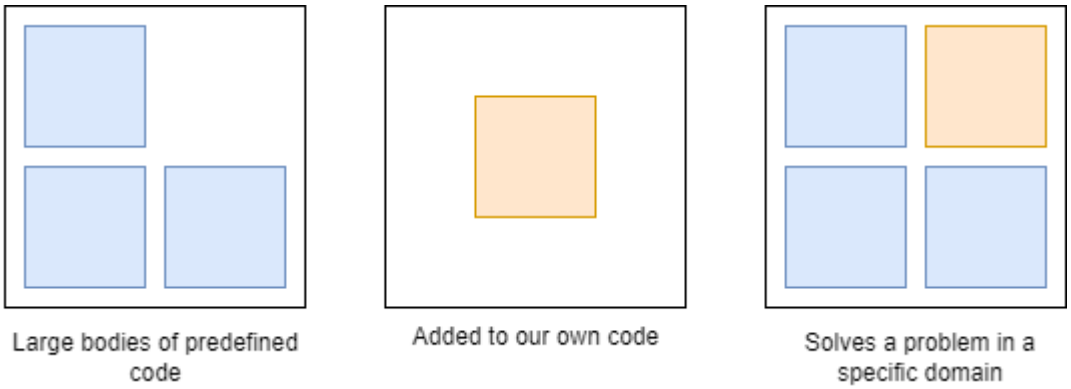


Figure 2.1.0 Java Framework Illustration

Besides, Java Framework is the body or platform of pre-written codes that Java developers use to develop Java applications or web applications. In other words, the Java Framework is a collection of predefined classes and functions used to process input, manage hardware devices, and interact with system software. It behaves like a skeleton that helps developers write their own code to develop applications.

There are several Java Frameworks which developers can decide which one to use based on their preferences and suitability to develop their applications. In this project, Spring Boot framework will be adopted to develop a Document Management System.

Spring Boot is built on the classic Spring framework. As such, it offers all the functionality of Spring and is easier to use than Spring. Spring Boot is a microservices-based framework that let developers build production-ready applications in no time. Everything is automatically configured with Spring Boot. Appropriate configurations must be used to take advantage of certain features. Spring Boot is also useful when developing REST APIs. The table below is

showed to clearly understand what the difference between Spring Framework and Spring Boot Framework is:

	Spring Framework	**Spring Boot Framework**
Definition	Open-source web application framework based on Java	An extension or module built on the Spring framework
Function	Provides flexible, completely configurable environment using tools and libraries of prebuilt code to create customized web apps	Provides ability to create standalone Spring applications that can just run immediately without the need for annotations, XML configuration, or writing lots of additional code
Key Feature	Dependency injection	Autoconfiguration
Servers	Need to set up servers explicitly	Comes with built-in HTTP servers like Tomcat
Configuration	Provides flexibility, but its configuration has to be built manually	Configures Spring and other third-party frameworks automatically by default "convention over configuration" principle
Approach	Unopinionated	Opinionated
XML configuration	Required	Not required

Table 2.1.0 Difference between Spring Framework and Spring Boot Framework

Besides, there are different layers and classes present in the architecture of Spring Boot. There are four main layers in Spring Boot:

a. Presentation Layer
   Consists of views (frontend part)
b. Data Access Layer
   CRUD (create, retrieve, update, delete) operations on the database
c. Service Layer
   Consists of service classes and uses services provided by data access layers

d. Integration Layer

Consists of different web services (any service available over the internet and uses XML messaging system)

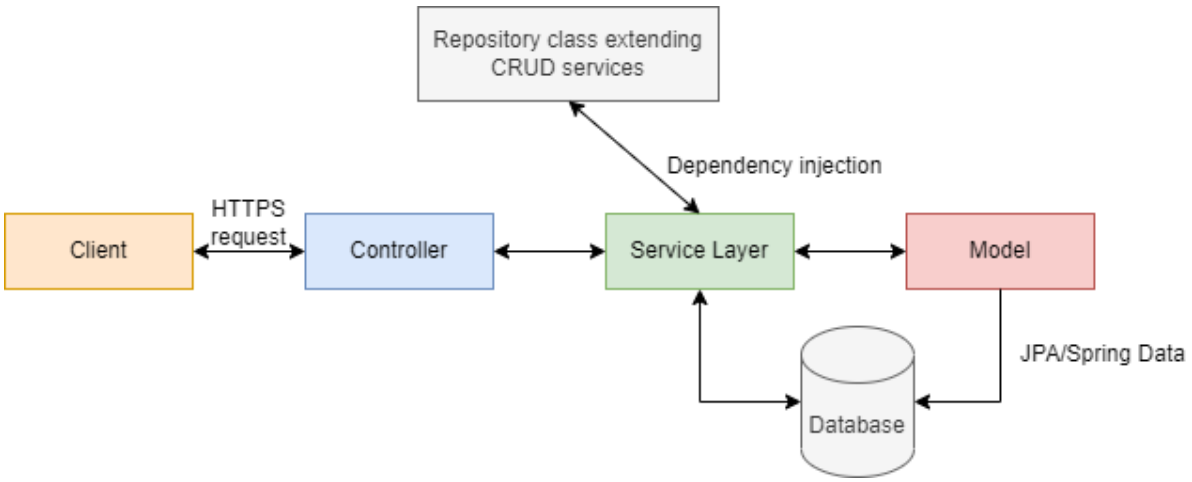


Figure 2.1.1 Flow Architecture of Spring Boot

2.2 PROBLEM STATEMENT

The information technology sector is enormous, diverse, and dynamic. An essential component of the industry, software development is directly impacted by a number of variables, such as rising complexity, market conditions, shifting technological trends, and rising software development issues. Teams of software engineers encounter a variety of difficulties across various industries and business environments. For instance, the "build once, deploy everywhere" paradigm, in which a single programme may run across various platforms, is now more frequently used to guide software development initiatives. The requirement to integrate many programmes and continue support for the duration of the product's lifecycle further complicates this problem. Additionally, software developers now have to meet more demands, which forces them to operate with tighter budget and timeframes.

Developing in Java has its own problems. Problems can arise from inadequate tools, lack of experience, or the project itself. Developers will face them every day, whether the build is broken, or security issues. The following are some of the issues developer might face during project implementation:

- Project Infrastructure

In terms of how it affects project delivery, an unestablished project environment is always a typical software development difficulty. If the environment is unavailable, there is no way to complete the project on schedule and within projected budget. Therefore, test and pre-production environments should be made available during the development, testing, and user acceptance testing (UAT) phases of a project to support efficient project development. Also, making an early investment in a reliable IT infrastructure to improve the environment for software development.

- System and Application Integration

For enterprises, there are countless varieties of technology, programmes, and systems. The project becomes significantly more complex when integrating third-party or other custom programmes, such as database management, website, or ERP systems. The major issue with integration is that is causes additional expenses, delays, poor quality, and occasionally even project failure because it is buried throughout the software development process and only becomes apparent at the very end. Hence, to fit the external constraints of other systems, the software solution should be adjusted by clearly comprehend the needs of the target user, implement a corporate framework for the application's platform architecture, discover and create new solutions or technologies, and the most importantly, to test and assess concepts to achieve best integration.

- Security Infrastructure

According to a recent survey, 96 percent of all web apps have at least one significant vulnerability, which rises the growth in security breaches. Hence, it is important to maintain the security of each layer of the programme or application. All parties involved, including management, project managers, business analysts, and developers are accountable for security. The best practises should be kept in mind to safeguard the organisation and infrastructure, for example, look beyond technology to increase software's security, utilized high-level programming language with built-in security measures to create applications, require security assurance procedures like code reviews and penetration tests, and carry out crucial foundational tasks to create safe apps and systems.

2.3 SCOPE OF WORK

After the Project Title has been decided and approved, different works are done in order to be able to implement this project smoothly with full understanding. The scope of work for Spring Boot Framework adoption on Document Management System consists of the following:

- Research on the key features of Spring Boot Framework

Do research on Spring Boot Framework to clearly understand its features in order to develop this project smoothly and efficiently. Besides, identify the advantages and disadvantages of Spring Boot Framework to fully utilized its advantages and to avoid its disadvantages during the implementation of this project. By doing so, a more favourable outcome could be produced.

- Understanding and explore the Spring Boot Framework

To study and explore the Spring Boot Framework by implementing some testing program in the environment. Also, exploring the dependencies, annotations, configurations, and so on in order to have more understanding to develop this project.

- Development of Document Management System

To list out the functions to be included in the system, then draft the user interface to have a clearer vision on implementing the system. Start writing the program by following the code structure of Spring Boot Framework in order to have a standardized outcome and to avoid unnecessary errors.

- Testing and Troubleshooting

After the system is ready, testing is done to make sure the system is free of errors by following the flow of user activities. If there is error occurs while testing the system, troubleshooting is carried out to identify the root cause of the error and to come out possible solution to fix the error in order to produce an error-free system in this project. After that, maintenance is done at the very last phase of this project to remove unnecessary code or to standardized the naming or indentation.

3.0 OBJECTIVES

The aim of this project is to develop a system with the least of effort to configure and set up the project in the environment. The ease of configuration and setting will help to reduce time consuming and hence can be more focus on the others project activities to produce a more satisfied result.

To expand on and be more specific based on the Student Industrial Project(SIP) primary purpose that was mentioned above, the additional details include the following:

- To solve complicated configuration on developing a Java web application
- To decrease boilerplate code issue
- To improve the management of dependency and code structure
- To ease the integration with different software
- To fully utilized the features of Spring Boot Framework
- To develop an error-free system within given timeframe

No matter how big or small, overcoming problems in software development is never simple. From challenges in requirements to integrating new technologies, from ensuring end-to-end security to challenges of duplicating efforts, product development requires development team to be clear and focused on the problem that have set up to solve in order to achieve the objectives.

4.0 LITERATURE REVIEW

The framework is a tool that makes people's work easier because it has numerous built-in functionalities that are available in the form of jars, and once jars are included, developers won't need to do any additional work. There are numerous ways to develop software, but each platform has certain advantages and disadvantages that affect how quickly and effectively it may be developed.

According to (Yongchang Ren, 2011), there is almost always a significant issue with the software development process. Sometimes, the quality and reliability of the software is inadequate, and its maintenance is also poor. Framework is therefore a superior solution to cut risk and save time and money. Besides, according to (JingJun Zhang, 2011), the majority of Java Web systems have embraced a number of open-source frameworks. These frameworks are helpful in many ways, including by helping to clarify programme logic and structure and by lowering the intensity of programmers' development. However, using OSGi always causes conflicts.

(Edwin, 2014) explains how the use of software frameworks, architectural patterns, and design patterns has lowered development costs, increased product quality, and simplified process management. Spring Boot is one of many Java frameworks that are available, and that ca be used depending on the project's requirements. Also, according to Singh (2015), Spring offers aspect-oriented programming, which addresses the problem of the separation of concerns at a much higher level. It also enables programmers to add features like transactions and security.

"Spring is a powerful framework for constructing corporate applications, and it can be readily integrated with struts and hibernate frameworks due to its lightweight feature," (Ankur Bawiskar, 2012) added about the framework. Other than that, according to Praveen Gupta's Spring Web MVC Framework for Rapid Open Source J2EE Application Development: A Case Study (2010), the software industry has a great demand for frameworks, which is why new and advanced frameworks are produced daily. Due to its traits and the fact that Spring Boot is an

expanded version of Spring Framework, Spring is a good framework when compared to other frameworks.

The struts framework with an effective MVC can help in developing the presentation tier, while Spring can handle business logic and hibernate will take care of the database, according to Neha Munsi (2014), who said that Spring and Hibernate framework can be combined to make an efficient E-commerce application. Also, according to Silva Maria Fonseca Silveira Massruha, frameworks can also be used in fuzzy operations. She goes on to explain that because a persistency layer is involved in these operations, open-source tools like Spring, Struts, and hibernate were crucial because they provided flexibility for future updates.

In short, there are numerous theories explaining frameworks and their benefits and drawbacks, thus research and analysis are required before working on any projects employing frameworks because sometimes selecting the incorrect framework might lead to significant problems in the future. For example, there are various differences dependent on framework, such as the difference between Spring and Hibernate, which is that Hibernate may be used to create connections using JDBC whereas Spring is better for developing enterprise applications. However, developers can utilise multi-frameworks to inherit the properties of various frameworks and create quick and dependable enterprise applications. If modifications are later required, it will be easier to update the code with multi-frameworks than with a single framework.

5.0 METHODOLOGY

5.1 METHODS AND TOOLS

Method

The methodology that is adopted in this project is Agile Methodology. Agile development is a new method taking the industry by storm. This software development methodology allows us to build a program incrementally with short iterations over 1 to 4 weeks. A step-by-step process helps align the development process with business needs. Agile development is an iterative software development methodology focused on creating adaptive products. A project goes through multiple iterations, each building on the previous iteration and gradually moving closer to the final product.

Agile methodologies also prioritize close face-to-face collaboration between development and business teams at all stages. Developers reassess business needs and development practices at the end of each iteration. Insights are used to adapt the product in the next iteration. Agile also welcomes ideas for new features that can be added with each iteration. While traditional development only considers business needs at the beginning of the process. This means that at the end of the development cycle, the product may not be optimized or meet business needs.

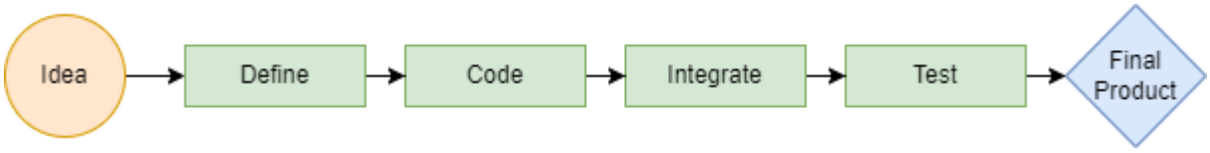


Figure 5.1.0 Traditional Development Method

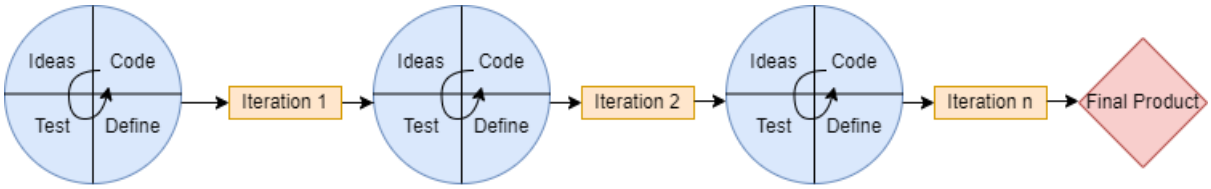


Figure 5.1.1 Agile Development Method

There are some advantages of Agile Methodology as below:

a. Flexibility and Adaptivity

   The Agile approach works best in a relatively uncertain environment. In such an environment, it is very difficult, if not impossible, to accurately define the requirements and design of the solution before starting the project. Flexibility and adaptability are essential to further define and refine the solution requirements and design as the project progresses.

b. Creativity and Innovation

   In the competitive environment we live in today, no one wants to buy an average off-the-shelf product. People expect higher levels of excellence, and that requires creativity and innovation. The Agile approach emphasizes creativity and innovation to maximize the business value of a solution. Too much emphasis on planning and control tends to stifle creativity and innovation.

c. Time-to-Market

   The Agile approach typically has a faster time-to-market due to a shorter start-up times. A staged development effort also allows us to deliver at least part of the solution early without having to complete the entire solution 100%.

d. Lower Costs

   The Agile approach can reduce project costs in several ways. Firstly, improve project team productivity. Next, significantly reduced overhead resulting from reducing unnecessary documentations and control requirements. Also, this approach makes it clear when a project begins to reach a point of diminishing returns where the incremental value of functionality no longer exceeds the incremental development cost.

e. Improved Quality

   In Agile project, quality is not a set of activities but an integral part of the development process. Developers know that quality is not “another responsibility”.

f. Customer Satisfaction

An Agile approach should result in higher customer satisfaction and more effective solutions, as the customer is deeply involved in providing feedback and input throughout the development process.

Tools

Throughout this project, I learned and utilized several tools and applications to help me in completing this project. Some of these tools are new to me, and some I am already familiar with. However, I have still learned new things about them. The tools that I have utilized are as follows:

1. Eclipse

Figure 5.1.2 Eclipse

Eclipse was initially developed as a Java-based integrated development environment. It is an open-source platform for extensible software development application frameworks, tools, and run times (IDE). Based on a variety of Equinox Open Services Gateway Initiative (OSGI) runtime-built open-source projects, Eclipse's runtime architecture supports Java IDE, static and dynamic languages, thick and thin client and server-side frameworks, modelling and business reporting, embedded and mobile systems, and more.

Despite the fact that Eclipse was designed for Java Applications, plug-ins enable developers to create software in a variety of other languages, such as C, C++, COBOL, Perl, PHP, and Python. Also, Eclipse may interact with modelling tools, concurrent versions systems, database management systems, network applications, and more thanks to the plug-in mechanism.

The key features of Eclipse IDE are as follows:

- The majority of things are plug-ins. The Eclipse Marketplace offers a variety of plug-ins, including those for static analysis, code style checking, and system version verification.
- Developers can increase the capabilities of the IDE by introducing plug-ins.
- Supports a variety of source knowledge features, including code editing with syntax highlighting, folding and hyperlink navigation, and a macro definition browser.
- Offers a visual debugging tool for code.
- Has a great user interface design feature.
- Supports source navigation, the administration of the framework, and the development of projects using several toolchains.
- Developers can automatically generate class documentation using Javadoc.

2. MySQL Workbench

Figure 5.1.3 MySQL Workbench

MySQL is a powerful database management system that was introduced in 1995. This DBMS uses SQL (Structured Query Language) to carry out different data-related operations and data modification. Fundamentally, MySQL operates under an open-source framework that is available to anyone who wants to use its services. These days, enterprises rely on the MySQL platform for a scalable and dependable data solution due to its wide range of capabilities and advantages. Moreover, MySQL has a definite advantage over products like Oracle databases and Microsoft SQL Server thanks to its simplicity. Additionally, when using this DBMS platform, you can use any programming language of

your choice for free. Another benefit of using MySQL lies in its ability to combine with other operating systems such as Linux, Unix, Windows, and others. Furthermore, MySQL lets you select its mode of implementation, so you can either use it online or after installing it in your local system.

The following features make MySQL a well-liked DBMS in the market:

a. High Performance: Great processing speeds and an intuitive interface are combined in a special way by the MySQL engine, which offers high performance. Additionally, it can host multiple clients at once and give them faster MySQL access from anywhere.

b. Compatibility: MySQL provides you with the opportunity to install a variety of web development tools in addition to its safe and low-latency data transfers.

c. Scalability: You can scale up or down your data load at any time using the MySQL platform. Additionally, this platform easily adapts to the majority of well-known operating systems, including Linux, OS X, Windows, etc.

The reason that I have utilized MySQL is because of the easy integration with Spring Boot. The following advantages come from using MySQL for deployment of Spring Boot-based applications rather than internal memory:

a. Unparalleled scalability provided by MySQL makes it simple to manage tightly linked Spring Boot applications that handle terabytes of data. Moreover, MySQL's primary characteristic is on-demand flexibility. This open-source option enables complete data modification and satisfies the particular database server needs of Spring Boot-based application for e-commerce companies.

b. Spring Boot has the ability to create standalone web apps with built-in servers. Now developer can reduce the time it takes for a query to execute by using MySQL Database rather than internal memory storage. The application created with Spring Boot MySQL Integration will be better suited for real-time operations and the user experience as a result.

c. Applications created using Spring Boot and powered by MySQL are not only affordable to produce but also a good option for tasks requiring an enterprise-level solution. In order to improve its capacity to package a complete service (such as user authentication) into a self-contained, fully deployable form that offers an API, Spring Boot can make use of MySQL. The installation and deployment of an application can be made much simpler in this approach. In the age of microservices, the Spring Boot MySQL Integration is also growing in prominence.

3. Figma

Figure 5.1.4 Figma

Figma is an online tool for designing user interfaces. With the help of Figma, you have access to all the resources you need for the project's design phase, including vector tools that can be used for full-fledged illustrations, prototype capabilities, and code production for hand-off. Figma is a browser-based programme, however there is a desktop version available for Windows and Mac OS. No matter whether this is the case or not, Figma is constantly running online inside of such programmes, which are probably wrap applications that are running a browser inside of them. In short, Figma is a UI & UX design tool that runs in the browser and has great design, prototyping, and code-generation tools. With powerful capabilities that assist teams working on every stage of the design process, it is currently the industry's premier interface design tool.

There are some key features of Figma as follows:

- Collaboration

  One of Figma's advantages is that real-time, live collaboration is possible. You can all log in to a design at once and make changes to it together with your team members. Because each of these designs is saved online, you never have to be concerned about a team member losing track of the project. You don't need to bother about pushing files to and from any third-party storage platform or transferring files between team members because the most recent modifications are always there in the file. This also implies that you and a client can start working on a design simultaneously. Therefore, you may still have a sit-down discussion with your client even if you are in a different location from them and are both focused on the same issue. The client is free to offer suggestions, which you can immediately put into practise. On top of all these live collaboration tools you can also leave comments directly on the canvas.

- Component Libraries

  With Figma, you can create libraries of reusable components that the entire team may use. Designers can get a head start on any existing design systems by using components, and if a component is updated in the central library, all designs are updated to reflect the changes.

- Prototyping and Code Output

  Create connections and hotspots on your design during prototyping to represent how a user might move around the interface. Figma can produce SVG, CSS, iOS, and Android code for the coding step.

- FigJam

  FigJam is the ideal place for everyone to digitally organize and brainstorm ideas. This tools includes many practical team features, including shapes, graphs, and objects, along with an internal voice conversation and timer for staying on track. In short, FigJam is useful for idea creation, planning, and gathering helpful contributions from your whole time.

Programming Languages

- Java
- HTML
- SQL
- JavaScript
- CSS

5.2 PROJECT ACTIVITIES

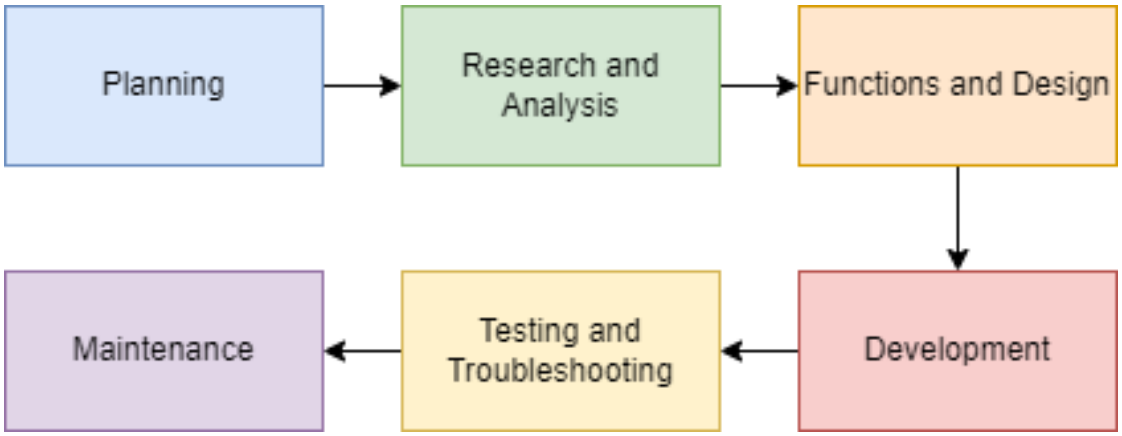


Figure 5.2.0 Flow of Project Activities

Planning

The first thing that was done is to decide on the project title for this Student Industrial Project(SIP). After that, define the project objectives that I want to achieve throughout the implementation of this project. Then, identify the scope of study according to the objectives that have been defined. Next, come out with a plan that is written in a table which including all the tasks, start date and end date. By referring to this plan throughout this project, I can always ensure that I'm on track and would not make my workload overloaded.

Research and Analysis

After planning for this project, I have start to do research on different Java frameworks and tools. The research is done by surfing on the internet and collect related data as much as

possible for my reference. After collecting relevant data and resources, I start to analyse the data that I have collected by identifying the key features, advantages, and disadvantages of the different frameworks and tools. Then, do comparison among the frameworks and tools, the one which is more compatible with my project title would be utilized. For example, I have chosen Spring Boot is because of its easy configuration and consist of dependency such as upload file, which will ease my development on Document Management System.

Functions and Design

After deciding on the frameworks and tools to be utilized, I start to determine the functions to be included in Document Management System and draft the UIUX for the pages using the tool Figma. The functions to be included are as follows:

- Role-based Login
- View, Download, Share, Upload Documents
- User Activity
- Inbox
- Pending (approve/reject by admin)
- Create, Update, Delete user

The following are the flow chart for each functions in Document Management System:

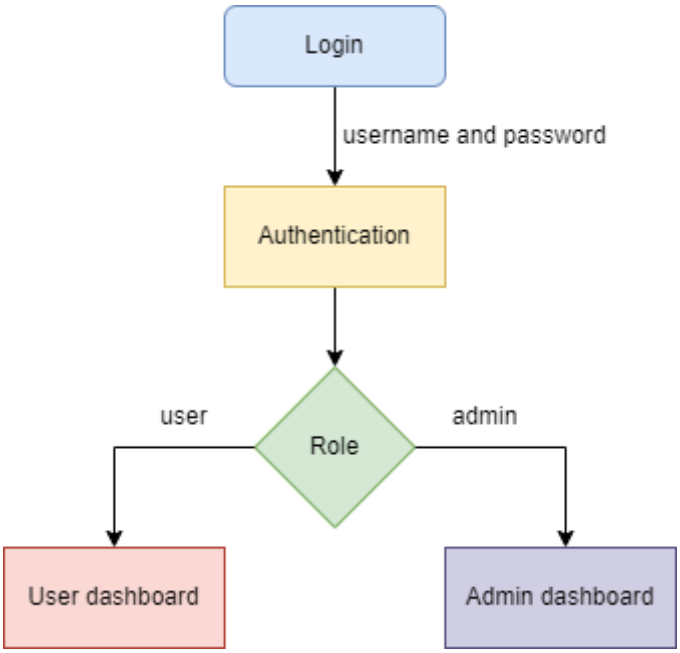


Figure 5.2.1 Flow of User Login

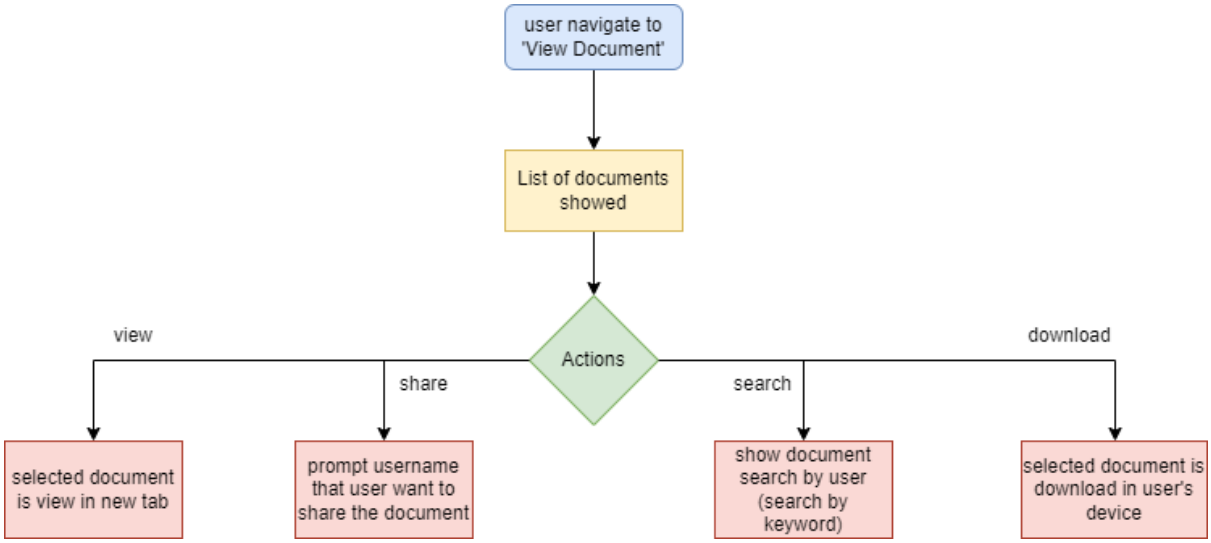


Figure 5.2.2 Flow of View Document

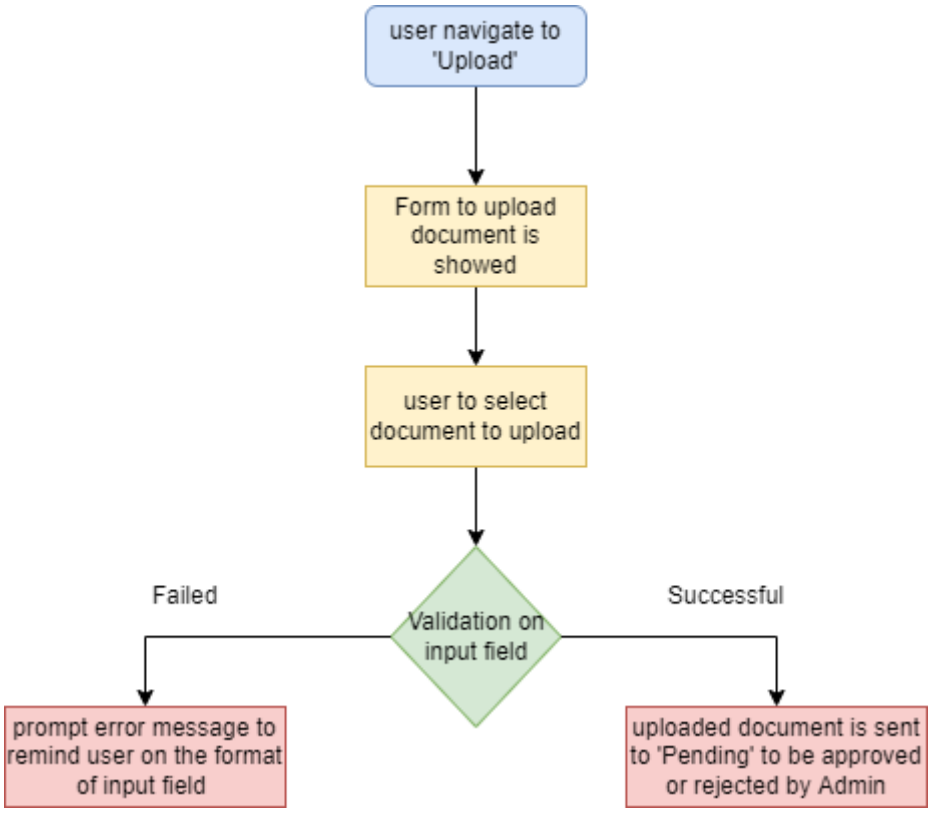


Figure 5.2.3 Flow of Upload Document

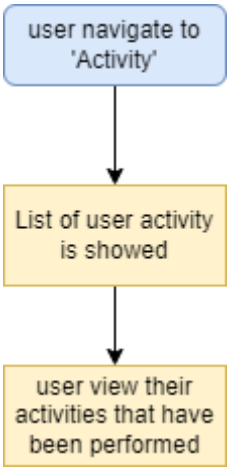


Figure 5.2.4 Flow of Activity

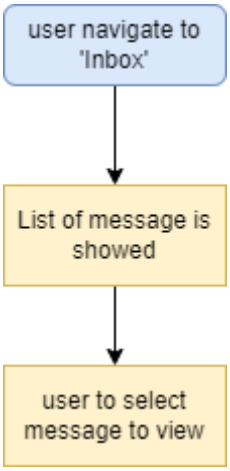


Figure 5.2.5 Flow of Inbox

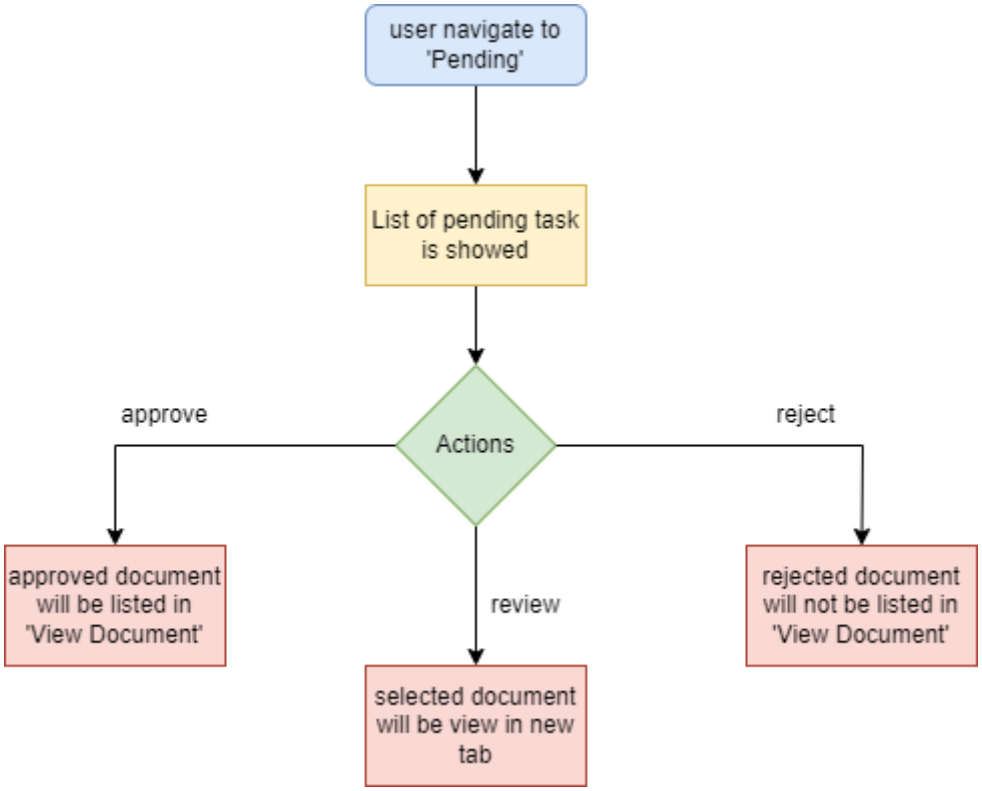
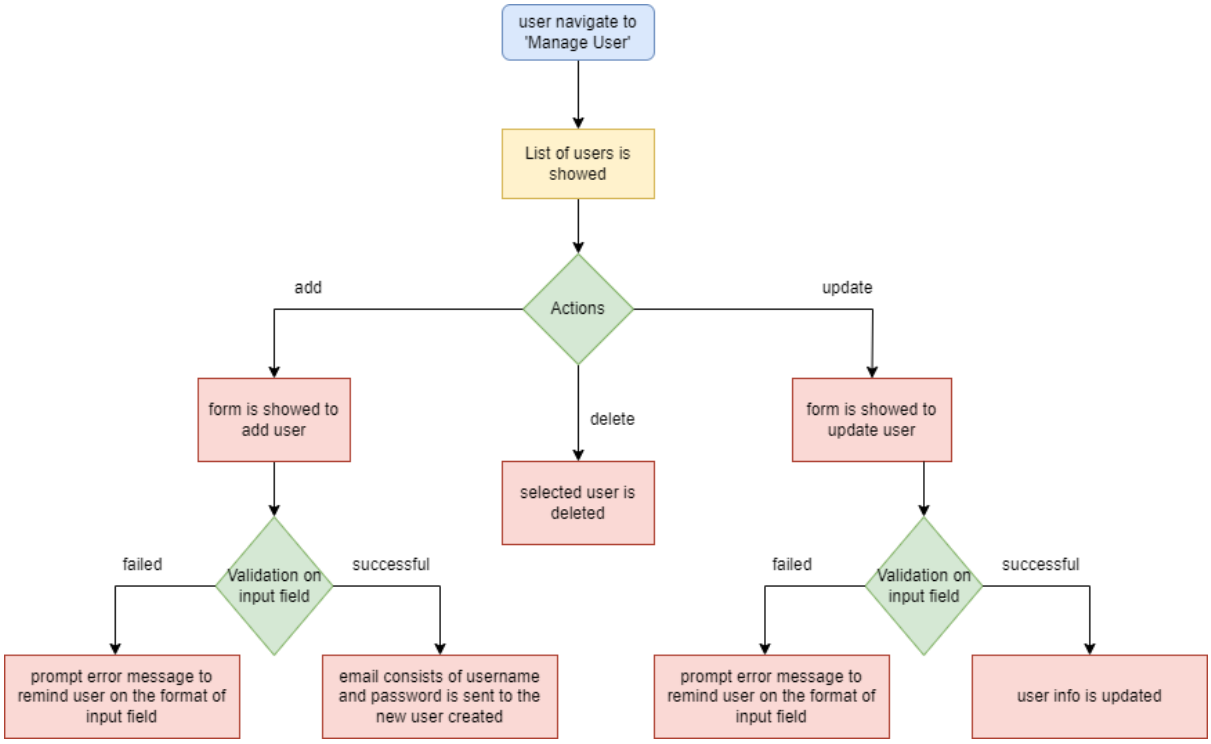


Figure 5.2.6 Flow of Pending



5.2.7 Flow of Manage User

Development

Firstly, development start by setting up the project in Eclipse. In development activities, I self-study and explore about the frameworks and tools by writing some testing programs and run the programs. By doing this, I able to deeply understand how the frameworks and tools work, and hence, some errors could be avoided during development of the project. Also, can be more productive and efficient throughout the development. Then, I start to develop the project which is logic implementation and writing code by following the agile planning.

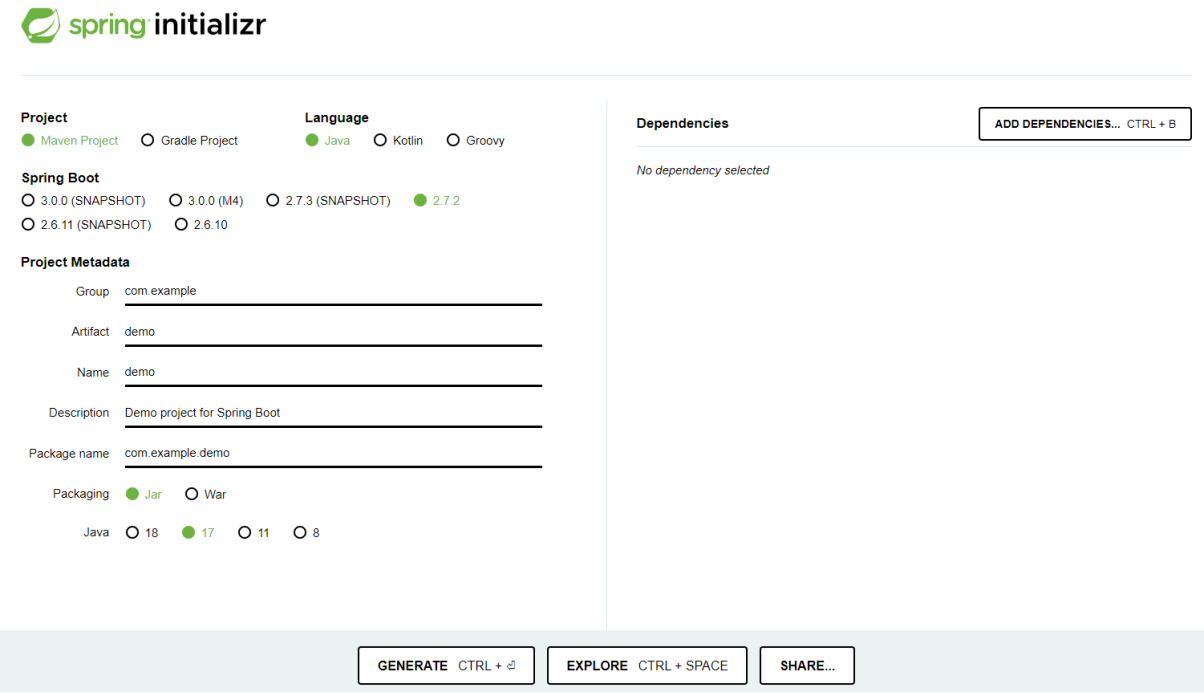


Figure 5.2.8 Spring Initializr

Testing and Troubleshooting

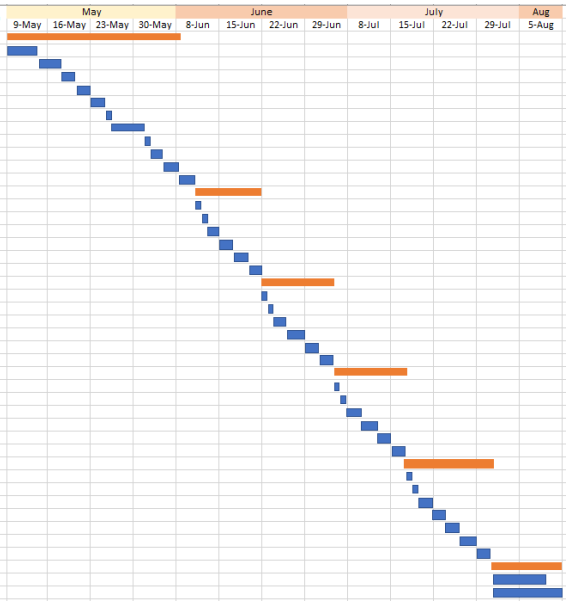
After development activities, I have test the system and identify if there is any error occurs. I test the system by following the flow, for example, user to upload document, admin to approve the document uploaded, document showed in page, etc. Then if there is error, I start to troubleshooting to come out with the possible solution to fix the errors.

Maintenance

After making sure the system is free of errors, some maintenance activities such as removing unused code, code standardization such as naming and indentation, delete test data that is not necessary in database, put comment for future reference, and so on.

5.3 GANTT CHART AND MILESTONE

No	Task	Descriptions	Start Date	End Date	Duration(day)
	Sprint 1		9-May	8-Jun	31
1	Project Planning	Define project objectives and scope of study	9-May	13-May	5
2	Research and Analysis	Determine which framework & tools to use	14-May	17-May	4
		Analyze the advantages & disadvantages	18-May	19-May	2
3	Functions and Design	Identify functions to be included	20-May	21-May	2
		UIUX drafting	22-May	24-May	3
4	Development	Setup Spring Boot project in Eclipse	25-May	25-May	1
		Explore framework & tools	26-May	1-Jun	7
		Database - create table	2-Jun	2-Jun	1
		Feature 1 - Role-based authentication	3-Jun	4-Jun	2
5	Testing & Troubleshooting	Review & identify error(if any)	5-Jun	6-Jun	2
		Possible solution to fix error	7-Jun	8-Jun	2
	Sprint 2		9-Jun	19-Jun	11
1	Project Planning	Determine feature to be develop in Sprint 2	9-Jun	9-Jun	1
2	Research and Analysis	Determine dependency to be used	10-Jun	10-Jun	1
3	Development	Feature 2 - View Document	11-Jun	13-Jun	3
		Feature 3 - Upload Document	14-Jun	15-Jun	2
4	Testing & Troubleshooting	Review & identify error(if any)	16-Jun	17-Jun	2
		Possible solution to fix error	18-Jun	19-Jun	2
	Sprint 3		20-Jun	1-Jul	12
1	Project Planning	Determine feature to be develop in Sprint 3	20-Jun	20-Jun	1
2	Research and Analysis	Determine dependency to be used	21-Jun	21-Jun	1
3	Development	Feature 4 - Activity	22-Jun	24-Jun	3
		Feature 5 - Pending	25-Jun	27-Jun	3
4	Testing & Troubleshooting	Review & identify error(if any)	28-Jun	29-Jun	2
		Possible solution to fix error	30-Jun	1-Jul	2
	Sprint 4		2-Jul	13-Jul	12
1	Project Planning	Determine feature to be develop in Sprint 4	2-Jul	2-Jul	1
2	Research and Analysis	Determine dependency to be used	3-Jul	3-Jul	1
3	Development	Feature 6 - Inbox	4-Jul	6-Jul	3
		Feature 7 - Manage User	7-Jul	9-Jul	3
4	Testing & Troubleshooting	Review & identify error(if any)	10-Jul	11-Jul	2
		Possible solution to fix error	12-Jul	13-Jul	2
	Sprint 5		14-Jul	26-Jul	13
1	Project Planning	Determine activities to be done for final sprint	14-Jul	14-Jul	1
2	Research and Analysis	Analyze current result (whether meet the target)	15-Jul	15-Jul	1
3	Development	Integrate all the features	16-Jul	17-Jul	2
4	Testing & Troubleshooting	Test for whole system	18-Jul	19-Jul	2
		Identify errors(if any)	20-Jul	21-Jul	2
		Troubleshooting to fix errors	22-Jul	24-Jul	3
5	Maintenance	Remove unused code, standardized code structure/na	25-Jul	26-Jul	2
	Presentation & Documentation		27-Jul	5-Aug	14
1	Presentation	Prepare slides	27-Jul	1-Aug	5
2	SIP Report	Prepare SIP Report	27-Jul	5-Aug	9



5.3.0 Gantt Chart

No	Task	Descriptions	Start Date	End Date	Duration(day)
		Sprint 1	9-May	8-Jun	31
1	Project Planning	Define project objectives and scope of study	9-May	13-May	5
2	Research and Analysis	Determine which framework & tools to use	14-May	17-May	4
		Analyze the advantages & disadvantages	18-May	19-May	2
3	Functions and Design	Identify functions to be included	20-May	21-May	2
		UIUX drafting	22-May	24-May	3
4	Development	Setup Spring Boot project in Eclipse	25-May	25-May	1
		Explore framework & tools	26-May	1-Jun	7
		Database - create table	2-Jun	2-Jun	1
		Feature 1 - Role-based authentication	3-Jun	4-Jun	2
5	Testing & Troubleshooting	Review & identify error(if any)	5-Jun	6-Jun	2
		Possible solution to fix error	7-Jun	8-Jun	2
		Sprint 2	9-Jun	19-Jun	11
1	Project Planning	Determine feature to be develop in Sprint 2	9-Jun	9-Jun	1
2	Research and Analysis	Determine dependency to be used	10-Jun	10-Jun	1
3	Development	Feature 2 - View Document	11-Jun	13-Jun	3
		Feature 3 - Upload Document	14-Jun	15-Jun	2
4	Testing & Troubleshooting	Review & identify error(if any)	16-Jun	17-Jun	2
		Possible solution to fix error	18-Jun	19-Jun	2
		Sprint 3	20-Jun	1-Jul	12
1	Project Planning	Determine feature to be develop in Sprint 3	20-Jun	20-Jun	1
2	Research and Analysis	Determine dependency to be used	21-Jun	21-Jun	1
3	Development	Feature 4 - Activity	22-Jun	24-Jun	3
		Feature 5 - Pending	25-Jun	27-Jun	3
4	Testing & Troubleshooting	Review & identify error(if any)	28-Jun	29-Jun	2
		Possible solution to fix error	30-Jun	1-Jul	2
		Sprint 4	2-Jul	13-Jul	12
1	Project Planning	Determine feature to be develop in Sprint 4	2-Jul	2-Jul	1
2	Research and Analysis	Determine dependency to be used	3-Jul	3-Jul	1
3	Development	Feature 6 - Inbox	4-Jul	6-Jul	3
		Feature 7 - Manage User	7-Jul	9-Jul	3
4	Testing & Troubleshooting	Review & identify error(if any)	10-Jul	11-Jul	2
		Possible solution to fix error	12-Jul	13-Jul	2
		Sprint 5	14-Jul	26-Jul	13
1	Project Planning	Determine activities to be done for final sprint	14-Jul	14-Jul	1
2	Research and Analysis	Analyze current result (whether meet the target)	15-Jul	15-Jul	1
3	Development	Integrate all the features	16-Jul	17-Jul	2
4	Testing & Troubleshooting	Test for whole system	18-Jul	19-Jul	2
		Identify errors(if any)	20-Jul	21-Jul	2
		Troubleshooting to fix errors	22-Jul	24-Jul	3
5	Maintenance	Remove unused code, standardized code structure/na	25-Jul	26-Jul	2
		Presentation & Documentation	27-Jul	5-Aug	14
1	Presentation	Prepare slides	27-Jul	1-Aug	5
2	SIP Report	Prepare SIP Report	27-Jul	5-Aug	9

Figure 5.3.1 Milestone

6.0 RESULTS AND DISCUSSION

6.1 RESULTS

In this Document Management System, there are two roles which are user, and admin. Different roles will have different access to specific functions. Following are the list of which role will have the access to specific functions in this system:

Functions/Roles	Admin	Non-Admin
View, Download, Share Document	/	/
Upload Document	/	/
Activity	/	/
Inbox	/	/
Pending	/	
Manage User	/	
Logout	/	/

The following are the user interface of Document Management System:

Figure 6.1.0 Welcome Page

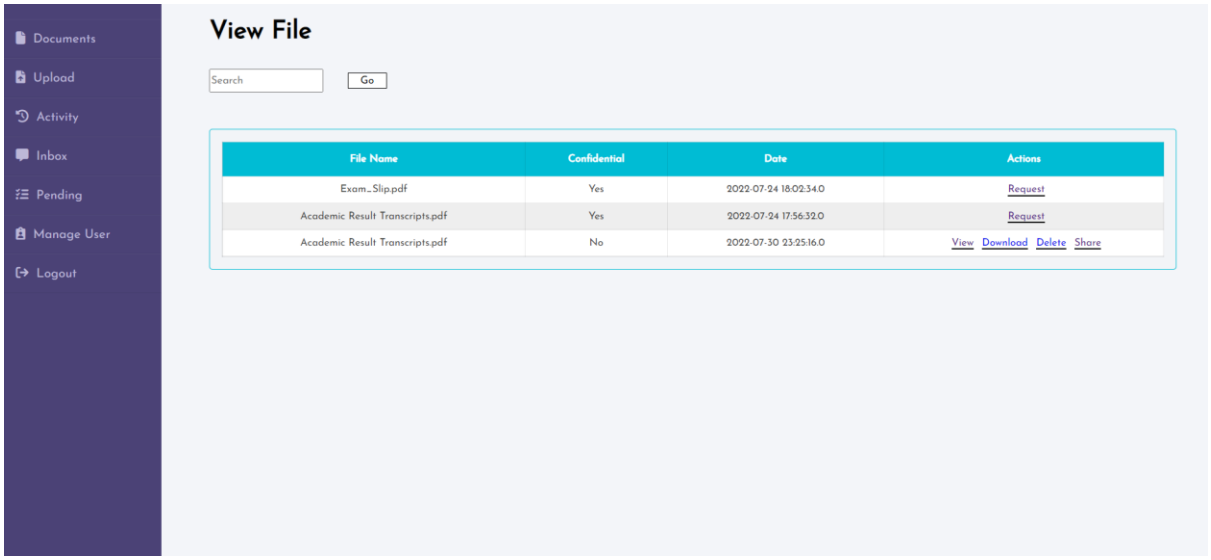


Figure 6.1.1 View Document Page

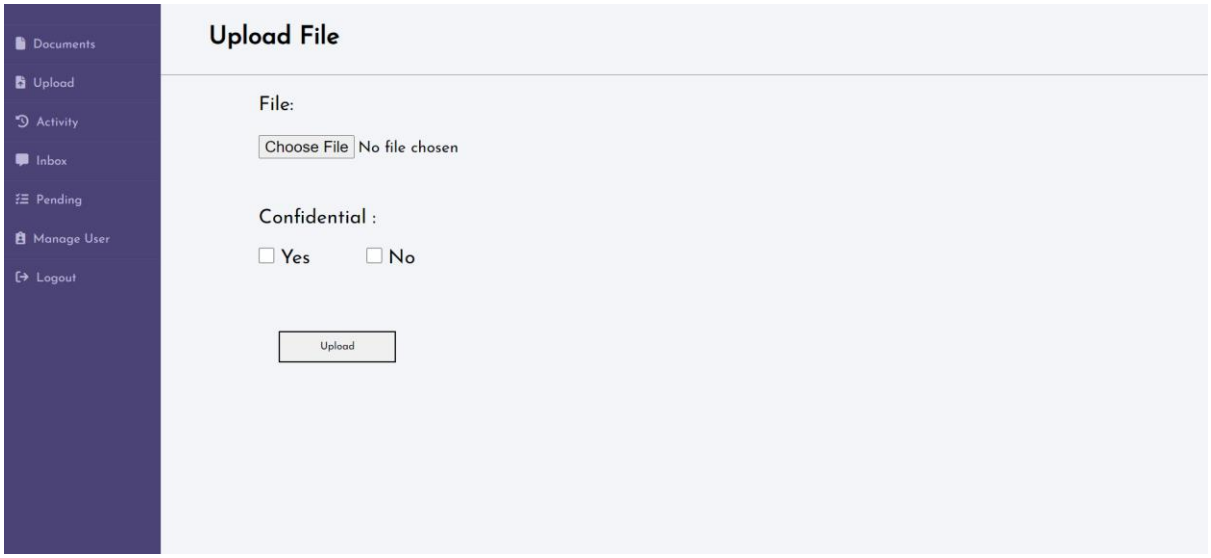


Figure 6.1.2 Upload File Page

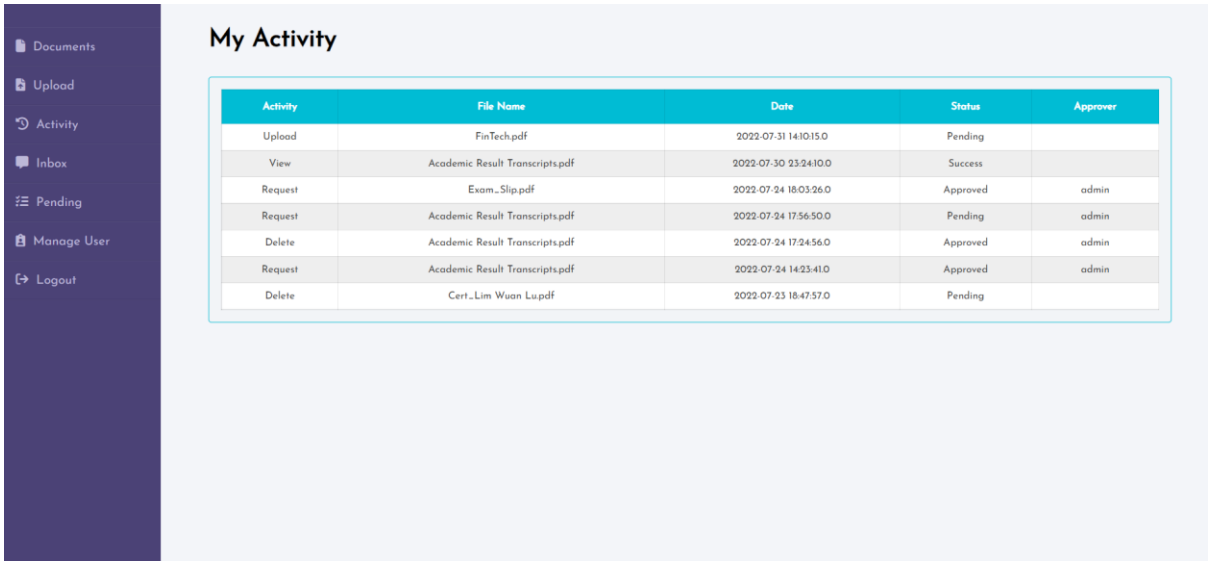


Figure 6.1.3 Activity Page

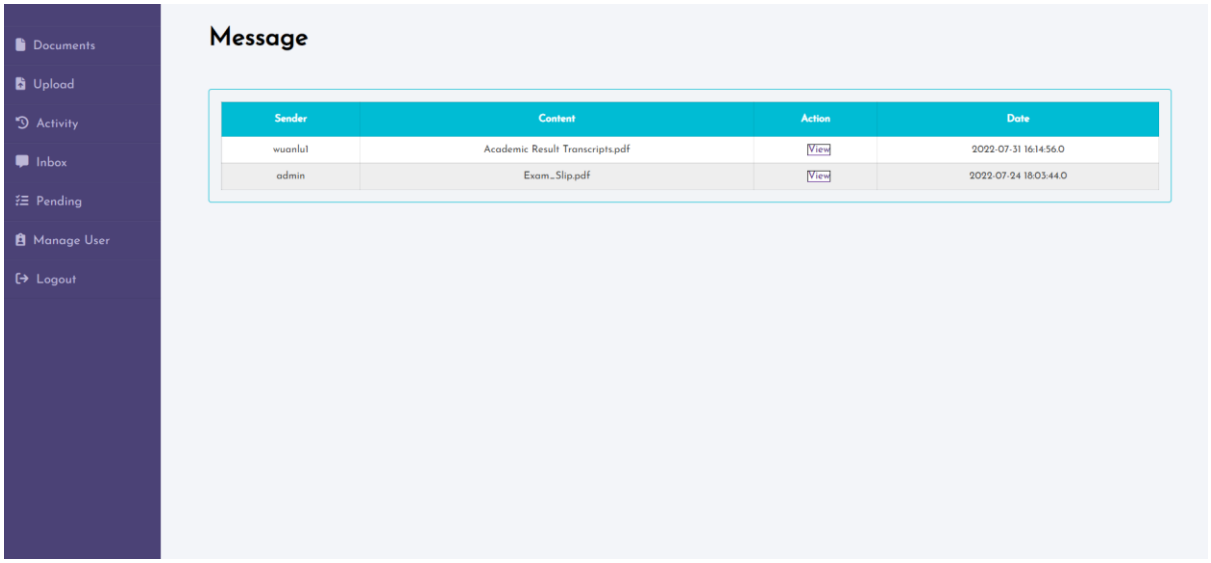


Figure 6.1.4 Inbox Page

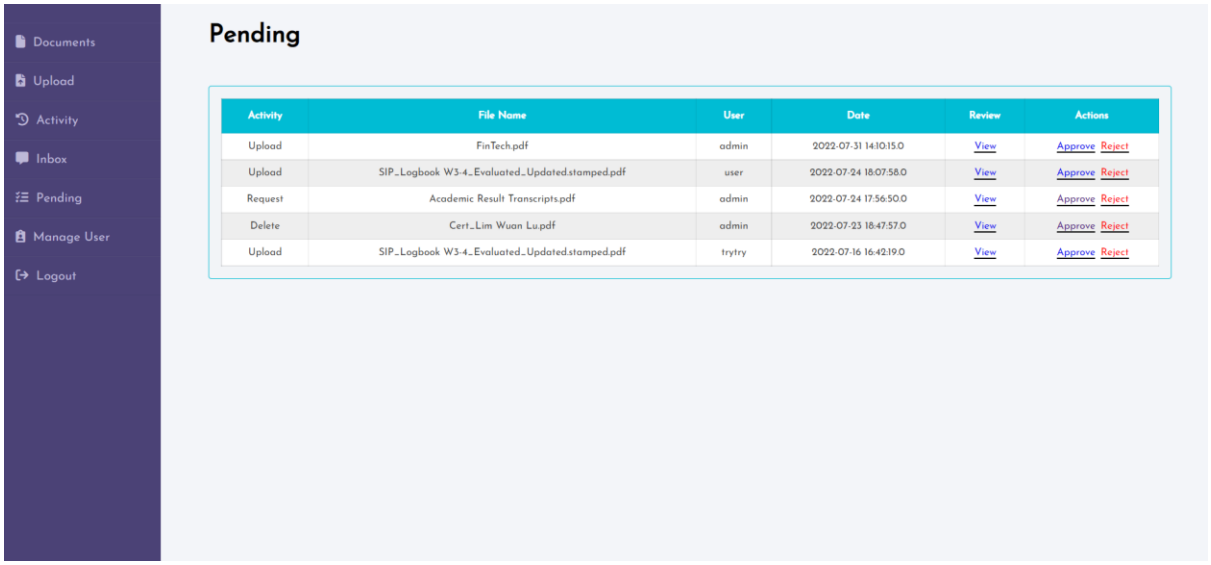


Figure 6.1.5 Pending Page

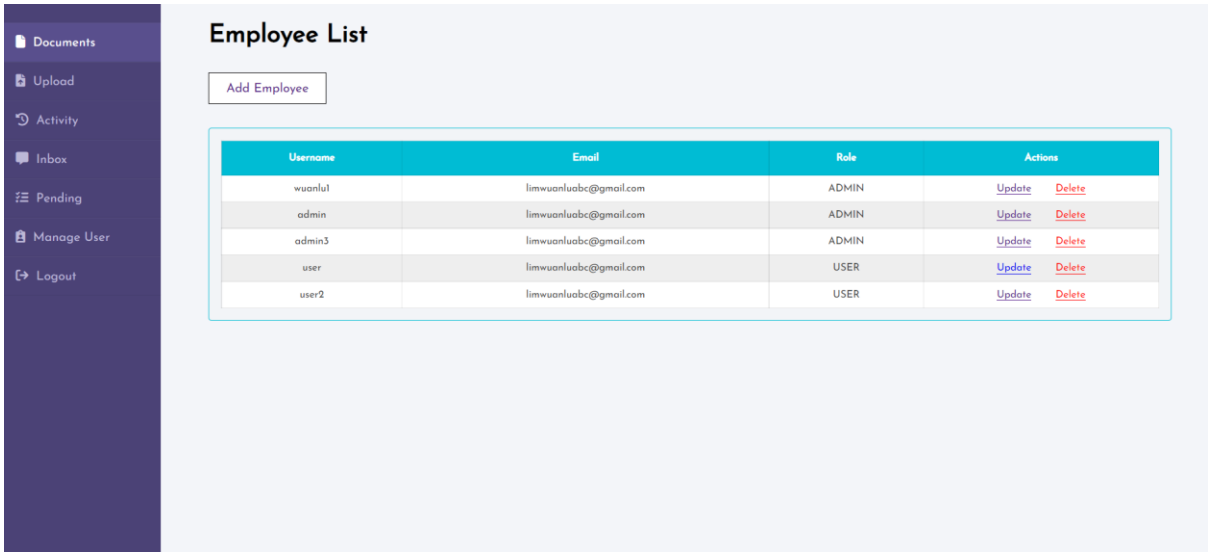


Figure 6.1.6 User List Page

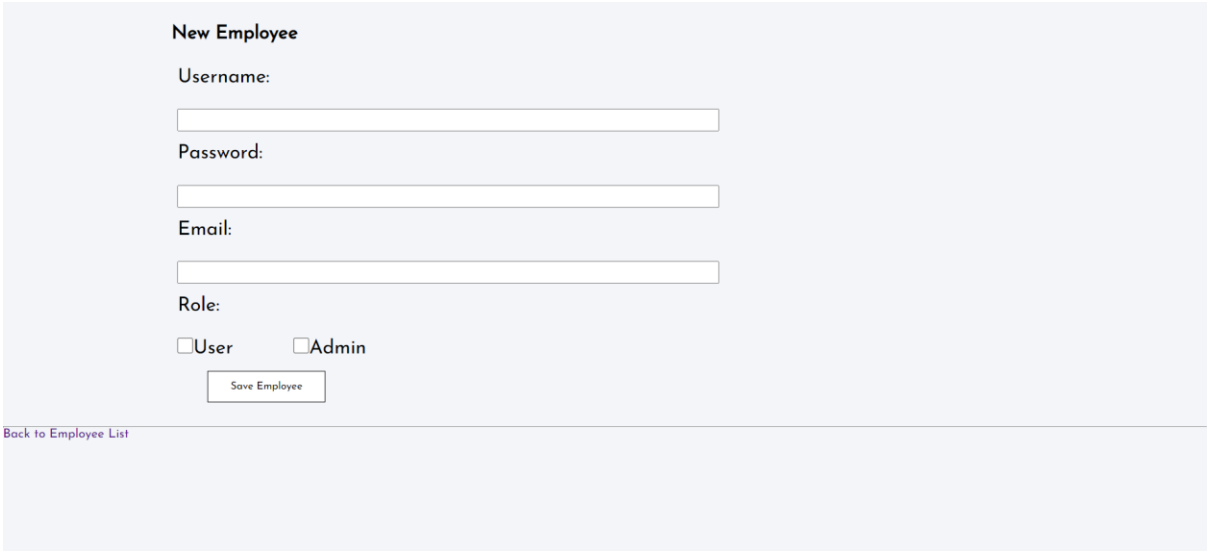


Figure 6.1.7 Create User Page

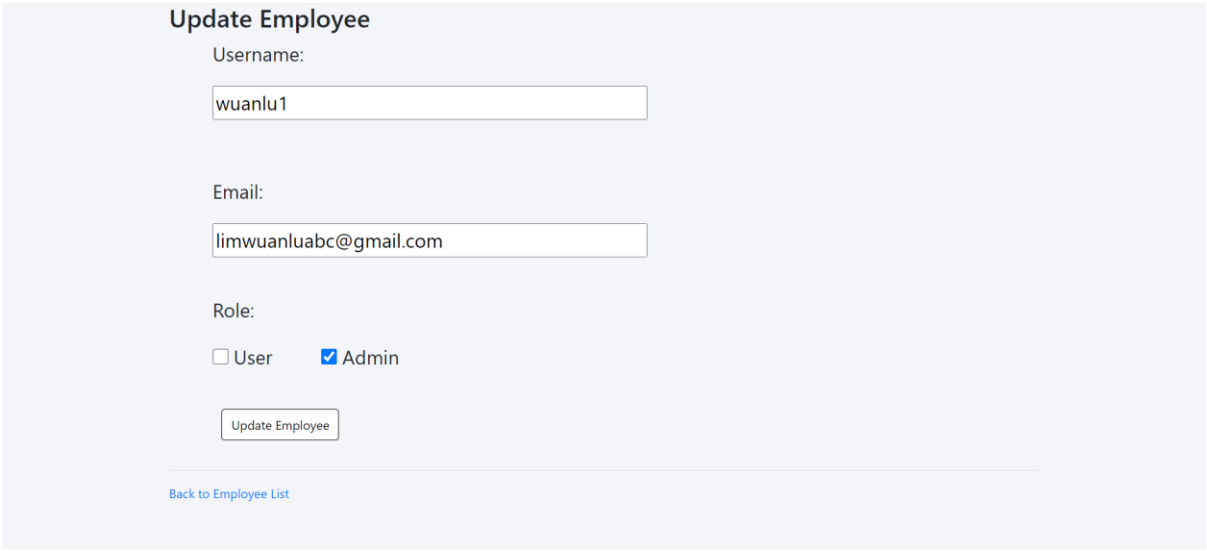


Figure 6.1.8 Update User Page

Project Structure

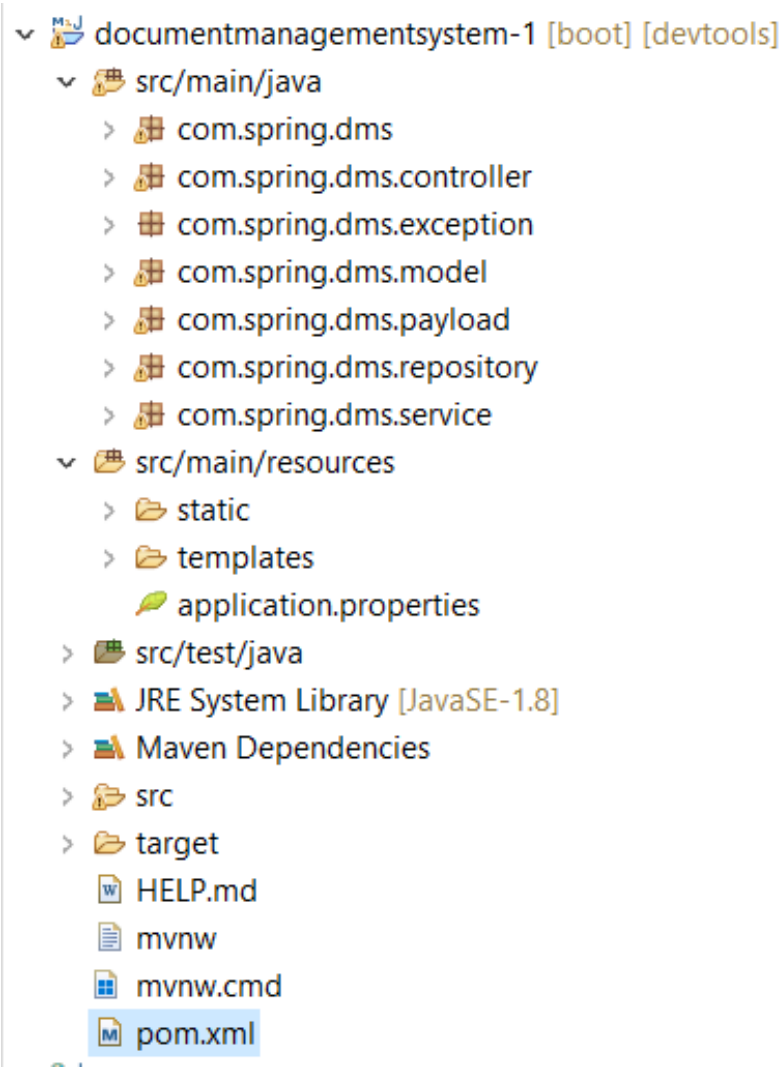


Figure 6.1.9 Project Structure

Code Structure

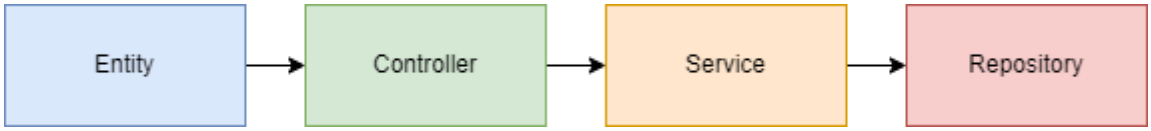


Figure 6.1.10 Code Structure

- **Entity**

A database is an organized collection of logically related data stored in software called a database management system (DBMS). Before the data enters the system, it needs to be in the form of a model to properly understand the database. Many models have been introduced and the entity-relationship model is one of them. It's a graphical representation of the data that gives you an overview of the data. This E-R model has basic building blocks including entities and objects. The figure below showed the example of an entity structure:

```
1  package com.spring.dms.model;
2
3  import java.util.Date;
16
17 @Entity
18 @Table(name = "myactivity")
19 public class MyActivity {
20
21     @Id
22     @GeneratedValue(generator = "uuid")
23     @GenericGenerator(name = "uuid", strategy = "uuid2")
24     private String id;
25
26     private String username;
27
28     private String action;
29
30     private String status;
31
32     private String fileName;
33
34     private String fileType;
35
36     private String approver;
37
38     private String fileStatus;
39
40     private String confidential;
41
42     private String fileId;
43
44     public String getFileId() {
45         return fileId;
46     }
```

Figure 6.1.11 Entity

- **Controller**

The Spring @Controller annotation is a specialization of the @Component annotation. The @Controller annotation indicates that a particular class assumes the role of controller. Spring Controller annotations are typically used in conjunction with annotated handler methods based on @RequestMapping annotations. Applicable only to classes. It used to mark a class as a handler for web requests. Also, it's mostly used with Spring MVC applications and this annotations acts as a stereotype for the annotated class, indicating its role. The dispatcher scans such annotated classes for mapped methods and detects @RequestMapping annotations.

```
1  package com.spring.dms.controller;
2
3  import org.springframework.beans.factory.annotation.Autowired;
11
12 @Controller
13 public class MyActivityController {
14
15     @Autowired
16     private ListActivityService activityService;
17
18     @GetMapping("/viewActivity")
19     public String viewActivityPage(Model model) {
20         final String currentUserName = SecurityContextHolder.getContext().getAuthentication().getName();
21         model.addAttribute("activity", activityService.getActivity(currentUserName));
22
23         return "activity";
24     }
25 }
```

Figure 6.1.12 Controller

- **Service**

In an application, business logic resides in the service layer, so the @Service annotation is to indicate that a class belongs to that layer. It is also a specialization of the @Component annotation as well as the @Repository annotation. One of the most important properties of the @Service annotation is that it can only be applied to classes. Also, it is used to identify a class as a service provider. So the general @Service annotation is used on classes that provide some business functionality. Spring Boot context will automatically discovers these classes when using annotation-based configuration and class path scanning.

```java
package com.spring.dms.service;

import java.util.List;

@Service
public class ListActivity implements ListActivityService {

    @Autowired
    private ListActivityRepository dbActivityRepository;

    @Override
    public List<MyActivity> getAllActivity() {
        return dbActivityRepository.findAll();
    }

    @Override
    public List<MyActivity> getActivity(String username) {
        return dbActivityRepository.findActivity(username);
    }
}
```

Figure 6.1.13 Service

- **Repository**

The @Repository annotation is a specialization of the @Component annotation and is used to indicate that a class provides mechanisms for storing, retrieving, updating, deleting on objects. Although a specialization of the @Component annotation, Spring Boot repository classes are automatically recognized by the Spring framework through class path scanning. This annotation is a generic stereotype annotation, much like the DAO pattern, where DAO classes are responsible for providing CRUD operations on database tables.

```
1  package documentmanagementsystem-1/src/main/java/com/spring/dms/service/ListFileImpl.java
2
3  import java.util.List;
16
17 @Repository
18 public interface ListActivityRepository extends JpaRepository<MyActivity, String>  {
19
20
21     @Query(value = "SELECT * FROM myactivity WHERE username = :username ORDER BY timestamp desc", nativeQuery=true)
22     public List<MyActivity> findActivity(@Param("username") String username);
23
24     @Query(value="select * from myactivity f where f.id like %:fileid%", nativeQuery=true)
25     MyActivity findFileById(@Param("fileid") String fileid);
26
27     @Query(value = "SELECT * FROM myactivity WHERE status = 'Pending' ORDER BY timestamp desc", nativeQuery=true)
28     public List<MyActivity> findPending();
29
30     @Modifying
31     @Transactional
32     @Query(value = "UPDATE myactivity SET status = 'Approved', approver = :currentUserName WHERE id = :id", nativeQuery=true)
33     public void approve(@Param("id") String id, @Param("currentUserName")String currentUserName);
34
35     @Modifying
36     @Transactional
37     @Query(value = "UPDATE myactivity SET status = 'Rejected', approver = :currentUserName WHERE id = :id", nativeQuery=true)
38     public void reject(@Param("id") String id, @Param("currentUserName")String currentUserName);
39
```

Figure 6.1.14 Repository

Database

The below are the tables that I have created in this project:

a. files

id	data	file_name	file_type	timestamp	confidential
0f0f8...	BLOB	Exam_Slip.pdf	applicatio...	2022-07-2...	Yes
1539...	BLOB	Academic Re...	applicatio...	2022-07-2...	Yes

Figure 6.1.15 files table

b. myactivity

id	action	data	file_name	file_type	status	timestamp	username	approver	file_id
7ce...	Delete	NULL	Academic...	NULL	Appr...	2022-07-...	admin	admin	34bb3...
6b2...	Requ...	BLOB	Academic...	applic...	Appr...	2022-07-...	admin	admin	1539f...

Figure 6.1.16 myactivity table

c. message

id	data	filename	receiver	sender	timestamp	fileid
12a...	BLOB	Exam_Slip.pdf	user	admin	2022-07-2...	0f0f8d...
ff00...	BLOB	Exam_Slip.pdf	admin	admin	2022-07-2...	0f0f8d...

Figure 6.1.17 message table

d. users

user_id	username	password	enabled	email	role
1	wuanlu	$2a$10$H...	1	limwu...	ADMIN
47	admin3	$2a$10$V...	1	limwu...	ADMIN

Figure 6.1.18 users table

e. roles

role_id	name
1	USER
2	ADMIN

Figure 6.1.19 roles table

f. users_roles

user_id	role_id
50	1
51	1

Figure 6.1.20 users_roles table

Dependencies

Dependency is a “Library” that offers particular functionality that developer can user in their application. Dependency Management and Auto-Configuration operate simultaneously in Spring Boot. It is the auto-configuration that makes handling dependencies easy for developer. Developer just has to include the dependencies in the pom.xml file. Then, Maven Central will download these new dependencies. Besides, the local file system’s “.m2” subdirectory will store the downloaded requirements. These dependencies are accessible to the Spring Boot application from the directory “.m2” and its subdirectories.

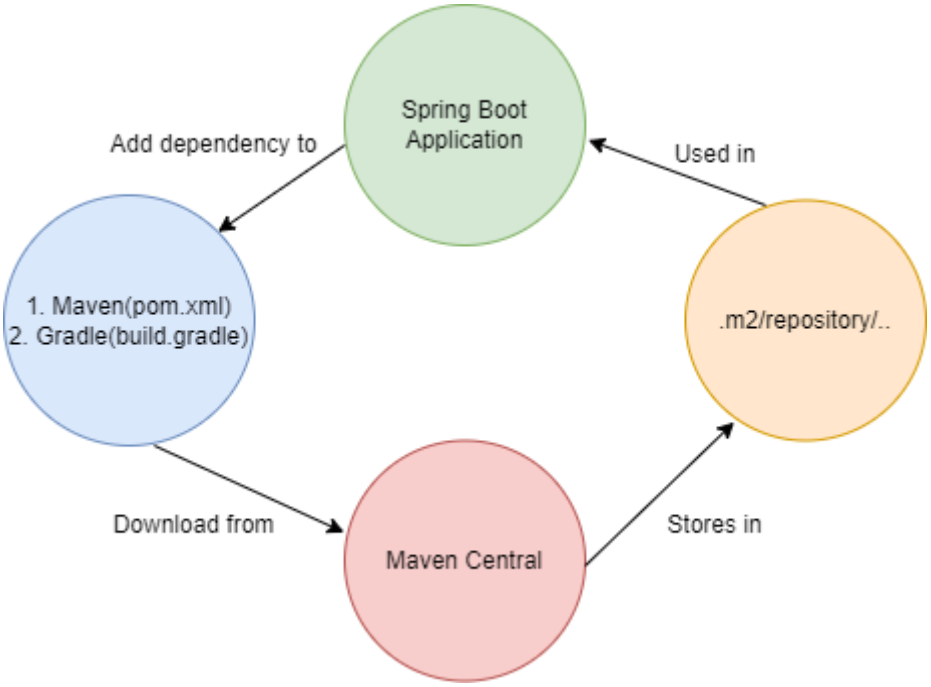


Figure 6.1.21 Flow of adding dependencies

The below are the dependencies that I have added in this project:

a. Spring Security

```
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Figure 6.1.22 Spring Security

Spring Security is a strong and incredibly configurable framework for access control and authentication. In terms of protecting Spring-based apps, it is the de-facto standard. Spring Security offers a number of security features. Authentication and authorization, for instance, are used to create secure Java Enterprise applications. Authentication and authorization are Spring framework's two main applications. Authentication is a process of identifying and knowing the target user who wants to access the application. As opposed to authorization,

which is the process of grating the authority permission to carry specific application tasks. A web request, methods, and access to a certain domain can also all be authorised.

The following are some of the key benefits of Spring Security:

- Protection from assaults like clickjacking and session fixation.
- Integration of Spring MVC.
- Java Configuration support.
- Integration of the Servlet API Portable.
- Defend against brute force attacks.

The following are the key features of Spring Security:

- LDAP

  LDAP is referred to as Lightweight Directory Access Protocol. With the help of Internet Protocol, it is an open application protocol that intends to manage and access dispersed directory information services(IP).

- JAAS Login Module

  JAAS is referred to as Java Authentication and Authorization Service. The Spring Security Framework employs this pluggable module, which was developed in Java, for authentication purposes.

- Single Sign-On

  This allows the user to access multiple applications with a single login (username and password).

- Web Form Authentication

  This web form collects and authenticates user credentials from the web browser using Spring Security. The Spring Security supports so as to implement the web form authentication.

- Authorisation

  Its goal is to grant the user access to the resources by authorising them. The developers can then specify the access policy in relation to the valuable resources in this way.

- Software Localisation

  It enables developers to create the user interface in any language.

- HTTP Authorisation

  The HTTP authorisation of requests made by web URLs utilising Apache Ant paths or RE is supported by the Spring Framework.

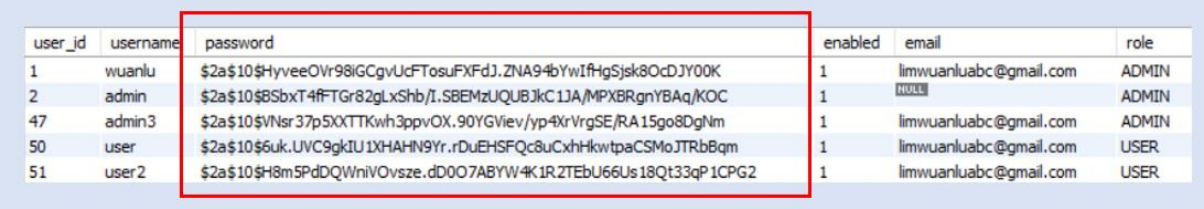
In this project, I have utilized Spring Security in three ways which are:

- Role-based Authentication

  To grant permission to users based on their roles. For example, if user is a normal user, then the user will not have access to some of the modules which are only admin is allowed to access.

- Password Encoder

  To encode password upon account is created for security purposes. For example, the password stored in database is in encrypted format:



user_id	username	password	enabled	email	role
1	wuanlu	$2a$10$HyveeOVr98iGCgvUcFTosuFXFdJ.ZNA94bYwIfHgSjsk8OcDJY00K	1	limwuanluabc@gmail.com	ADMIN
2	admin	$2a$10$BSbxT4fFTGr82gLxShb/I.SBEMzUQUBJkC1JA/MPXBRgnYBAq/KOC	1	NULL	ADMIN
47	admin3	$2a$10$VNsr37p5XXTTKwh3ppvOX.90YGViev/yp4XrVrgSE/RA15go8DgNm	1	limwuanluabc@gmail.com	ADMIN
50	user	$2a$10$6uk.UVC9gkIU1XHAHN9Yr.rDuEHSFQc8uCxhHkwtpaCSMoJTRbBqm	1	limwuanluabc@gmail.com	USER
51	user2	$2a$10$H8m5PdDQWniVOvsze.dD0O7ABYW4K1R2TEbU66Us18Qt33qP1CPG2	1	limwuanluabc@gmail.com	USER

Figure 6.1.23 Password Encoder

- User Session

  To get user session by username in order to retrieve their information. The way to get the user session is by adding the following code in Controller as shown below:

```
    @GetMapping("/viewActivity")
    public String viewActivityPage(Model model) {
        final String currentUserName = SecurityContextHolder.getContext().getAuthentication().getName();
        model.addAttribute("activity", activityService.getActivity(currentUserName));

        return "activity";
    }
}
```

Figure 6.1.24 User session

b. MySQL Connector

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
```

Figure 6.1.25 MySQL Connector

MySQL connector serves as a link between the MySQL server and applications created in different programming languages, including Java, C#, Python, Node JS, etc. A piece of software known as the connector delivers API implementations and an interface for running MySQL queries on the server instance. MySQL provides standards-based drivers for JDBC, ODBC, and .Net allowing developers to build database applications in their language of choice. Additionally, a native C library enables developers to incorporate MySQL right into their projects.

c. Data JPA

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Figure 6.1.26 Data JPA

Spring Boot JPA is a Java specification for managing relational data in Java applications. It enables developer to access and persist data between Java object or class and relational database. JPA follows Object-Relation Mapping(ORM). It's a collection of interfaces. For executing queries and transactions on the objects against the database, it additionally offers a runtime Entity Manager API. It employs the object-oriented query language JPQL(Java Persistent Query Language), which is platform independent.

d. Validation

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
```

Figure 6.1.27 Validation

The Spring Validation is used to restrict the input provided by the user. When Spring Boot encounters an argument marked with the @Valid annotation, it bootstraps the Hibernate Validator, the default JSR 380 implementation, and validates the parameter. A MethodArgumentNotValidException exception is raised by Spring Boot when the target parameter is found to be invalid. The Bean Validation API is a Java specification which is used to provide restrictions to object models. Here, length, a number, a regular expression, etc. can all be validated. In addition, developer can also offer personalised validations.

Some frequently used validation annotations are as follows:

Annotation	Description
@NotNull	Determine that the value can't be null
@Min	Determine that the number must be equal or greater than the specified value
@Max	Determine that the number must be equal or less than the specified value
@Size	Determine that the size must be equal to the specified value
@Pattern	Determine that the sequence follows the specific regular expression

e. Email

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-mail</artifactId>
</dependency>
```

Figure 6.1.28 Email

Using the JavaMailSender interface, the Spring Framework offers a simple abstraction for sending email, and Spring Boot offers auto-configuration and a start-up module for it.

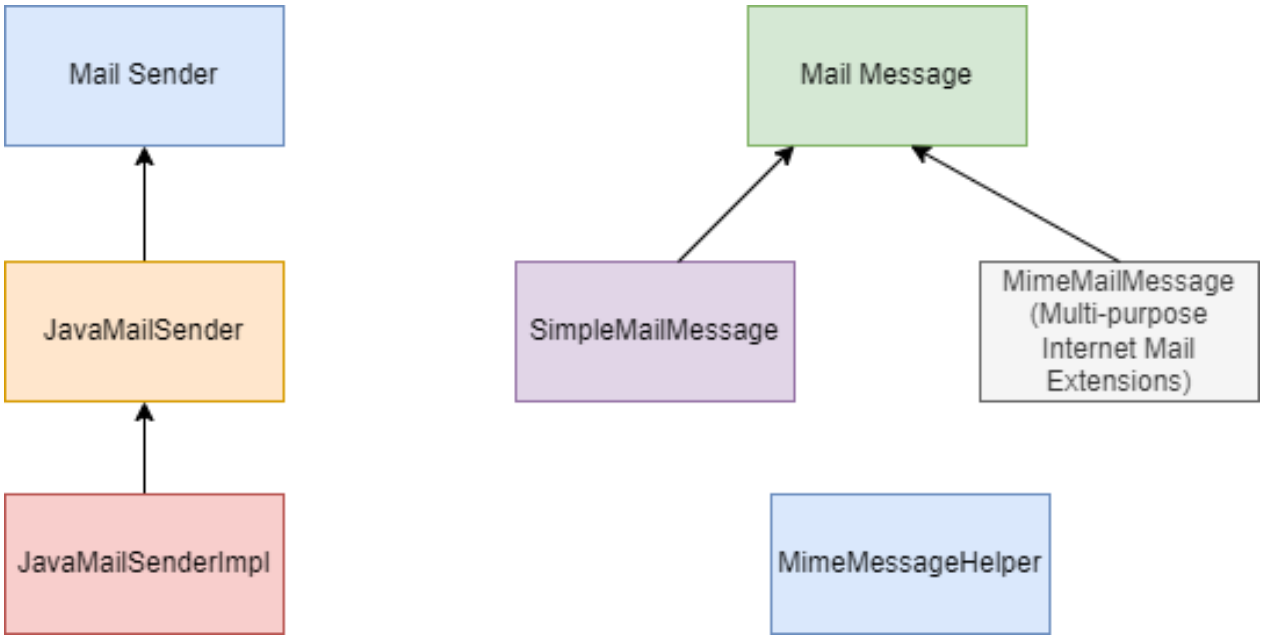


Figure 6.1.29 Spring Mail: Classes/Interfaces Hierarchy

f. File Upload

```
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>
```

Figure 6.1.30 File Upload

This dependency provides common configuration attributes and processing capabilities for multipart requests, uses a Map of Spring CommonsMultipartFile instances as a representation for uploaded files and a String-based parameter Map as a representation for uploaded form fields.

g. Thymeleaf

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
    <groupId>nz.net.ultraq.thymeleaf</groupId>
    <artifactId>thymeleaf-layout-dialect</artifactId>
</dependency>
<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>
```

Figure 6.1.31 Thymeleaf

Thymeleaf is a template engine for HTML5, XHTML, and XML. It is a servlet-based server-side Java template engine that may be used in both web-based and offline settings. It is ideal for current HTML5 JVM web development. Moreover, it also offers total Spring Framework integration. Besides, to display data or text generated by the application, a set of transformations are applied to template files. It can be used to serve XHTML and HTML5 in web applications too.

Thymeleaf's mission is to offer a stylish and organised method of building templates. It is based on XML attributes and tags. These tags specify how predetermined logic is to be executed on the DOM(Document Object Model), as opposed to being directly written as code inside the template. It also serves as a JSP substitute.

Also, because of the caching of parsed files, Thymeleaf's design enables the quick processing of templates. It executes with the fewest number of I/O operations the least possible.

Application properties

```
# DATASOURCE (DataSourceAutoConfiguration & DataSourceProperties)
spring.datasource.url=jdbc:mysql://localhost:3306/dms?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false
spring.datasource.username=root
spring.datasource.password=010903Lim...

# Hibernate

# The SQL dialect makes Hibernate generate better SQL for the chosen database
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect

# Hibernate ddl auto (create, create-drop, validate, update)
spring.jpa.hibernate.ddl-auto = update

logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type=TRACE


## MULTIPART (MultipartProperties)
# Enable multipart uploads
spring.servlet.multipart.enabled=true
# Threshold after which files are written to disk.
spring.servlet.multipart.file-size-threshold=2KB
# Max file size.
spring.servlet.multipart.max-file-size=200MB
# Max Request Size
spring.servlet.multipart.max-request-size=215MB


##EMAIL
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=limwuanluabc@gmail.com
spring.mail.password=utiqjsnpqldmuyhu
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

Figure 6.1.32 Application Properties

As is already known, Spring Boot is constructed on top of the spring and includes all of the Spring's functions. And is quickly becoming a favourite among developers these days because it gives them the opportunity to concentrate on the logic without having to worry about configuration or setup. When developer create a new Spring Boot application in spring starter or inside an IDE(Eclipse or STS), a file called application.properties is created in the src/main/resources folder.

As a result, the application.properties file is utilised in a spring boot application to store application-related properties. This file provides the many configurations needed to run the application in various environments, each of which will define a separate set of properties. We define every sort of property, such as changing the port, database connectivity, connection to the server, and many others, inside the application properties file.

For examples:

1. To change Port Number

   Sometimes when developer run their spring application, they may encounter error such as the port 8999 was already in use. In this case, developer may kill that process that is running on this port number, or they may change the port number and rerun the application:

```
server.port=8989
```

Figure 6.1.33 Server Port

2. To define name of application

   This can be done by writing the properties like this:

```
spring.application.name = userservice
```

Figure 6.1.34 Name of Application

3. Connecting with MySQL database

```
# DATASOURCE (DataSourceAutoConfiguration & DataSourceProperties)
spring.datasource.url=jdbc:mysql://localhost:3306/dms?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false
spring.datasource.username=root
spring.datasource.password=010903Lim...

```

Figure 6.1.35 MySQL Connection

4. Setting for Multipart Properties

```
## MULTIPART (MultipartProperties)
# Enable multipart uploads
spring.servlet.multipart.enabled=true
# Threshold after which files are written to disk.
spring.servlet.multipart.file-size-threshold=2KB
# Max file size.
spring.servlet.multipart.max-file-size=200MB
# Max Request Size
spring.servlet.multipart.max-request-size=215MB
```

Figure 6.1.36 Multipart Properties

5. Setting for Email

```
##EMAIL
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=limwuanluabc@gmail.com
spring.mail.password=utiqjsnpqldmuyhu
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

Figure 6.1.37 Email Properties

6. Hibernate

```
# Hibernate

# The SQL dialect makes Hibernate generate better SQL for the chosen database
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect

# Hibernate ddl auto (create, create-drop, validate, update)
spring.jpa.hibernate.ddl-auto = update

logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type=TRACE
```

Figure 6.1.38 Hibernate Properties

6.2 DISCUSSION

Based on the results as shown in 6.1 Results, Spring Boot framework able to help developer in solving the issues that developers usually faced during development of applications.

Firstly, everything is automatically configured with Spring Boot. Appropriate configurations must be used to take advantage of certain features. For example, if developer wants to use Hibernate(ORM), then developer can just add @Table annotations above model/entity class and add @Column annotation to map it to table and columns in the database. This allows developers to avoid heavy configuration of XML and avoid humans error when doing manual configuration.

In addition, after choosing a suitable starter package, Spring Boot will try to automatically configure your Spring application based on the jar dependencies you added. For example, adding Spring-boot-starter-web will cause Spring Boot to automatically configure registered beans such as DispatcherServlet, ResourceHandlers, and MessageSource. Also, with spring-boot-starter-jdbc, Spring Boot automatically registers DataSource, EntityManagerFactory, and TransactionManager beans and reads database connection information from the application.properties file. If developer don't use a database and don't provide manual connection details, Spring Boot will automatically configure a database in memory without any additional configuration. Automatic configuration can always be completely overridden using user settings.

Next, Spring Boot works well with several servlet containers. For example, Spring Boot uses Tomcat by default, but developer can easily switch to Jetty, Undertow, Resin, and Wildfly. Developer can also choose options that improve the specific types of functionality that interest them most. Just as importantly, Spring Boot automatically identifies which servlet developer set as the new default during the start-up sequence. These Spring Boot benefits give developer the flexibility to choose the embedded server that best suits their needs.

Besides, Spring Boot's in-memory database and embedded server (Tomcat) reduce or eliminate boilerplate code that typically required to set up an application. Without a lot of boilerplate code, development teams can reduce development time and updates cycles, resulting in more satisfied users and more productive employees. This is another advantage of Spring Boot that helps developers save time.

Other than that, WAR files are not required in Spring Boot. However, Spring Boot can use WAR(Web Application Resource) files, but it is not required. Instead, Spring Boot can rely on JARs(Java resources). The JAR has a shorter and simpler structure that is convenient for developers and users. Lightweight files work quickly to connect the application with the tools it needs to function. The option to use either WAR or JAR also benefits development teams. For example, if someone in the team has no JAR experience, they can fall back on WAR. It may have a subtle impact on speed, but it helps developers get their products to market as quickly as possible.

Also, Spring Boot able to speed up the dependency management process. Spring Boot implicitly packages the third-party dependencies required for any kind of Spring-based

application, with so-called starter packages (spring-boot-starter-web, spring-boot-starter-data-jpa, etc). Starter packages are collections of handy dependency descriptors that developer can include in their applications. They allow developer to get a universal solution for all Spring-related technologies without having to search for code examples or load the required dependency descriptors from there. For example, if use Spring Data JPA to access database, just add the spring-boot-starter-data-jpa dependency to the project. This allows developer to save time as they no need to find compatible Hibernate database drivers and libraries.

Last but not least, Spring Boot makes it easy to develop Spring-based apps using Java. Its opinionated approach to the Spring Framework frees up time to focus on building and testing apps, with less time spent on decision-making and repetitive tasks. Also, Spring Boot apps integrate seamlessly with other projects in the Spring Framework ecosystem such as Spring Data, Spring Cloud, Spring Security, and other trusted cloud services such as Microsoft Azure Spring Cloud. Spring Boot also provides plugins and tools to simplify development. For example, Spring Boot offers plugins to be able to work with in-memory databases, as well as other popular build automation tools such as Apache Maven. Additionally, Spring Boot's command line interface(CLI) tools and built-in HTTP server make it very easy to create an environment for developing/testing Spring-based applications.

6.3 SUSTAINABILITY

Environmental

- Bootstrapping saves memory space

Boot Initializer is used by Spring Boot to build the source language. Users can reduce the amount of storage needed on their devices and speed up programme loading thanks to this bootstrapping strategy.

By reducing memory consumption in devices, the hardware used by user will be decreased and hence less hardware will be produced. This able to contribute to environmental sustainability as less memory space is being used to develop an application. Especially in this modern era, more and more mobile and web applications are evolving, meaning more memory would be consumed. By bootstrapping strategy utilized by Spring Boot, environmental sustainability can be realized.

Economy

- Decreased boilerplate code

  Boilerplate code generally required to set up an application is reduced or eliminated by Spring Boot's in-memory database and embedded server(Tomcat). Without a lot of boilerplate code, development teams may speed up their update cycles and development timeframes, increasing customer satisfaction and boosting employee productivity. It's yet another benefit of Spring Boot that aids in time savings for developers.

- Simple setup and management

  Spring Boot's key features which are automatic configuration and ease of dependency management may reduce the cost for maintenance and hiring of professionals to set up the configurations.

In short, company can achieve economy sustainability as Spring Boot can speeds up the development and deployment processes by utilizing logical default settings for unit and integration tests. Additionally, Spring Boot saves developers time and energy by enabling them to create strong apps with clear and secure setups without having to spend a lot of time and energy learning the nuances of Spring.

Social

- Easy to follow up

  As the project structure and code structure of Spring Boot are simple, new joiner of the development team can easily catch up with the framework, and hence achieve social sustainability. Additionally, the ease of dependency management also allows new joiner to manage it clearly and easily.

- Opinionated approach

  An opinionated approach holds the view that there is one method that is much simpler than the others. The software restricts designers by default and encourages them to follow the instructions. It offers a clearly defined path and a best practise that will be effective for most people in most circumstances. These industry best practises and accepted norms were strictly followed when writing the app. With an opinionated

approach, it's much simpler to collaborate and acquire assistance for coding projects because other developers who are familiar with the framework will be able to contribute right away.

7.0 CONCLUSION AND RECOMMENDATION

To sum up, I managed to adopt Spring Boot Framework on developing a Document Management System within the given timeframe. Throughout the implementation of this project, I have a clearer and better understanding in the process of developing a system from scratch. Starting from planning phase, research and analysis phase, development phase, testing and troubleshooting phase, until maintenance phase of this project. Each of these phases are carried out smoothly as a project timeframe is being planned at the very beginning of this project implementation. Also, the results produced have met the expectations and achieved the objectives of this project.

Throughout the implementation of this project, there are some recommendations were observed. Firstly would be the performance of Thymeleaf. During the testing phase of system, slow performance of the web page is observed as the page keep loading when navigate from one page to another page. Hence, suggest to use other technology such as JSP instead of Thymeleaf. Next, one of the enhancement to be included would be adding confirmation message for user when they perform actions such as delete, update, and so on. This is to allow user to double confirm their actions that they want to perform. Other than that, a function which is to allow user to change their password after their account is being created as the password is given by admin. This is to ensure user's account is secured and only user themselves know their own password. In short, there are still some improvements and enhancements that should always be implemented to make sure the system is functioning properly and even updated to a more advanced system.

8.0 REFERENCES AND CITATIONS

Anastasia K. (July 4, 2021). *Pros and Cons of Using Spring Boot.* bamboo agile.

Retrieved from https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/

Rahul G. (Aug 26, 2019). *Introduction to Spring Boot*. GeeksforGeeks.

Retrieved from https://www.geeksforgeeks.org/introduction-to-spring-boot/

Azure. (n.d.). *What is Java Spring Boot*.

Retrieved from https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-java-spring-boot/

AmiyaRanjanRout. (Dec 15, 2021). *Spring Boot – Application Properties*. GeeksforGeeks.

Retrieved from https://www.geeksforgeeks.org/spring-boot-application-properties/#:~:text=So%20in%20a%20spring%20boot,different%20property%20defined%20by%20it.

Priya R. (Jan 10, 2020). *Spring vs Spring Boot: A Comparison of These Java Frameworks.*

Quick Code. Retrieved from https://medium.com/quick-code/spring-vs-spring-boot-a-comparison-of-these-java-frameworks-14a1b594657#:~:text=How%20Does%20Spring%20Boot%20Solve,is%20in%20the%20class%20path.

Chantelle L. (Nov 22, 2019). *7 challenges in web application design and development.* Tiller.

Retrieved from https://tillerdigital.com/blog/7-challenges-in-web-application-development/

Turnitin Similarity