



CS6P05 Project

**Work Capacity Management System to Facilitate Self Organisation of
a Scrum**

Project Report

Final Submission

Name: Lauren Spruce
ID Number: 18011848
Date: 12/04/21
First Supervisor: Dr Quan Dang
Second Supervisor: Dr Tingkai Wang

Declaration

Module: CS6P05

Deadline: 3pm Monday 24th January 2021

Module Leader: Dr. Quan Dang

Student ID: 18011848

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

Student signature QD Date: 24/05/21

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations* on Cheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples..."

- (i) Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.
- (ii) Falsifying data in experimental results.
- (iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
- (iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.
- (v) Collusion to present joint work as the work solely of one individual.
- (vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.
- (vii) Other conduct calculated to secure an advantage on assessment.
- (viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.
2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. " $e = mc^2$ (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Abstract

Working from home is an essential part of lockdown life, with many companies opting for online collaboration. This is to ensure the safety of their staff and the public, while still functioning as usual. *“Almost overnight, the office has become a digital space rather than a physical place,”* (Saran, 2020) All office work is preferred to take place online, as the pandemic continues, there is no predicted timeline when office work will return to normal. *“Nearly three-quarters of IT decision-makers (72%) believe employees will not want to return to office working.”* (Saran, 2020) Citrix suggests, further concluding there is high demand for innovative software which can store and handle all a business's infrastructure.

Plenty of software which has already been developed with this purpose in mind, many well known to the public because of the pandemic. Zoom allows groups of people to video call while presenting, this is aimed at schools and small teams. Microsoft teams allows for presenting within groups and private messaging, becoming popular amongst Universities and companies. More specialised software like Trello have been created, this allows businesses to assign tasks to individual team members, however it does not have a concrete messaging system. This leads to multiple application's being used, creating confusion and complicating workflow.

One part of technology that has been affected severely is software development. Normally meetings are held in an informal manner, like brainstorming and team meetings. Usually, these activities would take place in an office environment, however, restrictions like lack of equipment and no access to vital files put a strain on scrum teams' productivity.

My project explores these different applications, their uses, and benefits, as well as highlighting the need for improvement within this growing market. My goal will be to streamline them into one application. Furthermore, I will be demonstrating and implementing an easy-to-use web-based application, which would include all the popular functions, such as easy communication, calling and presenting, but suitable and specialised towards scrum teams, with emphasis on file sharing and goal tracking. This is to allow a scrum team a simple way for effective communication and collaboration, also a place where all files associated with their project are stored and easily accessible to the whole team.

List of Tables

Table 1: Feature comparisons between software management tools	31
Table 2: Cost comparison between software management tools.....	31
Table 3: Software main feature comparison table	39
Table 4: Software's overall costs and practicability.....	39
Table 5: Comparison of communication software's state of the art technologies	44
Table 6: Use Case: Register	55
Table 7: Use Case: View Website	56
Table 8: Use Case: Login	57
Table 9: Use Case: Update Credentials	58
Table 10: Use Case: Logout	59
Table 11: Use Case: Assign Roles	60
Table 12: Use Case: Assign Tasks	61
Table 13: Use Case: File Sharing	62
Table 14: Use Case: Messaging	63
Table 15: Use Case: Create Discussion.....	64
Table 16: Functional Requirements	65
Table 17: Non-Functional Requirements	65
Table 18: Extra Functionality	66
Table 19: User Table	82
Table 20: Task Table.....	82
Table 21: Messages Table	82
Table 22: Comments Table	83
Table 23: Discussions Table	83
Table 24: Logs Table.....	83
Table 25: Files Table.....	84
Table 26: Private Messages Table	84
Table 27: Discussion Members Table	84
Table 28: List of requirements for application.....	123
Table 29: Scrum applications desired functionalities	124
Table 30: Features included within QuickScrum	125
Table 31: Features included within nTask	125
Table 32: Features included in VivifyScrum.....	126
Table 33: Use case tools.....	127
Table 34: Review of report structure.....	137
Table 35: Amendments to Original WBS plan.....	138

List of Figures

Figure 1: nTask: Task UI.....	32
Figure 2: nTask: Main UI	32
Figure 3: VivifyScrum: Main page UI.....	33
Figure 4: VivifyScrum: Project UI	33
Figure 5: VivifyScrum: Main Board UI.....	34
Figure 6: QuickScrum: Main dashboard UI	34
Figure 7: QuickScrum: Backlog UI	35
Figure 8: QuickScrum: Planning UI.....	35
Figure 9: Trello main UI design	40
Figure 10: Trello Task Board Overview UI.....	40
Figure 11: Slacks main UI design.....	41
Figure 12: Zoom main UI design	42
Figure 13: Microsoft teams main UI design	43
Figure 14: Microsoft teams: Chat UI	43
Figure 15: Use Case Model for my software.....	52
Figure 16: Use Case: View Website.....	53
Figure 17: Use Case: Register.....	53
Figure 18: Use Case: Login	53
Figure 19: Use Case: Logout.....	53
Figure 20: Use Case: Update Credentials.....	53
Figure 21: Use Case: Join Discussion.....	53
Figure 22: Use Case: Assign Roles	54
Figure 23: Use Case: Assign Tasks	54
Figure 24: Use Case: File Sharing	54
Figure 25: Use Case: Messaging.....	54
Figure 26: Use Case: Create Discussion	54
Figure 27: Navigation Bar design	68
Figure 28: Right hand side of Navigation Bar design	68
Figure 29: Homepage design	69
Figure 30: Initial Homepage design	69
Figure 31: Final Login page design	70
Figure 32: Login Page Initial Design	70
Figure 33: Final Register page design.....	71
Figure 34: Initial Register Page Design.....	71
Figure 35: Final Discussion Page Design	72
Figure 36: Discussion Page Design	72
Figure 37: Final Discussion List Design.....	73
Figure 38: Initial Discussion List Design	73
Figure 39: Final Private Message Design.....	74
Figure 40: Initial Private Message Design	74
Figure 41: Initial Call Design.....	75
Figure 42: Final Video Call Design	75
Figure 43: Final About Us Design	76
Figure 44: Initial About Us Design.....	76
Figure 45: Final Contact Us Design	77

Figure 46: Initial Contact Us Design	77
Figure 47: Final Account Detail Design	78
Figure 48: Initial Account Details Design	78
Figure 49: Final All tasks Design	79
Figure 50: Initial All Tasks Design	79
Figure 51: ER Relationship diagram	80
Figure 52: Data Model diagram of my database	81
Figure 53: Project Folder Structure.....	85
Figure 54: System Master software architecture	86
Figure 55: Auth folder with security folder and files	87
Figure 56: User folder with all files	87
Figure 57: DAO folder with all files within	87
Figure 58: Discussion folder	88
Figure 59: Discussion folder	88
Figure 60: Private messages folder	88
Figure 61: Store folder	88
Figure 62: Task folder.....	89
Figure 63: Uploader folder.....	89
Figure 64: App Master Software Architecture.....	90
Figure 65: Source JavaScript folders and files for each webpage.....	91
Figure 66: MySQL: Setting up new connection	94
Figure 67: Properties file #1.....	95
Figure 68: Properties file #2.....	95
Figure 69: Properties file #3.....	95
Figure 70: MySQL: Creating new schema	96
Figure 71: MySQL: Executing MySQL Script	96
Figure 72: Running application	97
Figure 73: Starting Java Spring system master	97
Figure 74: npm install on local 1	98
Figure 75: npm install running	98
Figure 76: Successful execution of npm install	98
Figure 77: Executing cd frontend and npm start	99
Figure 78: Login page UI.....	99
Figure 79: Registration page UI.....	101
Figure 80: Successful registration	101
Figure 81: MySQL user's table	102
Figure 82: Password validation	102
Figure 83: Field validation	103
Figure 84: Email validation.....	104
Figure 85: Login UI	105
Figure 86: Username and password validation	106
Figure 87: Private message UI.....	106
Figure 88: Private message file attachment view	107
Figure 89: Discussions UI	107
Figure 90: Discussion List	108
Figure 91: Join discussion #1.....	108
Figure 92: Join discussion #2.....	109

Figure 93: Sending messages in discussions	110
Figure 94: Discussion members	110
Figure 95: Deleting the discussion	111
Figure 96: User view of discussion	111
Figure 97: Task UI.....	112
Figure 98: Adding tasks	112
Figure 99: Task table UI.....	113
Figure 100: Assigning a task.....	113
Figure 101: Marking a task as completed	114
Figure 102: Completed tasks.....	114
Figure 103: Private Message UI with private video calling.....	115
Figure 104: Private video call UI	115
Figure 105: User details	116
Figure 106: MySQL table view	116
Figure 107: MySQL: User data	117
Figure 108: MySQL: Changing user to admin.....	117
Figure 109: MySQL: Confirming applied changes #1	117
Figure 110: MySQL: Confirming applied changes #2	118
Figure 111: Updated role for user.....	118
Figure 112: Account details UI	119
Figure 113: Account details Password validation	119
Figure 114: Deleting account	120
Figure 115: MySQL: Deleting account #1.....	120
Figure 116: Deleting account confirmation dialogue message	121
Figure 117: MySQL: Deleting account #2.....	121
Figure 118: Original Project Plan	135
Figure 119: Updated and Amended Gantt Chart.....	144

Table of Contents

Declaration.....	2
Abstract.....	3
List of Tables	4
List of Figures	5
Chapter 1: Introduction	12
1.1 Project Topic and Rationale	12
1.2 Project Aims and Objectives	13
1.3 Methodology.....	15
1.4 The report Structure	16
Chapter 2: Background Research.....	18
2.1 Literature Review	18
2.1.1 Defining Software Engineering	18
2.1.2 Processes within Software Engineering.....	19
2.1.3 Activities vital for Software Engineering.....	20
2.1.4 Example of positive use of Software Engineering.....	21
2.1.5 Example of a failure of Software Engineering.....	22
2.1.6 Defining Agile Methodology	23
2.1.7 Defining Extreme Programming (XP)	23
2.1.8 Defining Adaptive Software Development (ASD)	24
2.1.9 Defining Scrum	25
2.2 Critical Evaluation of Related Products	28
2.2.1 Scrum Management Systems	28
2.2.1.1 Examples of Scrum Management Tools.....	29
2.2.1.2 nTask	29
2.2.1.3 VivifyScrum	30
2.2.1.4 QuickScrum	31
2.2.1.5 Comparison Tables.....	31
2.2.1.5 Scrum Management Interfaces.....	32
2.2.1.6 nTask UI.....	32
2.2.1.7 VivifyScrum UI.....	33
2.2.1.8 QuickScrum	34
2.2.3 Examples of Communication Web Applications	36
2.2.3.1 Trello	37
2.2.3.2 Slack	37

2.2.3.3 Zoom	38
2.2.3.4 Microsoft Teams	38
2.2.3.5 Comparison Tables.....	39
2.2.3.6 Trello UI Design	40
2.2.3.7 Slack UI Design	41
2.2.3.8 Zoom UI Design	42
2.2.3.9 Teams UI Design.....	43
2.2.10 State of the Art Technologies Used	44
2.3 The Scope of the Project	45
2.4 Review of development platforms and tools.....	46
2.4.1 Technology Selection Criteria	46
2.5 Summary of changes implemented during development period.....	49
Chapter 3: Requirement Analysis	51
3.1 Use Case Diagram	52
3.2 High-Level Use Cases.....	53
3.3 Use Case Descriptions.....	55
3.4 Functional Requirements.....	65
3.5 Non-Functional Requirements.....	65
3.6 Extra Functionality	66
Chapter 4: Software Design	68
4.1.1 Navigation Bar.....	68
4.1.2 Homepage	69
4.1.3 Login Page Design	70
4.1.4 Register Page Design.....	71
4.1.5 Discussion Page Design	72
4.1.6 Discussion List Design	73
4.1.7 Private Message Design	74
4.1.8 Call Design.....	75
4.1.9 About Us Design.....	76
4.1.10 Contact Us Design	77
4.1.11 Account Details Design	78
4.1.12 All Tasks Design.....	79
4.2.1 Database Design.....	80
4.2.2 Entity Relationship Diagram.....	80
4.2.3 Data Model Diagram	81

4.2.4 Data Dictionaries.....	82
4.3 Overall Program Architecture	85
4.4 Project Folder Structure.....	85
Application Folders	86
4.5 Scrum Management System Master.....	86
4.6 Scrum Management App Master.....	90
Chapter 5: Implementation of the application	93
5.1 Minimum System Requirements	93
5.2 Software System Requirements.....	93
5.3 Installing.....	94
5.4 Running the Program	94
5.4.1 Configuring MySQL Database	94
5.4.2 Configuring Software	97
Chapter 6: Testing the Program	101
6.1 Registration.....	101
6.1.1 Password and Confirming password not matching	102
6.1.2 Leaving a field blank.....	103
6.1.3 Incorrect email format	104
6.2 Login.....	105
6.2.1 Incorrect Username or Password	106
6.3 Private Messages	106
6.4 Discussions	107
6.4.1 Deleting a discussion.....	111
6.5 Tasks.....	112
6.6 Private Video Calling	115
6.7 Roles.....	115
6.7.1 Changing role to admin.....	116
6.8 Account Details	119
6.8.1 Entering password incorrectly	119
6.8.2 Deleting Account.....	120
Chapter 7: Evaluation of Results.....	123
7.1 Checklist of requirements	123
7.2 Comparison of results to Case Studies.....	124
7.2.1 QuickScrum	124
7.2.2 nTask	125

7.2.3 VivifyScrum	126
7.3 Use Case Tools	127
Chapter 8: Conclusions.....	129
8.1 A Summary of what has been achieved within the project.....	129
8.2 Reflections of lessons learned	131
8.3 Legal, Social, Ethical and Professional Issues.....	132
8.3.1 Data Protection Regarding Project	132
8.4 Future Work.....	133
Short Term Goals	133
Long Term Goals	133
Appendices.....	135
1.1 Original Project Plan.....	135
1.2 Review of Project Progress	136
1.3 Amendments to the Original Plan of Project.....	138
1.4 Lessons Learnt in Project Management.....	139
1.5 Survey Responses.....	140
1.5 New Project Gantt Chart.....	144
References	145
Bibliography	148

Chapter 1: Introduction

1.1 Project Topic and Rationale

Due to the pandemic Covid-19, many businesses have transferred all their work online, which brings in the demand for software's specifically designed for each team's needs.

Being able to communicate effectively and efficiently is a huge part of working as a team. In this project, its sole purpose is to eliminate any obstacles which occur with a self-organisation of Scrum, by creating a prototype of a system which will increase work productivity.

Many smaller scrum teams will often use impractical ways of sharing and distributing content between the team. Sometimes these notes are kept individually, which can lead to confusion on goals set, thereby decreasing the work efficiency on the whole project itself. As scrum teams have daily meetings, this system would be used to keep all members up to date, as in a real-life scenario some members might not attend every meeting. Scrum leaders will have a direct platform where they can contact every member of the team together, therefore everyone is kept up to date. By having a system where each member is given a clear role and goals, this will therefore increase the work efficiency and overall work productivity.

This topic is interesting to me because while working on a group coursework in year 2, we implemented a scrum team to complete our project. I found that whilst scrum is effective in quickly producing work, sometimes team members would often get confused on what task had been completed, this was because we did not have an official software to share our progress, we were using a WhatsApp group. I also noticed on WhatsApp that sharing files and links to individual progress often got lost within the chat. Leading to decreased collaboration and work productivity. Therefore, I believe my management system will enhance scrum team co-operation, as this gives the team a stable platform to share their progress, keeping informed.

To conclude, this project combines all my knowledge from the last 2 years, providing challenge, further displaying all my software development skills for a successful completion of a functioning and ambitious application.

1.2 Project Aims and Objectives

Within this project, the main aims will be to eliminate any misinformation in a scrum team. This is needed as some team members might be more updated than others, due to some not attending the daily meetings. To overcome this, the main objective will be to allow scrum leaders to create a discussion page, so the team is constantly updated, increasing productivity. This has been completed by using IntelliJ to implement all code. New discussion pages will also be implemented, allowing scrum teams to appropriately organise all the processes within their current project.

Another aim that was met, to successfully implement tasks. This was crucial, as it eliminates any need to use another software to assign roles for different user's tasks. From within the application, all aspects and needs for a scrum team are covered, due to the successful completion of implementing all original listed features, applying extra functionality where needed.

Video calls, and the ability to message individuals privately was successfully implemented, allowing easy communication. This eliminated the need for other software to be used for this purpose, for example, "*Microsoft teams*" (Mircosoft, 2021). However, a challenge faced was that it was hard to implement was group calling, as it requires a lot of knowledge and coding. The key reason video calling was implemented was to increase workflow, to allow for more engaging discussions, rather than all team communication to be completed primarily over messaging. Private messaging was implemented, as sometimes team members may only have a concern regarding one other team member, or perhaps a scrum leader requires to speak to only one team member. This may need privacy, so therefore privacy is crucial within any communication application.

Discussions have also been implemented, this is where Scrum teams can effectively collaborate, this eliminates the inconsistency of having multiple team conversations split between various Scrum software. Scrum teams will also have the option to create multiple discussions, the purpose of this is to allow different aspects of a project's progression to be recorded and discussed. Therefore, it allows for better organisation and convenience, rather than filtering through one discussion to find relevant information. Finally, file attachments can be sent to all members of a discussion. This allows for all vital information to be shared within a convenient software.

Another aim is to improve communication between team members and the scrum leader. a role system will be implemented; this will allow scrum leaders to assign roles to different team members. Therefore, allowing them to access role-based discussions. This will improve communication as each team member is fully aware of their responsibilities within the project. To complete this, MySQL will be implemented, to create a database storing the user's login details and roles. This will also include permissions, allowing login functionality.

File transferring will also be a main feature of the application, ensuring team members an easy convenient way to access all important files. This feature is implemented in both private and group discussions, ensuring consistency and freedom to collaborate further.

Another aim is to review the challenges of scrum, this is to ensure the application is effective and up to date with current scrum needs. To achieve this, various reports and case studies will

be reviewed. Therefore, allowing thorough analyse the issues with current scrum practice, therefore creating a more efficient prototype for teams to use.

To conclude, this application will allow Scrum masters an organised platform, where they are able to manage efficiently and easily. Ultimately, this will decrease complexities of introducing added software into their work environment.

Now, a summary of all basic functions will be listed, these project aims were promised to be implemented from the very start of this project:

- Login functionality
- Discussion pages, creating and deleting
- Meeting calls
- Role assignment
- Private messaging
- File transferring and sharing

Here is a summary of all completed objectives hit after the successful completion of the project:

- A fully deployed web application with backend database
- A login and register page which allows users to access the application, user authentication
- Creation of tasks which users can assign and mark as complete
- Table of tasks available to user, can switch between completed and not completed
- Ability to private message other users registered to the web application
- Ability to create and join and leave discussions and message within them, ability to send attachments
- List of discussions all users are a part of
- Attachments able to be sent within private messages and within discussions
- Logout feature
- Ability to change account details and permanently delete accounts
- Ability to set roles consisting of admin and regular user
- Allow for 2 users' to privately video call one another

1.3 Methodology

Firstly, vital background research was performed into the purpose of scrum, analysing what qualities will be needed for a useful software for scrum teams to utilise. This allowed for successful merging into their current infrastructure. Secondly, evaluating the benefits, and drawbacks, of current on the market scrum tools available. This was performed to help fully explore all popular features currently available on the market. Examples of these are "*Trello*," (*Trello*, 2021) and "*Slack*" (*Slack*, 2021). Therefore, this allows for successful integration of popular features, while adding extra purposeful additions to the application.

This project will be following Agile methodology and the Unified Software Development Process Model, to allow a structured development life cycle. This will aid in keeping on track with the overall designing and development of prototype and final project, using requirement analysis, design, implementation, and testing.

From this, designing the prototype was more convenient and effective option for scrum teams, making sure the final design is attractive and user friendly. IntelliJ was used, to implement all code, adding features like instant messaging, drawing platform for graphs and MySQL. Throughout this process, reviewal and cleaning the code making sure the project is on track, implementing all the needed features. Also, throughout the project, constant documentation occurred, to allow convenient tracking of all current and past progression. Another reason to ensure this happened, was to keep updated regarding the project's needs. Therefore, allowing for an increased motivation and workflow throughout.

1.4 The report Structure

To outline the entirety of the final report, this chapter shares an overview of the project topic. Within aims and objectives shared previous goals of features to be implemented, while providing the completed features that were implemented. This confirmed that all the project aims, and objectives were satisfied completely, that the project had been successfully completed. Methodology was outlined, this showed how the project was carried out.

Within chapter 2, extensive background research was carried out, on the market applications, these are currently providing solutions to scrum teams. By analysing the current and additional features offered, allowed to provide a solution to missing vital components within my projects aims. Therefore, this allowed to use suitable practices to implement the best solution to build a similar product, solving all issues with current software.

Continuing with the Unified Software Development Process Model, the next chapter involves requirements analysis and specification. This investigates all the functional and non-functional requirements for the project. This chapter will also explore the use case techniques, modelling appropriate models to further establish the prototype for elicitation.

Chapter 4 is where showcasing is documenting on the completed software design, using elaboration, implementing the user interface designs, database design and the overall architecture of the software.

Chapter 5 shares an overview of the summary of the implementation and testing that had been completed, testing allowed to completely analyse all functions and aspects of the application. Therefore, allowing to make final improvements and fix any bugs or errors which had been encountered. Within chapter 5 details are shared on the minimum specs, needed to run the application, including software requirements. Not to mention, full details are also provided to help run and set up the application for the first time.

Within chapter 6, this is where evaluation and conclusions made from the completed application, reflecting on work finished, also how the application could have also been improved. Reflection is crucial, as it gives motivation to continue background research and possibly develop a similar program for future work.

To conclude, chapter 7 details all conclusions made to the final projects overall progress, giving an achievement summary, including lessons learnt and reflection. Future work is also introduced here, due to the success of this project, plans for further development and research are undergoing.

Chapter 2: Background Research

2.1 Literature Review

2.1.1 Defining Software Engineering

"Software has become deeply embedded in virtually every aspect of our lives," (Pressman, 2019) this means that software is constantly being upgraded for many different purposes. When software is being developed, many different opinions and views must be taken into consideration during development. This is to ensure that it meets all the critical conditions and features needed to make it successful for the chosen audience. "*Large teams of people now create computer programs that were once built by a single individual.*" (Pressman, 2019) This is due to the higher demand of the software; this is also to ensure the software is developed and produced to its highest standards. These standards have changed as the complexity of the technology has advanced dramatically throughout the last decade, all businesses and transactions run on some sort of software, this can range from "*consumer electronics to medical devices to weapons.*" (Pressman, 2019) If these sorts of systems fail, it can lead to all sorts of negative effects, like private information being stolen or the collapse of a whole system due to a breach.

Majority of individuals in this timeframe, depend and rely on software for many of their daily needs and tasks, ranging from work on an organisation's software, to being used on their personal devices, mobile phones, and PCs. This also leads to the attention and drive for development teams to follow suitable software engineering methods to keep the software secure, and the target audience satisfied.

Furthermore, software can gain popularity over a range of multiple user's using it. This means that the software needs to be expandable as well as maintainable, it is crucial that the software's infrastructure and overall design is carefully created, for future expansions. Development teams will also desire to keep the user's content, so therefore upgrades and the implementation of new features all depend on quality software engineering techniques. This reinforces the ideas that software engineering plays an important role when developing and designing software.

Fritz Bauer gives this critical definition of software engineering as follows, "Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines." (Pressman, 2019) this felt like a good way to describe software engineering, as realistically, companies want to reduce their overall costs to gain stronger profits. This all needs to be done efficiently, as losses contribute to profit reduction, which needs to be avoided if possible. Software also needs to be efficiently, therefore, it does not use up lots of resources to make it easy to run. It also needs to be able to run on most machines its target audience, increasing the desirability of integrating on computers and mobile phones to increase the target audience and use across many devices.

However, this definition does not define the overall meaning of software engineering, it fails to include the consumers satisfaction or the overall quality. Therefore, "IEEE," (Pressman, 2019) have created a more inclusive definition, which goes as follows, "*the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*" (Pressman, 2019) This definition includes the fact that the approach to developing any software must be done carefully, with a lot of consideration into the maintenance, keeping the system running and stable, to the operation of carrying out the planned development efficiently.

Overall, software engineering provides the basic procedures to create efficient and maintainable software. This can be done with the methods being used, the tools being used such as reports and data models, and overall milestones needed to reach to complete a quality product.

2.1.2 Processes within Software Engineering

There are many processes which contribute to the success of developing any software. These are as follows:

- **Initial Planning**

First, any team will need to come together and create the ‘blueprints’ of the software. Many questions need to be discussed. Who is the target audience? What are the main features that need to deliver? *“Planning activity creates a “map” that helps guide the team as it makes the journey.”* (Pressman, 2019) This allows the team to analyse all the factors involved, including the risks that could occur while creating the software, the plan of the development moving forward, and any additional requirements needed to produce it.

- **Modelling of the software**

Models are needed for the team to truly understand the architecture of the software. It also allows them to grasp the critical small details required. Each part of the software needs to be linked together and fit, therefore modelling is an important process to include. This is also important if the software is being developed for a target audience, these models can be shown, this should be done throughout the process in case any adjustments need to be made, this also makes the development much more efficient as costs can be cut if problems are encountered earlier on.

- **Communicating as a team**

This process is crucial as it can allow the team to communicate with the stakeholders helping produce the piece of software. *“The intent is to understand stakeholders’ objectives for the project.”* (Pressman, 2019) This also allows the team to understand what resources they will need to develop the software. This also helps the team communicate with each other, all team members should be able to understand the importance of the features needed within the development. This avoids confusion later along the developments timescale.

- **Overall development**

This process includes the main construction within the software, all team members will work together to produce the code. All features will be implemented, this will be closely monitored and assessed to make sure it is following the initial planning of the software. If any adjustments have been added to the plan, the programmers will update and tweak the code to stay on track.

- **Testing of the software**

Before the deployment of the program can begin, the code must be finalised, testing is an important stage as it must be done on all parts of the software, this is to uncover any bugs or errors within the code. This is also an opportunity for the stakeholders to test the software, if any issues arise, then it can be fixed within this stage. The development team might also ask for many target audience individuals to come forward and give their feedback on the software, the more people that participate within this stage, the more stabilised and prepared the software will be for its final deployment.

- **Deployment**

The software has finally been deployed, its initial deployment will allow users and stakeholders to use it and give back their thoughts and feedback. Constructive criticism can be given in this stage, allowing the development team to consider all feedback and develop a strategy to complete any final adjustments needed.

These main processes are used and applied to many software developments, as all software needs to develop with these processes and thought processes in mind. This can be done for small scale applications to large applications needed for important business infrastructure.

2.1.3 Activities vital for Software Engineering

Now, definition will be carried out, this includes the vital activity's teams will participate in, to develop successful software. These are as follows:

- **Understanding the risks**

Before any development can take place, the team must come together and list all the risks associated with the project. This is done so that each team member is aware, therefore plans can be made to overcome them. This leads to better preparation and efficiency to develop as quickly and safely as possible.

- **Tracking project**

Tracking allows the team to keep track of the progress of the project, team leaders are also able to instruct certain parts of the team to work harder if the project is falling behind schedule. Without this plan, certain features of the software could be completely missed out or forgotten about, therefore it is essential to keep track of all the different aspects.

- **Quality checking**

The team must constantly check if the quality of the software is what the stakeholders require, this must be done regularly to ensure the that the project is on track and meets all expectations needed.

- **Reusability**

This is an important activity to complete as this allows the team to use parts of the software within future projects, or perhaps use certain parts of the code within other sections of the software, this increases the efficiency of the development, as it reduces the construction of the code.

- **Initial preparation and planning**

Activities within the preparation should include such as many “*models, documents, logs, forms, and lists.*” (Pressman, 2019) The better planning put into any software, the better the overall deployment will entail. This is due to all the thought put into the software, all team members and stakeholders agree to the infrastructure as the use of models and lists will provide evidence for the foundation of the overall software. Logs are also useful as they can show all the work put in by many of the team members. This also leads to better overall tracking of the software, one of the previous activities required.

2.1.4 Example of positive use of Software Engineering

"The Success Story of Instagram" (Faraz, 2020)

Instagram is an online social media platform, with the original concept of allowing users to share and give feedback on photo uploads. The reason Instagram has been used as an example of positive use of software engineering is mainly due to the developers keeping this vision constant and present during their time developing the software. Originally, Instagram's name was called 'burbn,' (Faraz, 2020) This was due to the original intent of the software, sharing pictures of favourite alcoholic drinks, being able to tag the location of where to purchase them. The original founder was "*a 27-year-old man by the name of Kevin Systrom.*" (Faraz, 2020) He originally did not have much coding knowledge, due to hard work and learning how to, he successfully created Burbn on the "*6th of October 2010.*" (Faraz, 2020)

Systrom showcased his prototype of the software to potential stakeholders, the stakeholder's loved the vision of the software and the features which were implemented. This therefore "*led him to raise \$500,000 in seed funding from the venture capitalists.*" (Faraz, 2020) Due to such positive attention, he quit his job and began working on Burbn full time. He founded a development team which shared the same ideas for the software and motivation to see it succeed, therefore they were to release it a short time after.

Systrom started off by planning the software, he looked at requirement analysis models from other successful social media platforms, like Facebook. Then he modelled the software to truly understand the architecture of the software. It also showed him all the features he wanted to implement, allowing him to grasp the critical small details required. Therefore, each part of the software was linked together and fit, this was something he needed to ensure to the stakeholders.

Another good method of software engineering was the fact that Systrom allowed various users to test his software before it is launch when it was in its artefact stage, this means that the artefact could be improved upon before it is final release. It is important to note that the development team constantly took the user's and stakeholder's needs into consideration. Therefore, all the features that were desirable got implemented and therefore this increased the satisfaction gaining even more attention.

Burbn gained a lot of attention after its initial deployment, so much that the software crashed. Because of this, Systrom and his development team began to prepare for the future of the software, to this day they are constantly implementing more features and functions to make the software the best it could possibly be to satisfy the target audience. "*At this point, burbn pivoted to become more focused on photo-sharing with likes and comments.*" (Faraz, 2020) Doing this, they also renamed Burbn to Instagram, this name was chosen as it is a "*combination of instant camera and telegram.*" (Faraz, 2020)

To this day, Instagram is one of the most popular social media platforms, in 2020, "*active reported users have held steady around 1 billion people.*" (Chen, 2020) This shows that following software engineering processes and activities, including good communication with stakeholders, means that you are giving your software the biggest advantages in order of it to be globally successful.

2.1.5 Example of a failure of Software Engineering

"Abandoned NHS IT system has cost £10bn so far" (Syal, 2013)

Perhaps one of the biggest software engineering fails in the UK would be the NHS software, which ended up costing £10 billion pounds to date. Originally, "*the project was launched in 2002,*" (Syal, 2013) the software originally was developing well, however, due to the scope of the project changing, the development team did not prepare properly, lack of planning lead to the project becoming over budget, they also never planned this properly. The project continued, this led to the development team discovering technical difficulties they could not overcome. Due to the project being in full development, it ended up getting abandoned as the budget went over so much, they no longer had the dedication to complete it. Due to lack of planning, the project was many years behind schedule, if they had applied a methodology to it, like Scrum, it may have changed the outcome. This is because Scrum follows a strict routine of including daily scrum meetings to track the progress of the project. Also, Scrum and agile methodologies like ASD and XP rely heavily on feedback from the stakeholders, the initial project plan's risks are always assessed from the initial participation from the team. This means that the team would have been ready to adjust when they realised, they did not have the resources to complete the project effectively, the costs would have also been cut considerably. The stakeholders and the development team could have made negotiations to adapt the software to their resources, saving a lot of money and wasted time.

Due to the initial project failing, the government concluded to "*salvage something from the failure said they would keep the component parts in place with separate management and accountability structures.*" (Syal, 2013) They tried to recycle some of the software, which was originally created, however in their future efforts even more failures were present and discovered.

"MPs also expressed surprise that the government's estimates of future costs have not taken into account the termination of contracts with another major IT supplier, Fujitsu." (Syal, 2013) This shows that the development team and the government were not communicating properly throughout the duration of the project, therefore the government cut ties with important suppliers which were helping the distribution of the software. There was no clear goal in mind, so therefore the government did not understand the actual resources which were needed to complete the software.

Modelling the software was ignored by both the development team and the stakeholder, so therefore they could not practically plan for all features which needed to implement, increasing the overall budget.

Throughout the whole development, the team failed to get users to test the artefact, this is proven as the stakeholder's did not have much involvement at all during any of the stages. This also meant that the artefact was not true to the stakeholder's overall vision of features.

To conclude, this very example shows the significance of software engineering, the principles of good communication and planning should not be ignored in the early planning stages, as it leads to an uncontrollable amount of expenses gained, if the risks are not considered and assessed from an earlier stage, this leads to the project's infrastructure to be unchangeable and inefficient.

2.1.6 Defining Agile Methodology

Agile methodology is well known for being the modern solution to software process. “*An agile team is a nimble team able to appropriately respond to changes,*” (Pressman, 2019) this showcases that within a project, nothing is certain.

There are many factors which are constantly changing, and unplanned or unforeseen circumstances can arise, such as team member changes and technology advancements, which can impact on how you develop a project. Therefore, strong support needs to be established within the software’s development life cycle. “*An agile team recognizes that software is developed by individuals working in teams and that the skills of these people, their ability to collaborate is at the core for the success of the project.*” (Pressman, 2019) Agile teams recognise that collaboration is the strongest factor of software development, due to the highly competitive market, their strategies and software will need constant adjustments to satisfy the projected audience.

Agile teams understand and recognise that the process of creating a software is very unpredictable, as customer needs may alter over time, also design, implementation and testing will also alter considerably after the initial plan. There to save resources and time, “*customer feedback is an operational prototype or a portion of an operational system,*” (Pressman, 2019) this is an important principle as change is always welcomed within agile methodology. “*Agile use iterative and incremental approach to develop a software product.*” (Hafidz & Sensuse, 2019)

In the industry, there are multiple agile process models, these are:

- **XP (Extreme Programming)** (Beck, 2012)
- **Adaptive Software Development (ASD)** (Highsmith, 2013)
- **Scrum** (Sutherland & Schwaber, 2020)

This research will evaluate the very core concepts of Extreme programming, Adaptive Software Development and Scrum, firstly to understand the concepts embedded in many development teams, to further help establish a strong understanding of key features needed within a management software. This also reinforces the key morals and principles needed to produce successful software.

2.1.7 Defining Extreme Programming (XP)

“*XP is a style of software development focusing on excellent application of programming techniques, clear communication, and teamwork.*” (Beck, 2012) Extreme programming was developed by Kent Beck. It became one of the first agile methodologies developed. “*XP was the dominant agile method in the late 90s and early 00s before Scrum became dominant.*” (Fowler, 2013) The foundations of ASD and scrum are based on the foundations of XP, as its main principles are to focus on communicating effectively and working together as a team. Ways to tell XP apart from other methodologies are as follows:

- ✓ It is incredibly flexible, the development teams or the stakeholders’ plans, and ideas might change within the development cycle, so therefore it can easily be adapted.
- ✓ XP has short and concise intervals within the development, these can be defined as, “*short development cycles, resulting in early, concrete, and continuing feedback.*” (Beck, 2012)

- ✓ As the plan of the development is expected to change over time, this means that XP features an “*incremental planning approach*,” (Beck, 2012) which allows for changes to be made earlier on within each stage of development.
- ✓ Communication is key within XP, team members must be constantly engaging within meetings between the team, also team leaders must partake in constant engagement from the stakeholder to ensure the quality and features of the software are being achieved.
- ✓ It also relies on “*an evolutionary design process that lasts as long as the system lasts.*” (Beck, 2012) The design must be flexible and adaptive to any changes; therefore, any adjustments can be updated efficiently.

This project will be using many of the principles from Extreme Programming, setting each task in short intervals will help plan out overall software more efficiently, allowing to spot any errors earlier on in code. Also, the fact that they use an initial structure for the software, this technique will be replicated, as there may be a few features which may update further down the line. Overall, XP methodology has been successful and provided many developments the template to create powerful software, which can be adapted and recycled for multiple purposes.

2.1.8 Defining Adaptive Software Development (ASD)

“Project managers John Highsmith and Sam Bayer are credited with inventing the Adaptive Software Development methodology in the early 1990s.” (ProductPlan, 2021) ASD was developed, it replaced an older version which was named Rapid Application Development (RAD). During the development, both Highsmith and Bayer developed over 100 software projects using the ASD methodology, these were documented and analysed in their book, using their “*new approach in Highsmith’s 2000 book, Adaptive Software Development.*” (ProductPlan, 2021)

Like XP methodology, ASD focuses on teams being able to change and adapt to different requirements given by the stakeholders. There are 3 main processes within ASD, these are as follows:

✓ Speculate

This allows all team members to come together and discuss all possibilities and routes needed to take to complete the software’s development. Meetings take place within the teams and stakeholders to ensure all parties understand the risks and resources needed to undertake such a software.

✓ Collaborate

This is where all team members work together to produce their part of the software. Models and diagrams are produced to keep the software’s target goal in constant discussion. Team members will all be assigned their role within the production and given clear goals, so they understand what they need to complete. Also, all risks are revised in this process, and the possibility to change is kept in mind.

✓ Learn

This is where the development team reflect on all their work they have completed, during deployment they will allow the stakeholders and other target audience test their software. They will adapt to any needed changes and learn from any errors or mistakes they may encounter. This is done through team meetings to discuss the issues encountered.

Overall, ASD focuses highly on the stakeholders and the target audience which will be using the software. This therefore “*can lead to better and more intuitive products.*” (ProductPlan, 2021) As there is increased motivation within the team, this allows for the delivery time of the software to meet the stakeholders agreed date, or even earlier. ASD encourages the communication between the team and stakeholders, allowing for more personalised software which meets all the requirements. This resembles the similar nature of XP, where end user’s needs are always taken into consideration. However, the drawbacks of ASD can be that the stakeholder cannot always give constant feedback on the software as they may have other responsibilities, this means that there will always be features that the software team will have to adapt on. Also, testing is completed within every stage of the software, therefore the overall expenses and budget may increase due the constant attention to completing a working software.

ASD will be used as a part of the project as software will constantly be tested for any errors that may occur, also due to the fact software’s goals needed to be followed consistently, which the supervisor will be expecting, referencing back to the Gantt chart and initial project proposal.

2.1.9 Defining Scrum

“*Scrum is a framework for developing software using the agile principle,*” (Hafidz & Sensuse, 2019) Scrum was originally created by Jeff Sutherland in the early 1990s. His definition of scrum is as follows: “*Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.*” (Sutherland & Schwaber, 2020)

It follows the principles of agile, its framework consists of:

- **Requirements**

This is where the team will plan out the initial requirements and compare this to their resources and experience, they have collectively. This will be constantly in discussion with the stakeholders to make sure the vison of the project stays true to their expectations.

- **Analysis**

This is where the team will compare their software to current data models, to ensure successful data models are used in reference, also to reinforce any risks that they might occur, as other development teams might have experienced the same.

- **Design**

This is where the team will plan the design and draw up models and initial drafts of the UI. Each team member will have their own part to play and participate in. This will be reviewed within the daily scrum meetings to ensure everything is on track.

- **Evolution**

The team will carry out testing on any artefact completed so far, they will involve both users and stakeholders within this process, to ensure that any feedback and criticisms can be adapted. The software will be evolved to suit the expectations.

- **Delivery**

This is where the final product will be deployed. Throughout this stage, testing is still be carried out to ensure that the software meets all requirements and expectations have been carried out. The

stakeholders will also give their final feedback, they will also plan with the development team the next stages involved with the software and how to update it if it is successful.

Here, listing all the events within scrum will be documented. These involve:

- **The Sprint**

A sprint's main purpose is to be carried out within a short timeframe, these can be made to be completed within a month or less. Throughout scrum, each sprint will be taken into effect immediately after the previous one is finalised and polished. The goal of the product is taken into consideration within every sprint, which includes, "*Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.*" (Sutherland & Schwaber, 2020) During each sprint, it is vital that the overall quality does not decrease, this could lead to variety between many of the components. Also, the backlog of the product needs to be reviewed and used as reference within any process of the sprint. Also, "*Scope may be clarified and renegotiated with the Product Owner as more is learned.*" (Sutherland & Schwaber, 2020) This allows for the team to be updated and change and necessary features.

- **Planning of a sprint**

Planning is crucial to make sure that the sprint is conducted efficiently and effectively. All the scrum team plays a part within the planning, to ensure everyone's ideas are discussed and issues erased. There are certain topics which are discussed throughout the planning, one example is "*Why is this Sprint valuable?*" (Sutherland & Schwaber, 2020)

- **Daily Scrum meetings**

Meetings usually take place within 15 minutes daily. "*To reduce complexity, it is held at the same time and place every working day of the Sprint.*" (Sutherland & Schwaber, 2020) The overall purpose of the sprint meetings is to give team members an insight on the progression completed so far, the scrum masters can also give feedback to everyone's progression.

- **Review Sprints**

Reviewing the sprints is important as it allows for feedback from all team members and the scrum leader. This feedback can be adapted to other scrum's, team members can adapt and overcome any issues raised and effectively eliminate any. This also gives the scrum team time to plan what they must achieve for their next sprint. "*The Sprint Review is a working session, and the Scrum Team should avoid limiting it to a presentation.*" (Sutherland & Schwaber, 2020) A less professional approach is better for the review as it allows everyone the chance to contribute and give ideas for future.

When implemented in an industry setting, a scrum team consists of the customer, scrum master and the development team. The scrum master's main purpose is to keep the team motivated and productive during the daily meetings. Also, to communicate with the customer regularly to make sure all goals are being met, any changes are also being implemented. Scrum teams use 'Sprints' to show iterations within the framework, this mainly consists of planning and team members ideas. "*User story used in scrum board as a card with task description or idea,*" (Hafidz & Sensuse, 2019) this further enforces scrum values and ideas the team will have to produce.

Agile methodology is flexible, designed to handle changes throughout the project's timeline, scrum is based on constant and direct engagement and communication between team members and the customer. The challenges of scrum consist of factors such as "*separate location, modification from*

other software developers, scalability for a large project, and the need for high-quality software.” (Hafidz & Sensuse, 2019) This enforces the importance of an appropriate software needed to complete scrum methodology to its highest standards, also what will be aimed for within the project.

Overall, scrum allows for teams to complete software's efficiently and on time with the stakeholder's deadline. This is a reoccurring theme through Scrum, ASD and XP methodologies, these three principles will be applied to the project as the deadline dates need to be met, throughout the course of this module. Short sprints within Scrum are something which will also be applied, as changes might be made to the software's core features implemented. This is like ASD and XP, short increments allow teams to make any changes earlier on within the project. Scrum is also effective due to team members efforts, these are always recognised within scrum meetings, so therefore the work each team member produces will be to a higher quality and more polished. In scrum meetings, any issues can be raised early on within the progress of the software, so therefore problems are encountered earlier and will not amount to a critical error within the final deployment.

However, because scrum is so focused on communication within scrum meetings, if team members are not committed to the software, the flow of the development can be reduced, not being as efficient as the stakeholders expected. Scrum meetings can lead to anger and stress within each team member as they take place daily, so therefore extra motivation and communication to certain individuals to keep them on track within the development. Also, scrum can only be successfully completed if all team members are experienced, due to all the processes and commitment expected from them. As each team member is assigned their own task to work on, if anyone left throughout any stage, this could lead to problems as it would halt the workflow on that specific task. This is also the case with both XP and ASD as they rely on all team members to take on their own individual roles. This can affect large projects more as there are many more complex features within it.

2.2 Critical Evaluation of Related Products

2.2.1 Scrum Management Systems

A management system is crucial tool for a team to manage their associated goals to produce their theoretical software. *“The level of complexity of the system will depend on each organization’s specific context.”* (ISO, n.d.) For a scrum team, the essence of the system will use communication as it is focal point.

DSD, Distributed Software Management has become more popular than ever due to the pandemic and can be defined as *“a scenario in which people working on a software project are geographically and/or temporally distant.”* (Santos, et al., 2018) It is argued that communication is the predominant main factor in a software’s success.

After researching many different team software, such as Trello and popular Zoom, here is a compiled list the basic functions they offer, consisting of:

- **Private and group messaging**
- **Creating and assigning tasks**
- **Audio and video calling**
- **File/Content sharing**
- **Role Assignment**
- **Task Assignment**

The most popular management systems are Trello and Slack. Both offer related features for scrum teams to incorporate into their agile model. Nonetheless, the main arguable difference between the two is that neither one fully embraces all features listed above. Trello is perfect for professionals allowing tasks to be assigned and comments added. However, it lacks a user-friendly interface which is quite complex for novices to understand, it also lacks the function of messaging between peers. Whereas Slack offers a unique and user-friendly interface, with the ability to create discussions and private messaging, however, it does not include group calling on discussions for free, it hides it behind a payment plan, putting off start up teams and non-professionals.

Research took place within more specialised scrum management tools including nTask and VivifyScrum, these were specifically made for scrum. Evaluation and comparison of the software will be discussed. Therefore, finally comparing to popular Communication Web Applications and how they differ.

2.2.1.1 Examples of Scrum Management Tools

While researching software, I have been constantly evaluating the positive and negative features included. In this section, I will be looking at more specialised applications used specifically by scrum teams.

2.2.1.2 nTask

The first example I have come across is ‘nTask.’ With this web application, you can organise many different aspects of a scrum team. This is including:

- **Tasks**
- **Meetings**
- **Timesheets**
- **Risks and issues within the workspace**
- **Third-party integrations**
- **Issue tracking**
- **Project portfolio**
- **Daily scrum management**
- **Team Collaboration**

You can create new projects, manage, and monitor previous ones using a simple UI list of all existing projects. This is similar with the tasks, all tasks are listed, and you can manage all aspects, this is done in real time so therefore other team members with the same last will be informed and updated on any new changes. All meetings are listed for the team members convenience, so therefore they can keep up to date on any upcoming or on-going meetings. Any issues with any tasks can also be recorded for the scrum team leaders to see. Risks can also be listed and organised within categories, these being:

- **Critical**
- **Major**
- **Moderate**

For this software, zoom integration is now a possibility, this will allow team members to communicate through audio and video calls, with the option to share screens. However, I find this to be a problem, within my project, I want to condense everything down in one single application. Therefore, this integration is a problem as it slows down setting up the application for a team to use. Some problems which could arise could be as follows:

- Increases complexity of application
- Team members may not be able to set up correctly
- Important files and meetings in multiple places, defeating the purpose of an ‘all in one’ product

Conclusion

This application focuses mainly on the task aspect of working in a scrum team, making sure all information is shown here and that all users can view their specified tasks. However, there is no

communication between members. To make this application better I think private messaging and group chats are a necessity, this is something I will be taking into consideration with my application. Overall, this application increases workflow and productivity, as well as team collaboration with the use of comments. It uses integration well; however, I feel this application would be more mainstream if it implemented messaging and calling features of its own.

2.2.1.3 VivifyScrum

Another application I researched included VivifyScrum. This software has a sleek and clean UI, keeping all information organised, increasing the workflow for teams. What I found to be a highlight of this software is that it includes free scrum training to any user who signs up, this is a way for the team to recap and reinforce the ideas and methodologies behind scrum. This is also useful for teams transferring over to scrum principles.

The key features of VivifyScrum are as follows:

- **A project calendar with tasks and important dates integrated**
- **Time tracking of the project**
- **Scrum board built in**
- **Roles and permissions**
- **Use of charts**
- **The use of third-party integrations**

I also feel as though this application would benefit greatly from integrating their own messaging and calling and audio features as the team has no way of communicating effectively on this application. This is the overall theme with many scrum management tools, therefore that is why I will be looking over Communication Web Applications.

With this software, you can create and join multiple organisations you are a part of, within these organisations you are able to view all tasks and projects, which also involves management, this is dependent on your permissions.

When creating a new project, you can add boards within it, for all team members to view. This can include active sprints which are on-going, or previous ones. This is good for workflow and efficiency within the team. When creating boards within your organisation, you can include more tasks, these are split up between different boards. When setting up, there are options to use either a scrum layout, which will include default sprints, and the Kanban board, which will be listed as 'do,' 'in-progress,' and finally 'done.'

Conclusion

This application is a great way for scrum teams to organise their work, providing an easy-to-use design simple enough for anyone to use. Like nTask, I feel what this application is missing is the ability to communicate efficiently and effectively. Comments are a given feature; however, this is not enough for the team to stay updated on the little details and changes made.

2.2.1.4 QuickScrum

QuickScrum follows all the basic features of scrum features, like both VivifyScrum and nTask. It provides a simple UI, simple use of dragging and dropping tasks and projects. This allows for easy organisation provided to the scrum team and ease of use. It appears to be the most user-friendly scrum management tool I will evaluate. Compared to nTask, there are certain features that may be difficult to inexperienced team members to use, whereas with QuickScrum it is easy to adapt to this application, meaning the integration of moving all a team's tasks would be a simple and time efficient.

Some of the main features implemented include:

- **Tracking of activities**
- **Tracking to view individual's progress**
- **Drag and drop implemented into the application**
- **Charts**
- **Third party integrations**

Like nTask and VivifyScrum, QuickScrum implements features like boards and allows the use of Zoom within their application. Again, I feel like they should have implemented communication features as this would give them a bigger audience a reason to transfer over to their application.

2.2.1.5 Comparison Tables

Here, I will be comparing all the desirable features against each Scrum management tool. All desirable features are ones I will be implementing within my software.

Table 1: Feature comparisons between software management tools

Software	Private Messaging	Boards	Video Calls	Audio Calls	File/ Content Sharing	Role/Task Assignment
nTask		✓			✓	Both
VivifyScrum		✓			✓	Both
QuickScrum		✓			✓	Both

With this table, I will be comparing all of applications cost against one another. nTask seems the most user friendly with the least amount for a pro version.

Table 2: Cost comparison between software management tools

Software	Cost	Added on Costs	Practicability
nTask	Free	\$1 a month	Standard + Pro
VivifyScrum	Free	\$8 a month	Standard + Premium
QuickScrum	Free	\$3 a month for plan	Standard + Business

2.2.1.5 Scrum Management Interfaces

2.2.1.6 nTask UI

The screenshot displays the nTask Task UI interface. At the top, there are four colored sections: 'Not Started' (orange), 'In Progress' (blue), 'In Review' (yellow), and 'Completed' (green). Each section contains a list of tasks with subtasks and team member assignments. The 'Not Started' section includes a 'Post Production Evaluation' task. The 'Completed' section includes a 'Brief Creative Team' task. A banner at the bottom states 'Releasing Soon - Available on nTask Premium and Business Plans.'

Figure 1: nTask: Task UI

This screenshot shows the different tasks a user can have, including the other team members which are also assigned to that task. For each task, they can be split into subtasks, in this example it shows that the team has added the sections, ‘Not started,’ ‘In progress,’ etc. This shows the usefulness of splitting up each task.

The screenshot displays the nTask Main UI interface. At the top, there is a navigation bar with 'Team Exampleteam' and 'Workspace Project1'. Below the navigation bar, there is a horizontal menu with tabs: Projects, Tasks (which is selected and highlighted in green), Meetings, Timesheet, Issues, Risks, and Settings. Under the 'Tasks' tab, there is a sub-menu with 'List', 'Grid', and 'Calendar' options. The main area shows a table with columns 'ID' and 'Task'. There is a button '+ Add New Task' and a single task entry: ID 000001, Task Task1.

Figure 2: nTask: Main UI

This screenshot shows the main view of a team. There is the ability to add new tasks, as well as the ability to view the different functions of the application, via the navigation bar at the top.

2.2.1.7 VivifyScrum UI

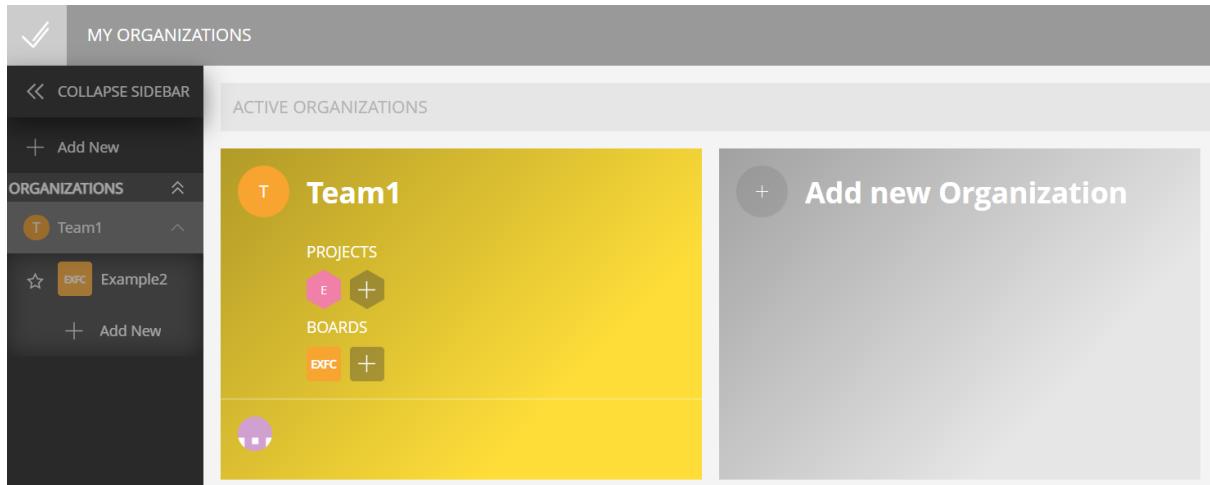


Figure 3: VivifyScrum: Main page UI

This screenshot shows the main UI, this is where all the organisations the user is associated with will be displayed. In each team, the user can create a new organisation, within this they can create new projects and boards.

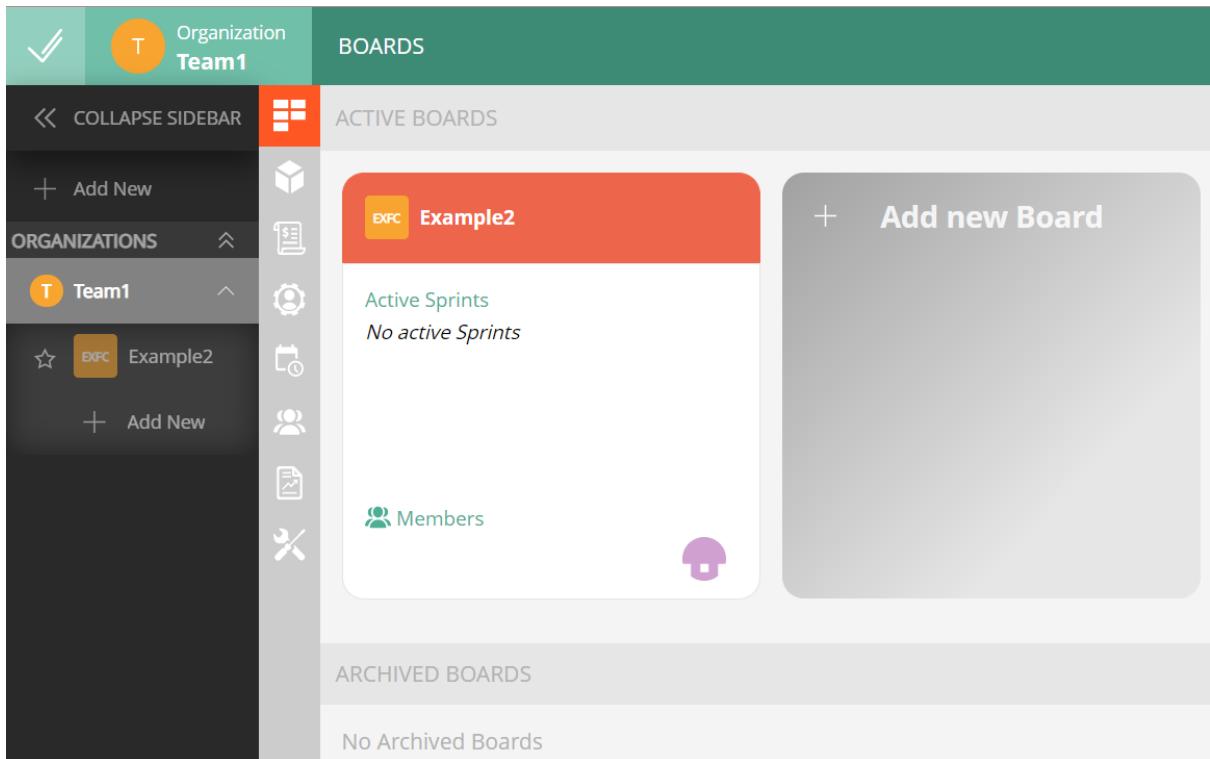


Figure 4: VivifyScrum: Project UI

This screenshot shows the main UI of each individual project. The user can assign different users to the boards, create new boards, and manage.

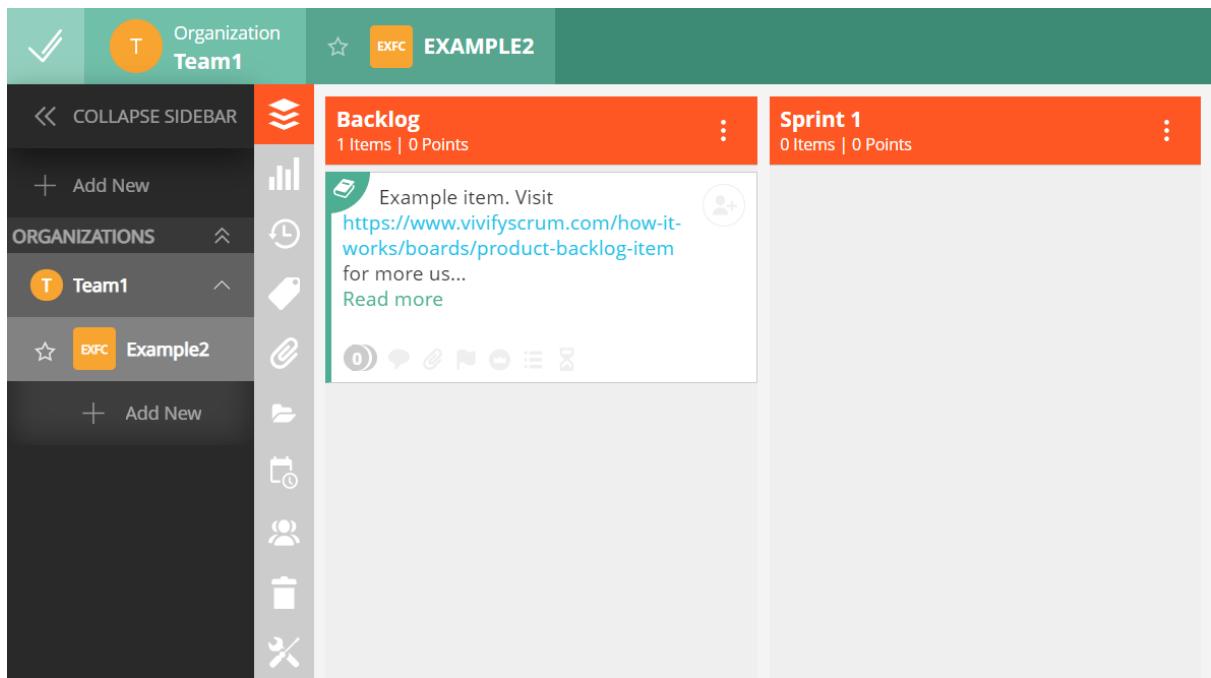


Figure 5: VivifyScrum: Main Board UI

This screenshot shows the boards. Within this example, the user selected a sprint default template, which splits up the board into sprints. The user can add information within these boards and add members onto the specific tasks.

2.2.1.8 QuickScrum

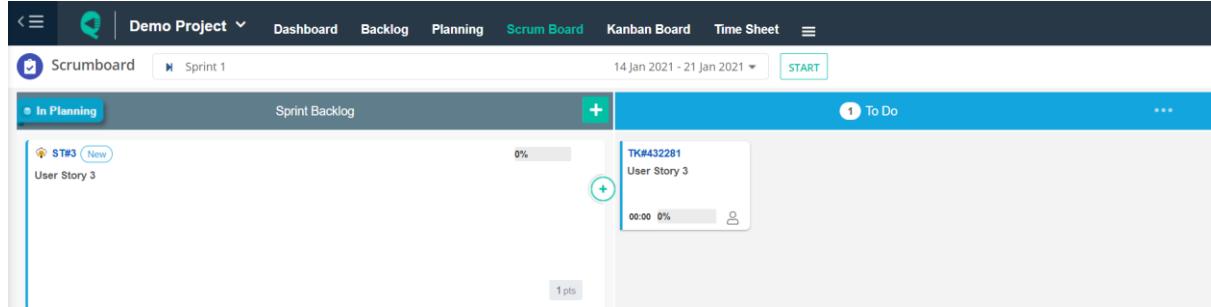


Figure 6: QuickScrum: Main dashboard UI

This screenshot shows the main UI of the user's dashboard. Here they can navigate between the different features using the navigation bar, the scrum board is also the main feature on this page.

The screenshot shows the 'Product Backlog' section of the QuickScrum application. At the top, there's a navigation bar with icons for back, forward, and search, followed by 'Demo Project' with a dropdown, and links for Dashboard, Backlog, Planning, Scrum Board, Kanban Board, Time Sheet, and a menu icon. Below the navigation is a header for 'Product Backlog' with 'Code' and 'Name' sorting options. A table lists four user stories:

	Code	Name
1	ST#3	User Story 3
2	ST#1	User Story 1
3	ST#0	User Story 0

Figure 7: QuickScrum: Backlog UI

This screenshot shows the user's backlog where they can add and manage certain 'stories.' It is simple and easy to read and use.

The screenshot shows the 'Planning' view of the QuickScrum application. At the top, there's a navigation bar with icons for back, forward, and search, followed by 'Demo Project' with a dropdown, and links for Sprint Planning, Planning, Scrum Board, Kanban Board, Time Sheet, and a menu icon. A sub-header indicates 'Sprint Planning' for 'Release1' from '14 Jan 2021 - 21 Jan 2021'. The main area displays a table titled 'Release Backlog items not assigned to any Sprint' with columns for 'Code', 'Name', 'T', 'P', and 'Est.'. A dropdown menu shows 'All Userstories' and a button to '+ ADD STORY'.

Figure 8: QuickScrum: Planning UI

This screenshot shows the planning view of the UI. This is where the user can drag and drop tasks onto the planning to easily organise and collaborate with their team members. This will be updated for all team members.

2.2.3 Examples of Communication Web Applications

Previously we have evaluated the positives and negatives of the two main team management systems, as a result, it is known that both lack all required features of a successful management system. Furthermore, I will be evaluating popular mainstream software's such as Zoom and Microsoft teams.

Zoom is popular because of its integration of use across platforms, "*Our easy, reliable cloud platform for video, voice, content sharing, and chat runs across mobile devices, desktops, telephones, and room systems.*" (Zoom, 2019) This makes it appealing to many businesses and education systems, as it offers user friendly UI, and the flexibility to use on the go.

Microsoft Teams is popular as it is integrated within windows PCs themselves, allowing effortless installation and ease of use across multiple platforms.

Trello and slack are also popular applications as they offer teams slightly more specialised features compared to Zoom and Slack. Slack offers messaging and calling as well as the ability to split up team's goals, however Trello is like the previous Team management tools like nTask, where it focuses primarily on setting tasks and roles to all team members. Furthermore, I will be comparing all the communication web applications.

Here is a recap on the main features I will be continuing to evaluate:

- Private and group messaging
- Creating and assigning tasks
- Audio and video calling
- File/Content sharing
- Role Assignment
- Task Assignment

2.2.3.1 Trello

Trello is used by a lot of software development teams due to the easy-to-understand UI and the easy ability to assign different tasks to team members. All team members can easily collaborate through group discussions and on a private level. File sharing is also featured on Trello, so therefore team members have one application to keep all important files on for sharing.

Some of its main features include:

- Task assignment
- Messaging
- Commenting on tasks
- Ease of organisation with labelling
- File attachments
- Mobile friendly
- Discussions

2.2.3.2 Slack

Slack has a smart and simple UI design, it implements features like creating different channels and messaging users privately and topics, however, the feature to video and audio call is hidden behind a pay wall, apart from that the application is very easy to use and has a sleek UI. Communicating is the main feature within Slack, this is portrayed well within the UI. It was the ability to be used on mobiles, good for teams constantly on the go. Slack also offers integration with services relating to google drive, allowing for a good place for all files to be shared. It is a lot simpler compared to Trello, where goal is to keep teams in collaboration and more of a communication tool, whereas Trello focuses more on tasks.

Main Features include:

- Messaging (privately and within discussions)
- Create new channels for different tasks
- File attachment
- Third party integration
- Audio and video calls

2.2.3.3 Zoom

Zoom is becoming one of the most popular applications for teams, organisations, and the education system to use. This is because of its simplistic design, and important features. Zoom is based purely on video and audio communication, this means that it includes the ability to edit your video settings, like customising the lighting and even changing your background. This is useful for users wanting privacy and peace of mind, as you can hide background objects and retouch your appearance, suitable for a professional work environment. Messages within meetings can be sent, and the user can send messages to other users.

Some important features include:

- Calendar integration
- Waiting room
- Meeting rooms
- Screen sharing
- Video editing software

2.2.3.4 Microsoft Teams

Microsoft Teams like Zoom is also becoming popular with the education system, this is due to the many features implemented. There is also a calendar like zoom, also it has the ability for users to create new teams, which they can invite many people from. Microsoft teams can also be linked with University contacts, meaning that it is easy for other students to look up students, by their name or their ID number. There is the option for messaging user's privately, viewing all the activity within teams, and calling users individually. Teams allows for audio and video calls with the use of content sharing.

Overview of features:

- Assignments
- Private and group messaging
- Private and group audio and video calling
- Content sharing
- File sharing
- Activity history
- Meeting recording

2.2.3.5 Comparison Tables

I will be comparing some of the on-market product's features against each other, to see what the best value for money is between all of the products. This will include all the important features needed within a team management system.

Table 3: Software main feature comparison table

Software	Private Messaging	Discussion pages	Video Calls	Audio Calls	File/ Content Sharing	Role/Task Assignment
Trello			✓	✓	✓	Both
Slack	✓	✓	✓	✓	File sharing	Both
Zoom	Only in meeting		✓	✓	✓	Roles only
Teams	✓	✓	✓	✓	✓	Assignments

I will be comparing the prices of all the products, pricing is important since different teams may have different budgets, small teams prefer to spend the least on necessary software, they also do not need to manage members on a large scale. Whereas large businesses will need to keep track of more advanced statistics, more members so therefore more security and features must be included.

Table 4: Software's overall costs and practicability

Software	Cost	Added on Costs	Practicability
Trello	Free	Business class paid	Standard + Business
Slack	Standard version free	Pay for calls, two paid plans	Standard + Business
Zoom	Free	Pay for large group calling and organisations	Standard + Business
Teams	Free	Organisations need subscription	Standard + Business

2.2.3.6 Trello UI Design

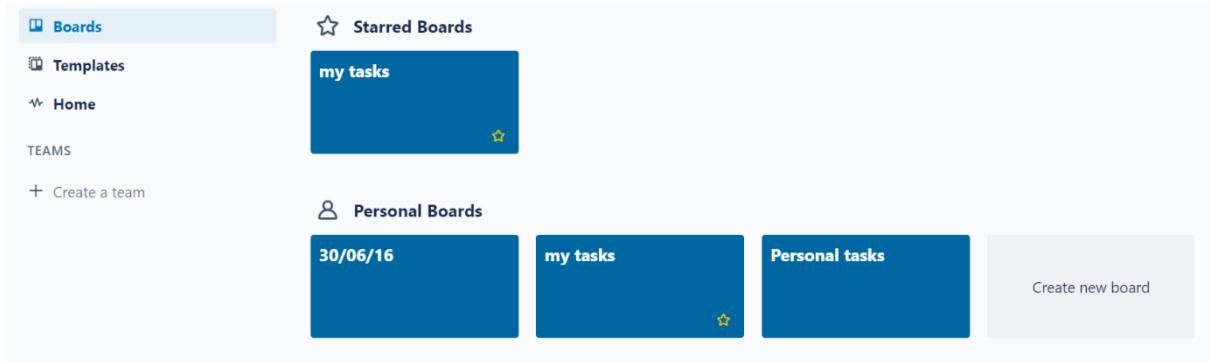


Figure 9: Trello main UI design

This is the main dashboard for Trello, where users can create new personal tasks. Also, the organisation they joined tasks will be displayed in the starred boards. Clicking on each board will take the user to the task page, where they are split up into different boards. Within each board, the user can add tasks.

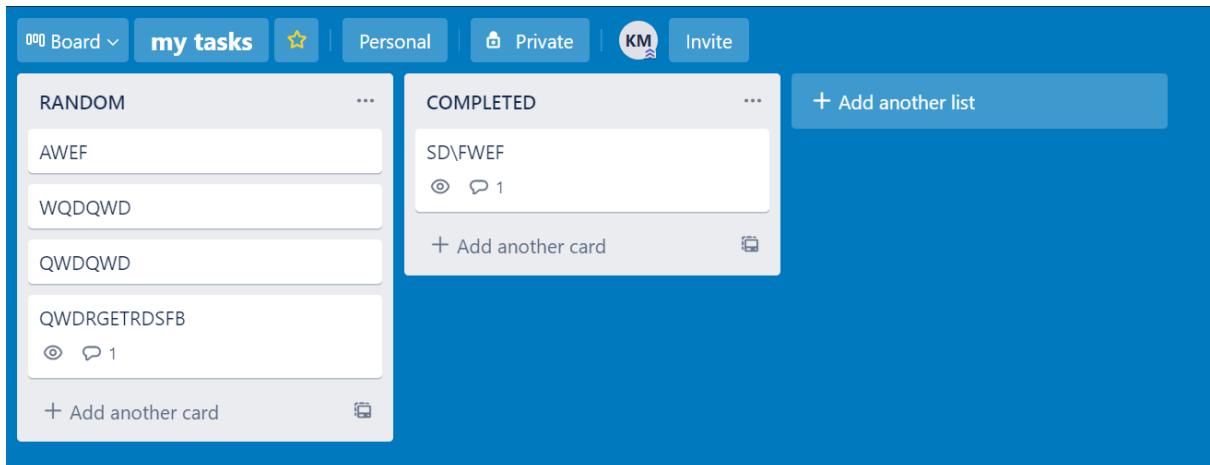


Figure 10: Trello Task Board Overview UI

This is the task board page. Users can create new boards and delete them. They can also be edited, for example, changing the name. Comments can be made by any of the team members, it also can show who has viewed them. More lists can be added within each task, they can also be highlighted for importance.

2.2.3.7 Slack UI Design

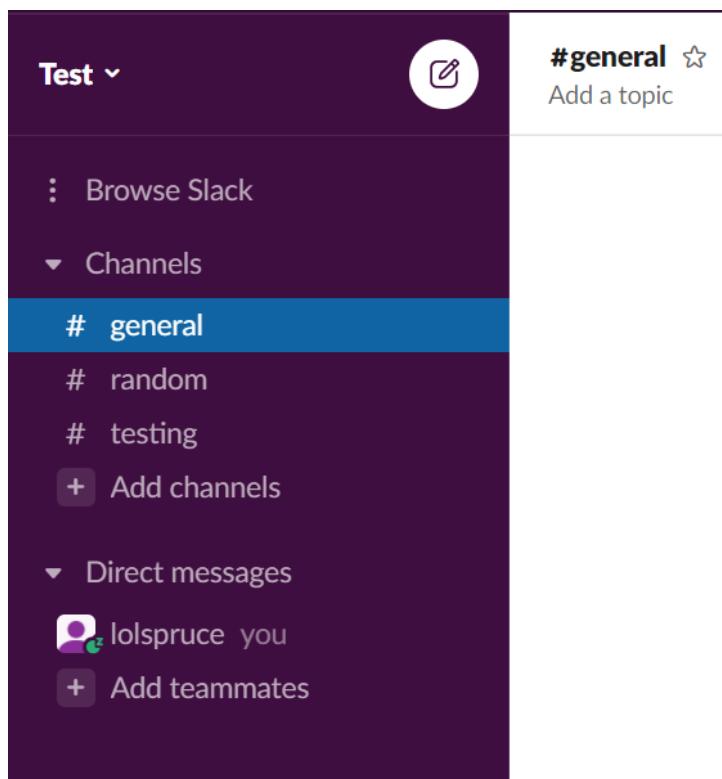


Figure 11: Slacks main UI design

This is the main dashboard for slack. There is a side navigation, consisting of the different channels which the user can create or edit. Direct messages are also displayed here, within each chat and channel, the user can send messages and attachments to other members. There are settings, which allow users to customise their profile picture and name.

2.2.3.8 Zoom UI Design

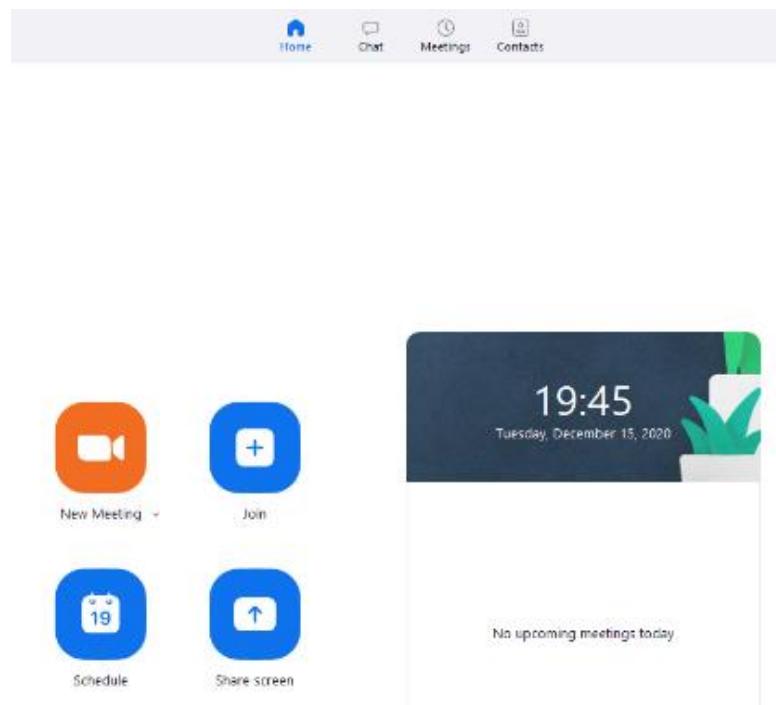


Figure 12: Zoom main UI design

This is the main dashboard for zoom. Here, the user can access many of the features, including to join and create meetings. There is also the option to schedule meetings through the integrated calendar. The navigation bar gives users the option to view their chats, view their meeting history and view all the people they can contact.

2.2.3.9 Teams UI Design

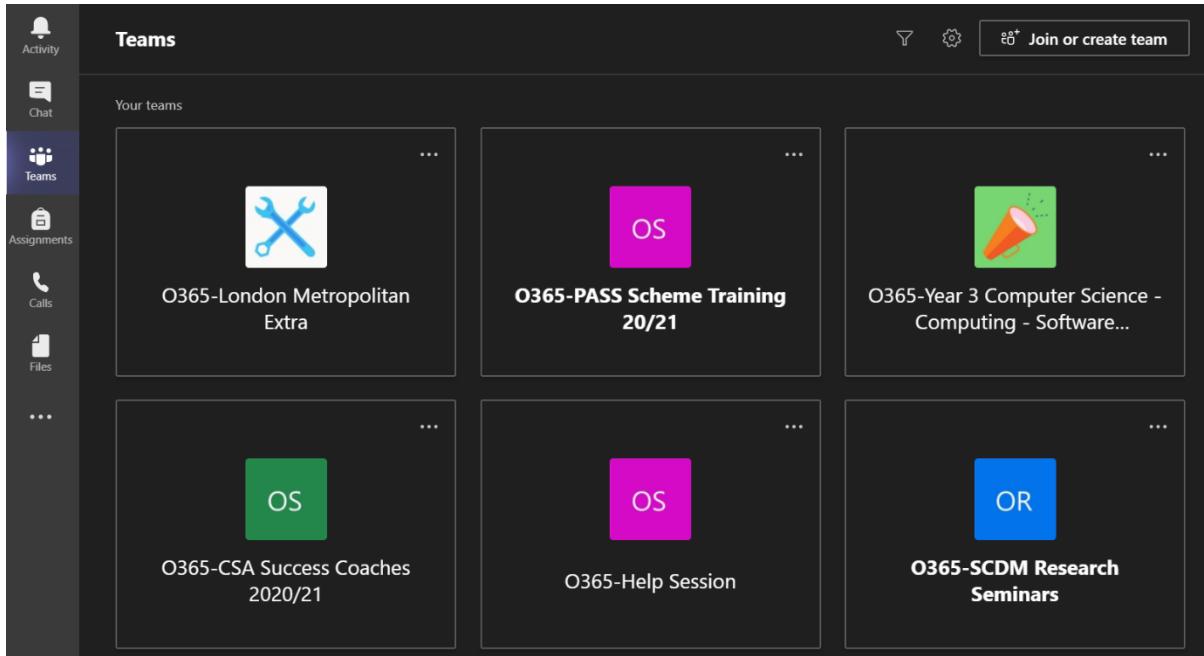


Figure 13: Microsoft teams main UI design

This shows the main dashboard for teams, focusing on the teams. Here the user can view all the teams they have joined, the navigation bar is positioned on the left which includes all the features, like chats, calls and activity. The user can also create or join pre-existing teams.

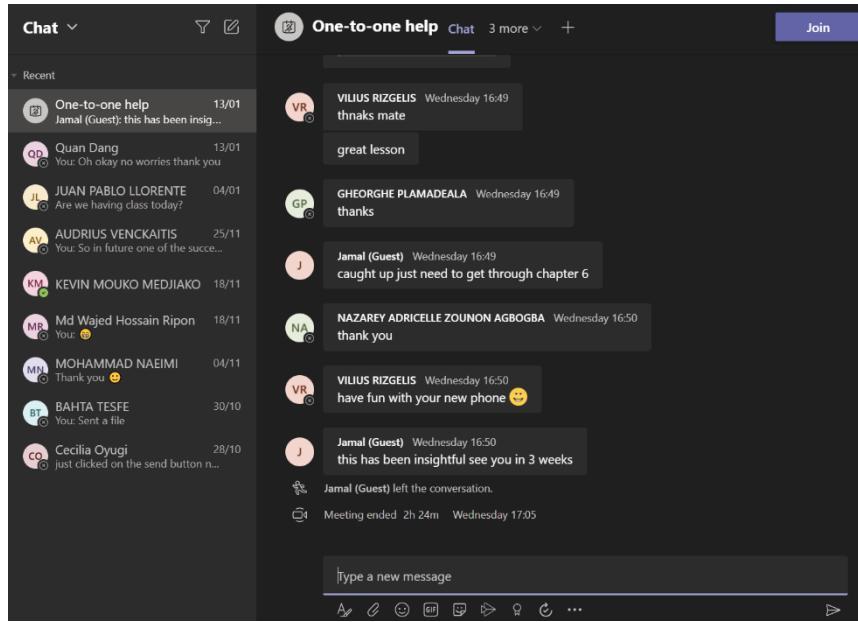


Figure 14: Microsoft teams: Chat UI

This is the main chat UI. Here the user can view all their active chats on the side in the form of a list, the messages will be displayed in the centre of the application, showing all the message history. Users can type a new message, add emojis, share files and attachments and are able to create a new meeting.

2.2.10 State of the Art Technologies Used

To continue the implementation and planning of my project, I will first need to compile the current state of the art software technologies which are currently being used with mainstream software. The software's I have researched include the following technologies:

- Server Technology
- Web Framework
- Database Management System
- Web Design
- Programming Language

I have completed a comparison table to show what technologies have been used within the web communication apps, while researching I could not obtain any information on how the software management tools were developed. This is something I plan to research in more depth for the final project report.

This comparison table shows the differences between all the features I am looking to implement. Many are ones which I do not have much experience in, so therefore within the scope of my project I will be trying to find the balance within these technologies to finalise my choices.

Table 5: Comparison of communication software's state of the art technologies

Software	Server Technology	Web Framework	Programming Language	Database Management System	Web Design
Trello	AWS	Redis	JS and CoffeeScript	MySQL	HTML5 CSS3
Slack	Ubuntu	NodeJS	PHP, Java	MySQL	HTML5 CSS3
Zoom	AWS, Azure	Electron	PHP, JS, Python, C#	Oracle Cloud Infrastructure	HTML5 CSS3
Teams	Microsoft Web Servers	Electron	Angular and Type Script	Microsoft Dataflex	HTML5 CSS3

2.3 The Scope of the Project

The overall objective of my project is to deliver a web application which implements several of the state-of-the-art technologies and features explored previously mentioned within current software which is available. These technologies will be Java, which is currently used in Slack. This is also the programming language I am most comfortable with as I have plenty of experience from past assignments. I have no experience with PHP and C# so it would be wiser to continue using Java. I will also be using HTML and CSS which are used for all Trello, Slack Teams and Zoom. This is the industry standard, and I have a lot of experience using. I will not be using AWS, Ubuntu, Microsoft web servers or Azure, which is implemented in Trello, Zoom, Teams and Slack, as I am not familiar with this technology, I will be using Java EE, as this is what I am most comfortable using and have most experience in.

I will not be using Electron, Redis or NodeJS what all my current market examples use as I have no experience using these web frameworks, it will slow down my project's workflow if I were to learn a new technology. Therefore, I will be using the IDE NetBeans/Eclipse, to ensure my project is completed as efficiently as possible.

I will not be using MySQL for my database, I will opt for the alternative JDBC as it is already integrated into my chosen IDE's, this will make the application more compact and more efficient.

However, not all current software's include all the necessary features needed by a scrum team, for example, Trello includes all the features which include setting tasks and messaging, however it does not include the option to call users or have video meetings. therefore, I will try to implement all which are highly requested and needed for.

As for Zoom, it implements the feature to video and audio call, however it does not give the user the option to privately message users outside of the current 'meeting.' It also does not allow you to set tasks for individual users.

Slack has a smart and simple UI design, it implements features like creating different topics and messaging users privately and topics, however, the feature to video and audio call is hidden behind a pay wall. For my software, I will make sure everything is free for the user to use with all features implemented.

Finally, Microsoft teams is the closest to implementing all desired features. It includes private and group messaging, the ability to create teams, different channels for group messaging, video and audio calling with the ability to share content and to even link your calendar with any events happening. However, it does not include tasks which can be set to user's like on Trello. Therefore, I would like to develop a software which will consider all flaws and benefits of every current market application researched, creating a simplified free to use alternative.

My project's main goal is to merge all features and technologies I have learnt whilst studying at the University, while incorporating new technologies featured in the software's above. I will be implementing a user-friendly UI, with effortless navigation, ability for private messaging, ability to join a general discussion and create and delete them. Role assignment is a vital part, also ability to facilitate meetings.

It is second deliverable is to review and evaluate scrum, which I will achieve by continuing gathering references and researching appropriate sources, presenting these in the final report.

Finally, once my software is complete, I will be asking numerous people to act as my testers, this will be to test the artefact. I will do this to uncover any bugs or errors, this will be done in my testing section of my Gantt chart, to gain my first user feedback and amend any errors or criticism they may have. This will allow my software to reach its full potential and eliminate any risks for software failure.

2.4 Review of development platforms and tools

2.4.1 Technology Selection Criteria

I will now outline what technologies that were originally going to be implemented within my project, along with the final technologies implemented. I considered three main factors when choosing and implementing different technologies within my project. The most important factor considered was the usability of each technology. Meaning, every technology I select will be something I already am experienced and skilled within. Another factor considered is the cost of the project. As this is a university project, I will not be using any technologies other than open source. I will also be looking into the scalability of the project, if my project is successful and I put it onto the current market, would it handle the growth and scale of a public deployment? My final point I will be covering is the timescale of using such technology, this applies to all different factors, as being a project, I must work to the appropriate timescale I outlined in my Gantt chart.

Programming language: Java

"Java was developed by James Gosling, who is known as the father of Java, in 1995." (Javatpoint, 2018) Java started out as a project to be used for television, however, it was too advanced for its time, it was originally being developed by the green team, who *"initiated this project to develop a language for digital devices such as set-top boxes and televisions."* (Javatpoint, 2018) Eventually another company took over the Java technology known as Netscape. This is when Java was shaped into the technology it is today, used for programming and web services.

I will be using Java throughout this project, as it is what I am most comfortable with, as while at university this is what I have mostly coded in. This is because Java is an all-round universal programming language, *"millions of developers running more than 51 billion Java Virtual Machines worldwide."* (Oracle, 2021) it is an industry standard for a programming language and is supported on many different machines and programs, many IDE's such as NetBeans and Eclipse support the use of Java on their platform.

There are many open-source libraries free to use. Java can be compiled to run on a singular PC or throughout a whole server within a network. For scalability, Java can be run on different operating systems like windows and Linux. Java also allows you to connect databases to the applications, using JDBC, which is how I am going to implement my database within the program. Timescale wise, should not affect me within this technology as I am already comfortable using Java for website implementation and database implementation. One of the current market programs, Slack, also uses Java, showing it is suitable to use for my implementations.

Java was the programming language I had completed my project using. This is due to having already a lot of experience coding with it. It aided in the efficiency and completion of the project as many functionalities had already been practised consistently throughout previous University modules and coursework.

Original Server framework: Java EE (Oracle, 2021)

Changed to Java Spring (Spring, 2021)

“Java Platform, Enterprise Edition is the industry standard for developing portable, robust, scalable and secure server-side Java applications.” (Oracle, 2021) I will be using Java EE as it is already built into NetBeans, it provides plenty of simple yet powerful APIs, allowing for easy implementation of my program. Java EE allows for multiple purposes, it can provide APIs for web development and services and communication, which will provide my website with the best industry standards.

Scalability wise, Java EE is perfect as it is popular for creating software, it improves the application performance whilst reducing the complexity of a program. It is completely open source, ideal for my university project. Provides *“developers with a powerful set of APIs while shortening development time”* (Oracle, 2017) This is perfect for my program as time is an important factor, shortening development time will prove beneficial.

During development time, I switched from Java EE to Java Spring. This was due to the project demanding a higher need for fast responsiveness and reactive dialog messages which would ensure the web application looked professional. This change was also made due to the integrated development environment changing to *“IntelliJ.”* (JetBrains, 2021) Spring also allows for *“fast, secure, and responsive web applications connected to any data store.”* (Spring, 2021) This shows that it was a more convenient choice compared to Java EE, due to the easy implementation of any database, so therefore there was more freedom picking a more suitable one for the project.

Backend database: JDBC (Oracle, 2021)

Changed to MySQL (MySQL, 2021)

JDBC was first developed and *“released, as part of JDK 1.1 in 1997,”* (Tyson, 2019) by Sun Microsystems. JDBC was originally released for client-side purposes only, that could interact with a data source provided by Sun Microsystems. Server-side support was later released as a package in JDBC 2.0. *“The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases.”* (Oracle, 2021) Examples of databases supported are SQL, Excel, and other spreadsheets.

JDBC is a Java API which allows for the connection of databases. It is widely used with developers who use Java as their primary language, *“Write Once, Run Anywhere”* (Oracle, 2021) as it provides quick and easy implementation and connectivity to their program. This therefore simplifies time spent on using external software, as it is already integrated within NetBeans IDE. Also, it can read any database, allowing for even better scalability if the program were ever to reach public market. More tables could be created with ease, allowing it to be easily upgradable.

During development, JDBC was replaced for MySQL, due to it being very efficient and easier to integrate into the application. The positives of MySQL outweigh JDBC, evaluating the positives and negatives of JDBC compared to MySQL. JDBC does not allow for updating tables in a sequence, this is always random. Inserting data is a lot more complicated, however, within MySQL, the UI makes it easier to update tables while testing and creation, this was useful throughout the testing lifecycle of the project. Also, within JDBC, the correct drivers always need to be deployed for each type of database.

Frontend Development: HTML and CSS

Changed to Java React (React, 2021)

The origins of HTML go all the way back to the 1930s, the first documentation of HTML was by an engineer named Vannevar Bush, where he wrote an article, this raised attention as “*the system he discussed in this article is very similar to HTML.*” (Bellis, 2019) Because of this, the framework and theory of HTML helped others invent basic web development, including URLs, HTTP and HTML. This was done by Tim Berners-Lee. “*Berners-Lee's browser editor was developed in 1991-92. This was a true browser editor for the first version of HTML and ran on a NeXt workstation.*” (Bellis, 2019)

This was the first ever developed version HTML, the official version was released in the following year, “*June 1993.*” (Bellis, 2019) Since then, it has grown to become the industry standard.

After HTML was created by Berners-Lee, CSS had not been developed yet, CSS was “*first proposed by Håkon Wium Lie in October of 1994. It was called Cascading Style Sheets, or just CSS.*” (Hoffmann, 2017) CSS stands for ‘Cascading Style Sheets,’ when using HTML, CSS instructs the scripting language how to format and stylise the page. It is known as a “*declarative programming language,*” (Hoffmann, 2017) this means that we are instructing HTML how to display the page, so then the chosen browser can render and format the page correctly, corresponding to the given CSS instructions.

HTML and CSS are the true customary languages used widely within web designing and development. I have previously implemented both within my coursework last year for a games review website, plenty of resources are made available to help style and layout my website. It is free, easy to implement and integrate within programs and easily able to be expanded on for future upgrades.

During the implementation period of the project, Java React was used to fully experiment with all possible frontend development. This was due to already being confident within CSS/HTML, the project needed more goal and ambition, this also allowed for more in-depth research of another JavaScript library.

Integrated Development Environment (IDE): NetBeans/Eclipse

Changed to IntelliJ IDEA (JetBrains, 2021)

I will be using one of the 2 stated IDE’s as I am comfortable with both considering I have used them throughout my degree. NetBeans started off like my final project, it was a student project developed in “*Czechoslovakia, now the Czech Republic, in 1996.*” (NetBeans, 2021) NetBeans was developed by a group of students, originally, its name was Xelfi, this changed when an entrepreneur, Roman Stanek, had interest in the project as the students released a website and built a company around the early release. “*The original business plan was to develop network-enabled JavaBeans components.*” (NetBeans, 2021) Finally, in 1999, they released “*NetBeans DeveloperX2.*” (NetBeans, 2021)

As for Eclipse, it was developed on a business scale and need by IBM, this was developed “*late 1990s in response to the need for a common development platform for all IBM products.*” (Botello, 2021) Eclipse was developed with the purpose of giving companies and businesses to use an open-source IDE. This was successful as many companies started to incorporate Eclipse within their own products which were available on the market, growing its success. It began to claim even more fame when the Eclipse Foundation was created, companies that were involved with the development were “*Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP, and Serena Software.*” (Botello, 2021)

They are both open source, support Java and JDBC. With this dynamic language support, I can easily implement HTML and CSS whilst using Java for backend implementation. They also support the scalability as they provide support for Java standards and platforms, like Java Mobility Support, allowing me to move the application over to mobile for future expansions.

Due to consistent issues, and a lack of motivation using NetBeans, it was decided to use another Integrated Development Environment, IntelliJ. IntelliJ was used, as was designed for developers to build full scale web applications within. *“In every design and implementation decision that we make, we keep in mind the risk of interrupting the developer’s flow and do our best to eliminate or minimize it.”* (JetBrains, 2021) This is backed up by IntelliJ automatically adding tools which relate to your specialisation of work and offer *“Smart code completion.”* (JetBrains, 2021) This greatly increased the amount of workflow produced, as it allowed for quicker completion of certain amounts of code auto filling in long statements.

2.5 Summary of changes implemented during development period

Here are all the changes within the technology, that had occurred during development:

- **Framework:** Java Spring
- **Frontend:** Java react
- **Database:** MySQL
- **Integrated Development Environment:** IntelliJ

Due to experiencing many issues regarding setting up NetBeans, I decided to change the IDE to IntelliJ, as I encountered many bugs regarding setting up the server, glassfish, and the database using MySQL. I opted for IntelliJ as I enjoy working on it while practicing coding and experimenting, I feel it performs much more efficiently compared to NetBeans. IntelliJ is now my main IDE due to these reasons and I feel for further implementations within my artefact it will give me more stability to work with.

I also changed the database over to MySQL, due to it being very efficient and easier to integrate into my artefact, this was due to extensive research, evaluating the positives and negatives of JDBC compared to MySQL.

I also used java spring for the backend framework of the development and java react for the frontend. I felt like I needed to experiment with my project to the best of my abilities, so I spent time researching how to use Spring and React, I feel like the UI looks much cleaner and more presentable using this over standard html and CSS. I feel like this went very successfully and I am happy with the results as to date, I would like to experiment more towards the final submission. For example, dialogue boxes have better animation and look more professional compared to standard html and CSS.

Chapter 3: Requirement Analysis

"Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified." (VisualParadigm, 2021) This is a very important process of software engineering, as it provides evidence and a blueprint which is defined, it shows the tasks and features which the software requires for it to be successful. Many developers will refer to their diagrams and models throughout the entire development process. All conditions and features are argued and displayed, these must meet the stakeholder's requirements and the team must be able to balance this out along with their own experience and resources.

A general definition of Software requirement is as follows, "*Software requirement is a capability needed by the user to solve a problem or to achieve an objective.*" (VisualParadigm, 2021) This also emphasises the focus and attention needed to satisfy a stakeholder's requirements. Within this, we are aiming to achieve ultimate quality and standard for the final product, taking the budget into consideration.

The main activities associated with Requirement Analysis are specific and follows it in this format:

- Understanding the stakeholders needs and requirements, this is done by the initial communication before anymore planning is carried out.
- Then, the development team must come together and use the stakeholder's requirements and compare to their own resources to see if the project can be carried out effectively, or perhaps certain aspects they may not have the expertise or experience to complete.
- Then, the team must apply technical analysis to this, they will need use previous models of existing software to make sure they are completing and implementing functions effectively and correctly.
- Then, the development team must come up with a plan to ensure that all activities and processes are carried out, this is done by a schedule and meetings that must be attended.
- Finally, the development team must finalise the software and create "*system definitions.*" (VisualParadigm, 2021) This just involves finalising the overall systems functions and properties.

I will be applying requirement analysis into my project by using a use case diagram. UML diagrams are the most effective way of showing all the requirements of a software from an early stage within the planning. This will further show how the software will function to an end user or tester using it.

Use case diagrams show the basic functions of the software, I have not gone into complete detail of all features that will be implemented within the software. Also, my diagram does not show all the relationships between the actors and systems, only a few are established within it. My use case diagram also shows the steps performed, so if the user would want to register it shows the unregistered user interacting with the system to produce the end goal, becoming a registered user with all access to the application. Therefore, within the chapter 3, these are the main diagrams I will be discussing.

- **UML Case Diagram**
- **High-Level Use Cases**
- **Use Case Descriptions**

3.1 Use Case Diagram

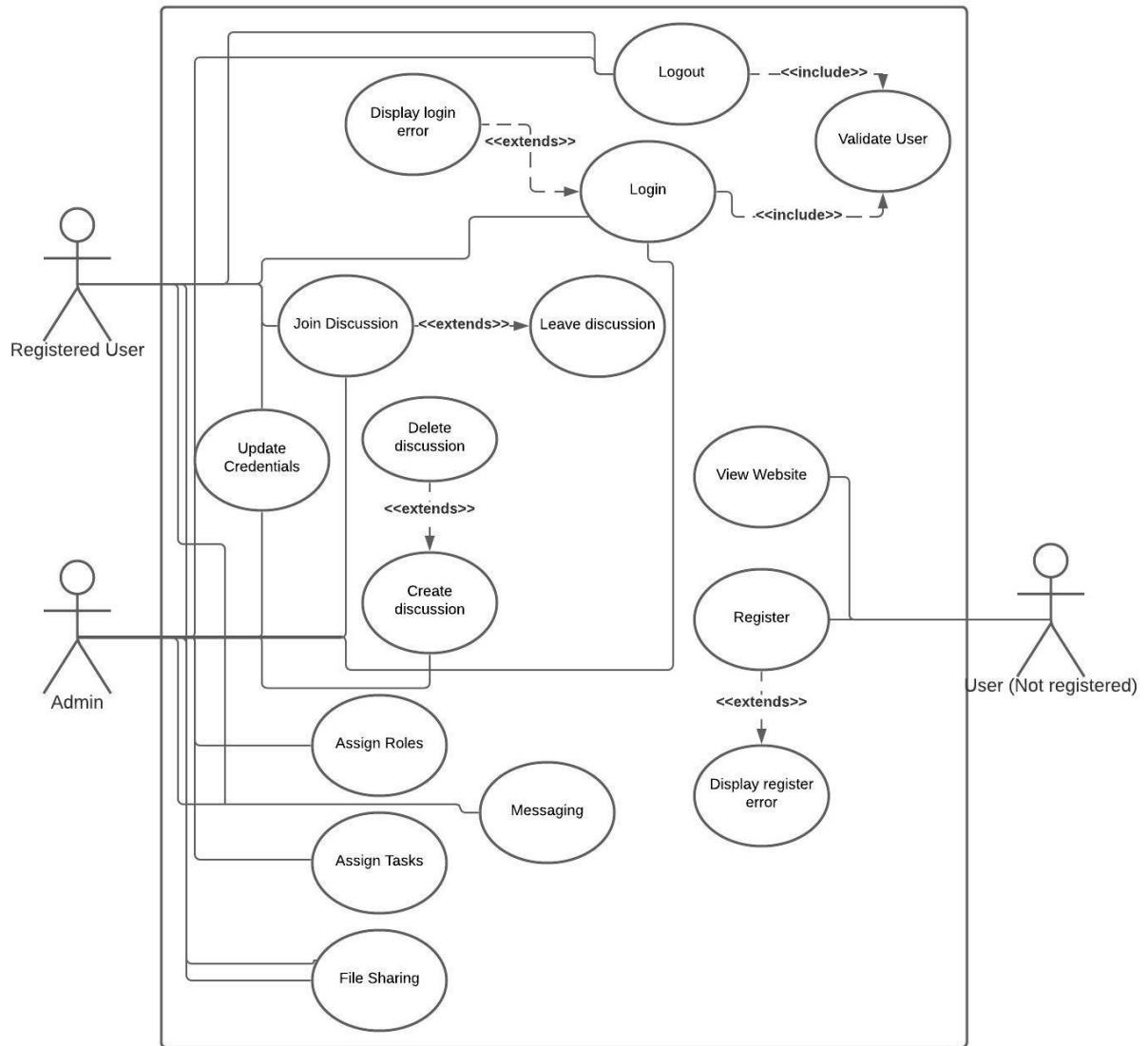


Figure 15: Use Case Model for my software

3.2 High-Level Use Cases

1. Use Case: View Website

Actor: User (Non-registered)

Description: Any user can visit the website to view all the functions and features it offers, the website will list reasons why scrum teams should migrate their work onto this site.

Figure 16: Use Case: View Website

2. Use Case: Register

Actor: User (Non-registered)

Description: User can view website and decide if they would like to sign up. They can access the navigation bar which will lead them to a register page. They can fill out the form and successfully register into the website.

Figure 17: Use Case: Register

3. Use Case: Login

Actors: User, Admin

Description: A user or admin can use the navigation bar to use the log in link, they will then be prompted to enter their credentials, this being their chosen username and password. If it is successful, they will be redirected to the main page showing their current discussions and tasks.

Figure 18: Use Case: Login

4. Use Case: Logout

Actors: User, Admin

Description: The user can click the log out link on the navigation bar to close their session.

Figure 19: Use Case: Logout

5. Use Case: Update Credentials

Actors: User, Admin

Description: Both user and admin can click onto the navigation bar which will display an option to change credentials. When the user has changed their desired credential, they can then click on save changes, this will then display a success message to confirm that the changes have been confirmed.

Figure 20: Use Case: Update Credentials

6. Use Case: Join Discussion

Actors: User, Admin

Description: Both a user and admin will be allowed to join a discussion if they are a part of that certain scrum team. This is done by clicking on the join discussion tab.

Figure 21: Use Case: Join Discussion

7. Use Case: Assign Roles

Actor: Admin

Description: Admin is the only user allowed to assign roles to a user. Once a user has joined the discussion, the admin can click on their name and add or remove any role given to the user.

Figure 22: Use Case: Assign Roles

8. Use Case: Assign Tasks

Actor: Admin

Description: Admin is the only user allowed to assign tasks to a user. Once a user has joined the discussion, the admin can click on their name and add or remove any task given to the user.

Figure 23: Use Case: Assign Tasks

9. Use Case: File Sharing

Actors: User, Admin

Description: Any admin or user can share files by simply selecting the add file button in a discussion or private message and share anything of relevance to other team members.

Figure 24: Use Case: File Sharing

10. Use Case: Messaging

Actors: User, Admin

Description: Users can send a message by selecting an individual or discussion they want to message on, type the message within the text field and send it to the appropriate users.

Figure 25: Use Case: Messaging

11. Use Case: Create Discussion

Actors: Admin

Description: Admins can create discussions as they are the ‘scrum master.’ They will choose what discussions should be named by selecting the create new discussion button, they will then allow team members to join.

Figure 26: Use Case: Create Discussion

3.3 Use Case Descriptions

1. Use Case: Register

Actor: User (Non-registered)

Description: User can view website and decide if they would like to sign up. They can access the navigation bar which will lead them to a register page. They can fill out the form and successfully register into the website

Event Table:

Table 6: Use Case: Register

Actor	System Responses
1. The user navigates to the application form and fills out their details.	2. System will check if the username and password are approved.
	3. Checks the user's email, name and DOB are all correct and suitable.
	4. Informs the user know all their details are correct and proceeds to finalise the registration.
5. User will be then redirected to the login page where they will re-enter their details to login.	

Other Actions:

Action 1.1: When filling out the form, the user leaves fields blank. System cues user to fill out all the fields.

Action 1.2: User has inputted an incorrect format of one of the personal details. System will cue user to re-enter.

Action 2.1: Username is already in use; system will cue the user to provide another username. This will also be checked if suitable.

Action 2.2: Password is not suitable as it does not include the standard format of capitals and numbers. System will cue user to re-enter a stronger password.

Action 4.1: The registration might not be confirmed on the webpage due to a possible error, therefore the user will have to try again, ending the use case.

Action 4.2: There could be an error with communication between the server and the website. In this case, the user will be told to register again, or wait some time before trying again. This will cause the use case to end.

Action 4.3: External factors, including a laptop battery dying or loss of power could cause the webpage to lose the user's information before submitting, therefore the user will have to re-enter all their details. The use case will then end.

Pre-conditions:

The user must have a working email address and internet connection to connect to the website.

Post conditions:

The new registered user can now successfully login and view the main dashboard of the website.

2. Use Case: View Website

Actor: User (Non-registered)

Description: Any user can visit the website to view all the functions and features it offers, the website will list reasons why scrum teams should migrate their work onto this site.

Event Table:

Table 7: Use Case: View Website

Actor	System Responses
1. The non-registered user will search up the name of the website or either copy or paste the URL into the search bar.	2. The website will display the main welcome page with the navigation bar will all the webpages listed.
3. The non-registered user will click on a page they would like to look at.	4. This would then load for the non-registered user.
5. If the non-registered user has not registered or is done looking at the website, they will then close the webpage via the browser.	

Other Actions:

Action 1: The webpage is currently down, as the server hosting it is having technical difficulties. The browser the user is using will display an error message, this will cue the user to reload the page or try again later, ending the use case.

Action 2: The links on the website may not all be working as supposed to. The website will show an error message, ending the use case.

Action 3: The website is not loading all the components; therefore, the user is prompted to reload or use the website again later. This will end the use case.

Pre-condition:

The user must have a stable internet connection and a functioning desktop/laptop.

Post condition:

Not Applicable

3. Use Case: Login

Actors: User, Admin

Description: A user or admin can use the navigation bar to use the log in link, they will then be prompted to enter their credentials, this being their chosen username and password. If it's successful they will be redirected to the main page showing their current discussions and tasks.

Event Table:

Table 8: Use Case: Login

Actor	System Responses
1. The registered user will load up the website and navigate to the log in form page where they will fill in their credentials, username, and password.	2. The system will check that both fields have been filled with appropriate data types.
	3. The system will then authenticate if the username and password match.
	4. Displays a success message to the user.
	5. Redirects user to the main dashboard.

Other Actions:

Action 1: The user leaves empty fields for either username and password or both. The system will cue the user to fill out all the fields.

Action 1.2: The user is not registered, the system will display a message informing the user that their username does not exist, prompting the user to register and then try to login in. This will end the use case.

Action 3: The server is down to the website and is taking too long to respond. The system will then display a message to the user to prompt them to try again later or reload the page. This will end the use case.

Action 5: The website's main page is not loading. This will make the system display an error message to the user, prompting them to reload the page or try again. The use case ends.

Pre-condition:

The user will need to be registered successfully as a user, also they will need stable internet connection.

Post condition:

They will be able to access their chats and team chats.

4. Use Case: Update Credentials

Actors: User, Admin

Description: Both user and admin can click onto the navigation bar which will display an option to change credentials. When the user has changed their desired credential, they can then click on save changes, this will then display a success message to confirm that the changes have been confirmed.

Event Table:

Table 9: Use Case: Update Credentials

Actor	System Responses
1. The registered user will select the update account details on the navigation.	2. The system will load the current stored credentials of that user.
3. The user will change different details and then proceed to click the save changes button.	4. The system will check if the updated details are suitable and correct, the system will then ask the user to confirm for a final time.
5. User confirms the changes	6. The system will modify the credentials and then allow the user to view the changes.

Other Actions:

Action 2: The website might not be able to load to the change detail form, displays an error message for the user telling to reload or try again, this will end the use case.

Action 3: The user leaves one or all the detail field empty. This will cue an error message, making the user fill in all the text fields.

Action 3.1: The password might not be suitable; error message tells user to re-enter stronger password.

Action 3.2: The user might try to put an illegal datatype in one of the fields, error message will inform the user to re-enter with correct details.

Action 3.3: User might try to take a username which is already in use, error message will inform user to choose another name.

Action 6: The website could crash, this would mean any unsaved detail changes would be lost, therefore the user would need to fill out the form again, ending the use case.

Pre-condition:

The user will need to be registered on the website already, the user will also need a stable internet connection and a suitable device to access the website from.

Post Condition:

The user will be able to modify their details and they will be successfully changed.

5. Use Case: Logout

Actors: User, Admin

Description: The user can click the log out link on the navigation bar to close their session.

Event Table:

Table 10: Use Case: Logout

Actor	System Responses
1. A logged in user navigates to the log out button, clicks it.	2. System will authenticate the request.
3. User redirected to the main page of the website, logged out.	

Other Actions:

Action 1: The webpage could stop responding, this will show an error message to inform the user to reload the page, this would end the use case.

Action 2: The server might be down so therefore the system cannot authenticate the request, ending the use case. An error message will be displayed.

Action 3: The webpage might not redirect the user to the main page logged out, an error message will be displayed, ending the use case.

Pre-conditions:

The user will need to be registered and already logged in with a stable internet connection.

Post Conditions:

The user will successfully be logged out and can then register another account or log in again.

6. Use Case: Assign Roles

Actor: Admin

Description: Admin is the only user allowed to assign roles to a user. Once a user has joined the discussion, the admin can click on their name and add or remove any role given to the user.

Event Table:

Table 11: Use Case: Assign Roles

Actor	System Responses
1. The Admin will select a discussion.	2. The system will load the discussion.
3. Admin will click on the user they want to modify roles, change their role, and confirm changes.	4. The system will authenticate and update the role change.
	5. The user will be informed of the role change.

Other Events:

Action 2: Server is taking too long to respond, error message is displayed informing the user to reload or try again later, this ends the use case.

Action 4: The server cannot authenticate the change, due to crashing or other error, suitable message displayed to user, ends use case.

Action 5: Message is not displayed for the user, error message will display, telling user to reload to see modified changes.

Pre-conditions:

The user is an admin of that discussion, they are registered on the website and have a stable internet connection.

Post conditions:

The selected user's role is updated, and the user is informed, changes are saved, and user allowed to access role locked permissions.

7. Use Case: Assign Tasks

Actor: Admin

Description: Admin is the only user allowed to assign tasks to a user. Once a user has joined the discussion, the admin can click on their name and add or remove any task given to the user.

Event Table:

Table 12: Use Case: Assign Tasks

Actor	System Responses
1. Admin will navigate to discussions, select a discussion.	2. The system will load the selected discussion.
3. Admin will select the user, assign a task, and confirm changes.	4. System will check changes and authenticate.
	5. Once successful, changes will be confirmed, and user will be informed.

Other Actions:

Action 2: Server is taking too long to respond, error message is displayed informing the user to reload or try again later, this ends the use case.

Action 3: Admin adds a task containing illegal data types, this displays an error message informing the Admin to amend the task.

Action 4: The server cannot authenticate the change, due to crashing or other error, suitable message displayed to user, ends use case.

Action 5: Task is not displayed for the user, error message will display, telling user to reload to see modified changes.

Pre-conditions:

The user is an admin of that discussion, they are registered on the website and have a stable internet connection.

Post conditions:

The selected user's task is updated, and the user is informed, changes are saved, and user allowed to access role locked permissions.

8. Use Case: File Sharing

Actors: User, Admin

Description: Any admin or user can share files by simply selecting the add file button in a discussion or private message and share anything of relevance to other team members.

Event Table:

Table 13: Use Case: File Sharing

Actor	System Responses
1. User will click on the selected private chat or discussion	2. System will load selected chat/discussion.
3. User will click the file share button, choose their file, and click confirm.	4. Server will authenticate the request and send file to discussion/user.
	5. The designated participants will receive the file.

Other Actions:

Action 1: Webpage does not respond, error message to inform user to reload or try later, ends use case.

Action 2: Website crashes, displays an error message for the user, ends use case.

Action 3: User selects an incompatible file type, error message is displayed, telling user to select an appropriate type.

Action 4: Server takes too long to respond, error message displayed, ending use case.

Action 5: File is not sent to other users; error message is displayed. Ends use case, user informed to try again.

Pre-Conditions:

The user will need to be registered, have a stable internet connection and suitable device to access website from.

Post Conditions:

The user will successfully share a file between other users, in a private message or in a discussion.

9. Use Case: Messaging

Actors: User, Admin

Description: Users can send a message by selecting an individual or discussion they want to message on, type the message within the text field and send it to the appropriate users.

Event Table:

Table 14: Use Case: Messaging

Actor	System Responses
1. User selects discussion/user they would like to message.	2. System loads the discussion/chat page.
3. User types out message and clicks send.	4. Server authenticates message and sends the message to the other client or discussion.
5. Message visible in chat to all users.	

Other Actions:

Action 2: Webpage does not respond, error message to inform user to reload or try later, ends use case.

Action 3: User leaves message field blank, this will trigger an error message informing the user to fill out the field.

Action 4: Server takes too long to respond, message fails to send, error message displayed, ending use case.

Action 5: Message not visible to other users, error message displayed to tell user to reload the discussion/private message page.

Pre-Conditions:

The user will need to be registered, have a stable internet connection and suitable device to access website from. User also needs to be part of the discussion or able to private message other users.

Post Conditions:

The user successfully sends a message to the discussion page or the private message and it is displayed for all users.

10. Use Case: Create Discussion

Actors: Admin

Description: Admins can create discussions as they are the ‘scrum master.’ They will choose what discussions should be named by selecting the create new discussion button, they will then allow team members to join.

Event Table:

Table 15: Use Case: Create Discussion

Actor	System Responses
1. User will click create discussion button.	2. System will load create discussion form.
3. User will enter discussion details, then proceed to confirm.	4. System will authenticate the creation, successfully create discussion.
	5. Redirects user to newly created discussion page.

Other Actions:

Action 1: Webpage does not respond, error message to inform user to reload or try later, ends use case.

Action 2: Website crashes, displays an error message for the user, ends use case.

Action 3: User leaves fields blank, error message display to inform user to fill out all fields.

Action 4: User enters incorrect data types, error message displayed, informing the user to re-enter suitable details.

Action 4.1: Discussion page name is already taken, error message displayed, informing user to choose another suitable name.

Action 5: Website crashes, takes too long to respond, error message displayed, ends use case.

Pre-Conditions:

The user will need to be registered, have a stable internet connection and suitable device to access website from.

Post Conditions:

The user successfully creates the new discussion, allows other users to join.

3.4 Functional Requirements

Function requirements detail the systems behaviour, how it will process different actions under different conditions.

These are the requirements the program must perform:

Table 16: Functional Requirements

Requirement	Status
✓ The website will allow any guests viewing the website access to login or register using the forms.	Completed
✓ The website will allow a user to update their details once they are logged into the site by filling out the change details form.	Completed
✓ The website will allow users to join discussions, send files, and messages within them.	Completed
✓ The website will allow for roles to be assigned to users so they can keep track of their assignments.	Completed
✓ The website will allow for users to private message one another, also send each other files.	Completed
✓ The website will also allow Admins to assign roles to users changing their privileges, through MySQL	Completed
✓ The website will store all the user's messages and files and discussions in a database, so everything is saved.	Completed

3.5 Non-Functional Requirements

Non-functional requirements list all enhancements to the overall software's security and functionality.

Here are the criteria which the program will follow:

Table 17: Non-Functional Requirements

Requirement	Status
✓ The website will have similar theme and feel throughout using a smart and simple GUI which will be easy navigable.	Completed
✓ The website will have a pleasant and modern design and style.	Completed
✓ The website's UI will be able to transferrable to a mobile version, to allow for future expansions and growth of the website.	Completed
✓ User passwords are protected by hashing them, within the database	Completed

3.6 Extra Functionality

Here is the extra functionality added into the program throughout the time of the software's development. These functionalities were not discussed within the interim report, due to completing tasks relatively, ultimately deciding to implement other functionalities to give the application more features.

Table 18: Extra Functionality

Requirement	Status
✓ Users can privately video chat with another user.	Completed
✓ Users can log out of the website.	Completed
✓ Task creation and assignability, also the ability to complete a task.	Completed
✓ Admins can delete discussions.	Completed
✓ Ability to delete account.	Completed

Chapter 4: Software Design

For my software design, I used Java React to design all the frontend development. At the beginning of my development, I used NetBeans Java Frames as a template to design the initial design, this was used during the interim report to show my ideas and planning for frontend development. This was then developed within IntelliJ using Java React, JavaScript to create a more professional feel compared to standard HTML and CSS, that I had originally planned.

To show off my application, I will now show all the webpages featured within my application, this will include a screenshot of the overall design and an explanation of the purpose of the webpage. I also have combined the initial frontend design of the web page and combined it with the finished application to compare and discuss any significant changes made.

4.1.1 Navigation Bar

This is the navigation bar I decided to implement after completely re-designing my website within the final version. The user can click 'Scrum Mgt App,' this will take them to the homepage of the application. There are tabs for all the main pages of the website, including extra tabs for 'Contact Us,' also 'About us,' users can find helpful information about the applications purpose. The navigation site is the main tool the user will use to navigate around the site, as all pages are listed here.

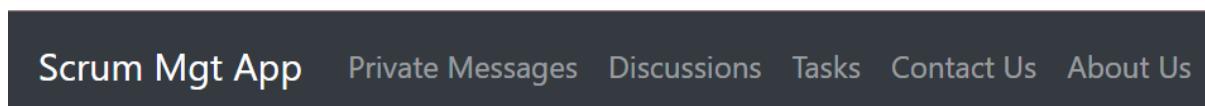


Figure 27: Navigation Bar design

On the right-hand side of the navigation bar, the user will be able to change account details or delete an account by navigating to the 'Account Details' tab. Also, while logged in, the user will be able to view which user is logged into the site, this was crucial to include while testing, as it allowed me to easily view the current user that was logged in, including the role status of current user.

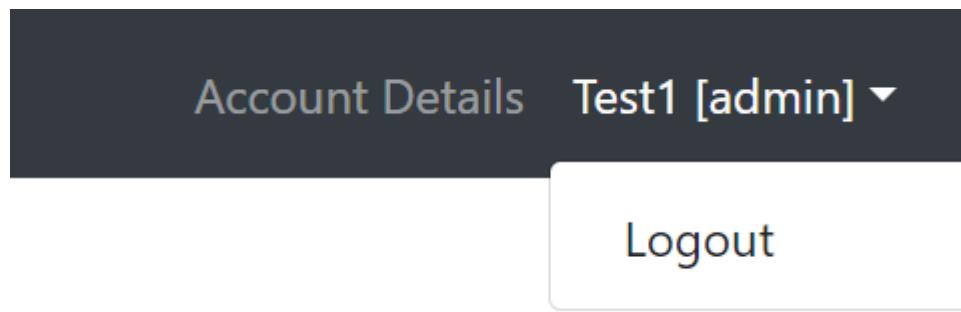


Figure 28: Right hand side of Navigation Bar design

4.1.2 Homepage

This is the first page that will be displayed is what will appear when user searches for the website. This page will give basic details of the site and its functions and entice scrum teams to move their work to this site.

Within the initial design, it shows the original design idea I had created, compared to the final product it captures the same vision I had originally had, only difference is the ‘Login’ and ‘Register’ within the navigation bar. These tabs were not included in the final application, as I felt that within a work environment the user would stay logged in within their own account, therefore they were not used within the final design, simplicity allows for easier navigation around the web application.

Within this screenshot, the left-hand side of the navigation bar is shown, as well as some of the information given to the user to show the user what page they are currently viewing.

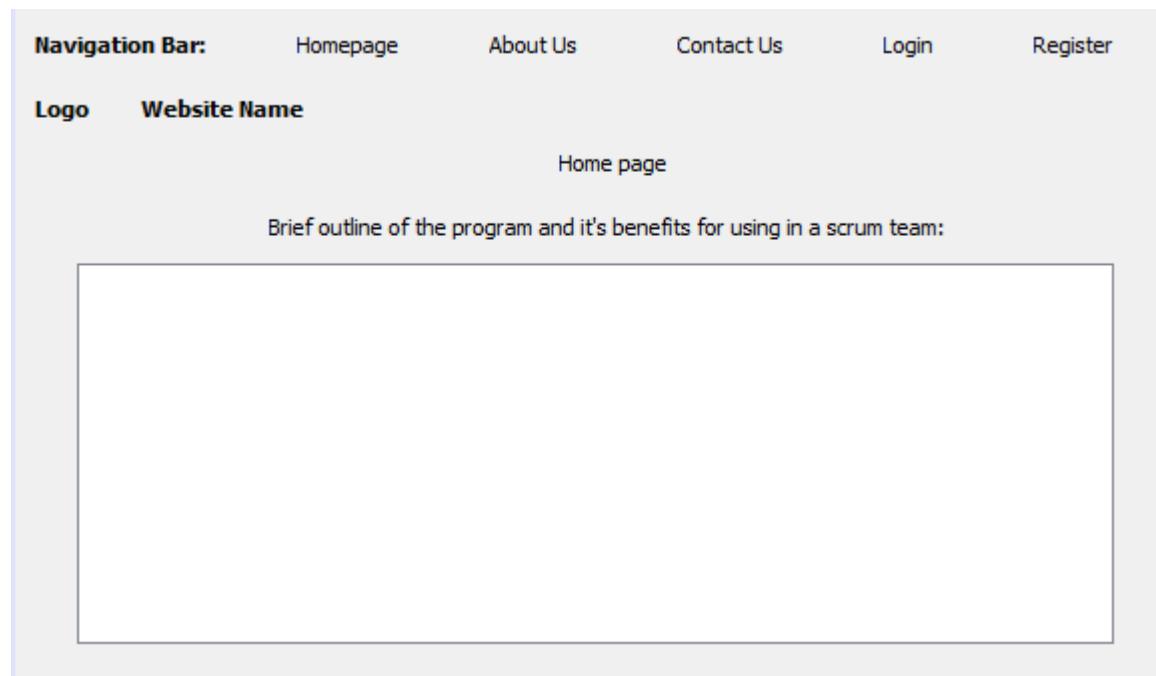


Figure 30: Initial Homepage design

Scrum Mgt App Private Messages Discussions Tasks Contact Us About Us

Scrum Management Application Homepage

Purpose of application?

This application is designed specifically for scrum teams.

To ensure maximum work efficiency, this application will allow scrum teams to effectively communicate through discussions, where files and attachments can be sent, as well as through private messaging.

Figure 29: Homepage design

4.1.3 Login Page Design

When a user opens the web application, they will be greeted with the login page. I decided to combine both the login and register page together, for easier access. The login page is simple, user's will only need to enter their username and password they used to register to the website. Once the user successfully logs in, it will redirect to the main homepage. The final design stays true to the initial design, using all aspects of the design, slightly improving on the layout and efficient login/register design.

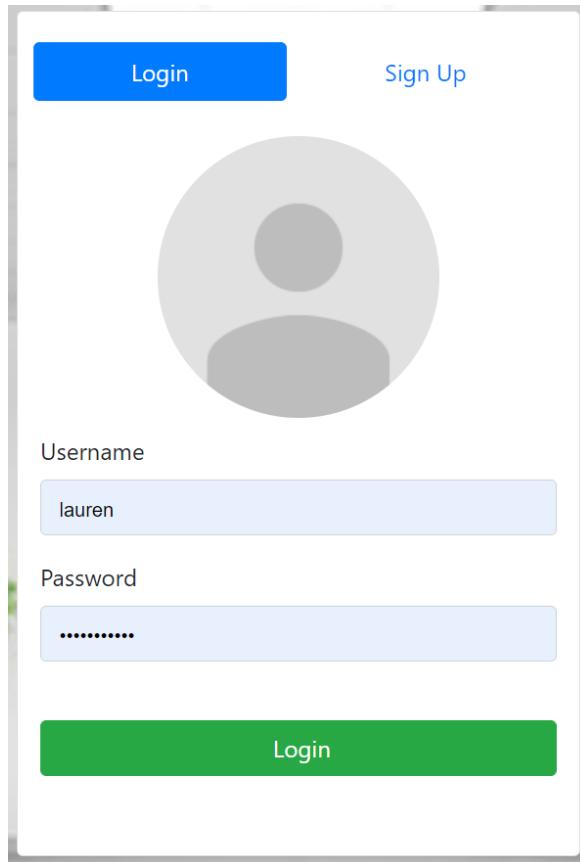


Figure 31: Final Login page design

A screenshot of an older, more basic login interface. It features a title 'Please Login' at the top. Below it are two input fields: one for 'Username' containing 'Bobby' and another for 'Password' containing four asterisks ('****'). There is a small checkbox labeled 'Remember me' followed by a blue 'Login' button.

Figure 32: Login Page Initial Design

4.1.4 Register Page Design

For my register page, the user can enter their details, once entered, the user will need to click the ‘sign up’ button. If the user has already registered on the web application, they can click the ‘login’ button, this will re-direct them to the login page. All fields will be required, the user has the option to change them later in the ‘account details’ page. The final design stays true to the initial design.

The final register page design features a clean, modern look. At the top right are two buttons: 'Login' in blue and 'Sign Up' in white with a blue border. Below these are six input fields: 'Username' (empty), 'Email' (containing 'lauren'), 'Password' (containing '*****'), 'Confirm Password' (empty), 'First Name' (empty), and 'Last Name' (empty). A large green 'Sign Up' button is positioned at the bottom right.

Figure 33: Final Register page design

The initial register page design is a wireframe version. It includes a navigation bar with 'Homepage', 'About Us', 'Contact Us', and 'Login' links. Below the navigation is a section for 'Logo' and 'Website Name'. The main form area is titled 'Register' and contains six input fields: 'Username' (placeholder '<User Code>'), 'Password' (placeholder '<User Code>'), 'First Name' (placeholder '<User Code>'), 'Second Name' (placeholder '<User Code>'), 'Email Address' (placeholder '<User Code>'), and 'Phone Number' (placeholder '<User Code>'). A 'Register' button is located at the bottom right. A note at the bottom says 'Already a user? [Login](#) here instead!'

Figure 34: Initial Register Page Design

4.1.5 Discussion Page Design

The discussion page is where the current discussion the user wants to view is shown. The messages from all the users and tasks are also be displayed, also the list of team members is also shown. For a regular team member using the discussion it will only list the names roles and tasks assigned to their name. The user will also be able to call the team members which will redirect them to the call page. Within the discussion, a user can create a task and assign it to themselves. All tasks can be viewed on the tasks tab.

The screenshot shows the final design of the discussion page. On the left, there's a sidebar with a 'List of Discussions' section containing 'Discussion 1' and 'Discussion 2'. A blue button 'Add a Discussion' is at the top of this section. Below it is a 'Team Members' table with two rows:

ID	Member	Task
2	Test Two	Select...
1	Test One	Select...

In the center, a large box titled 'Discussion 1' shows two messages from 'Test1': 'Hello' (Mar 31, 2021 02:13:21) and 'Test1' (Apr 9, 2021 02:51:15). At the bottom of this box are input fields for a message ('Write your message...'), file upload ('Choose files'), and a green 'Send' button.

To the right of the discussion box is a 'Tasks' section with a table:

ID	Task Name	Status	Assignee
1	Do paperwork	Completed	test2
2	New Task	Completed	Test1

Figure 35: Final Discussion Page Design

This wireframe illustrates the layout of the discussion page. At the top is a navigation bar with links: Homepage, Discussions, Private Messages, All Tasks, Contact Us, About Us, 'Hi, "Username"', Account Details, and Logout. Below the navigation is a logo and website name.

The main area has three main sections: 'List of Discussions' (button), 'Discussion Name' (text input), and 'Team Members' (button). The 'Discussion Name' section contains a large empty box for the discussion content. The 'Team Members' section contains a table with two rows:

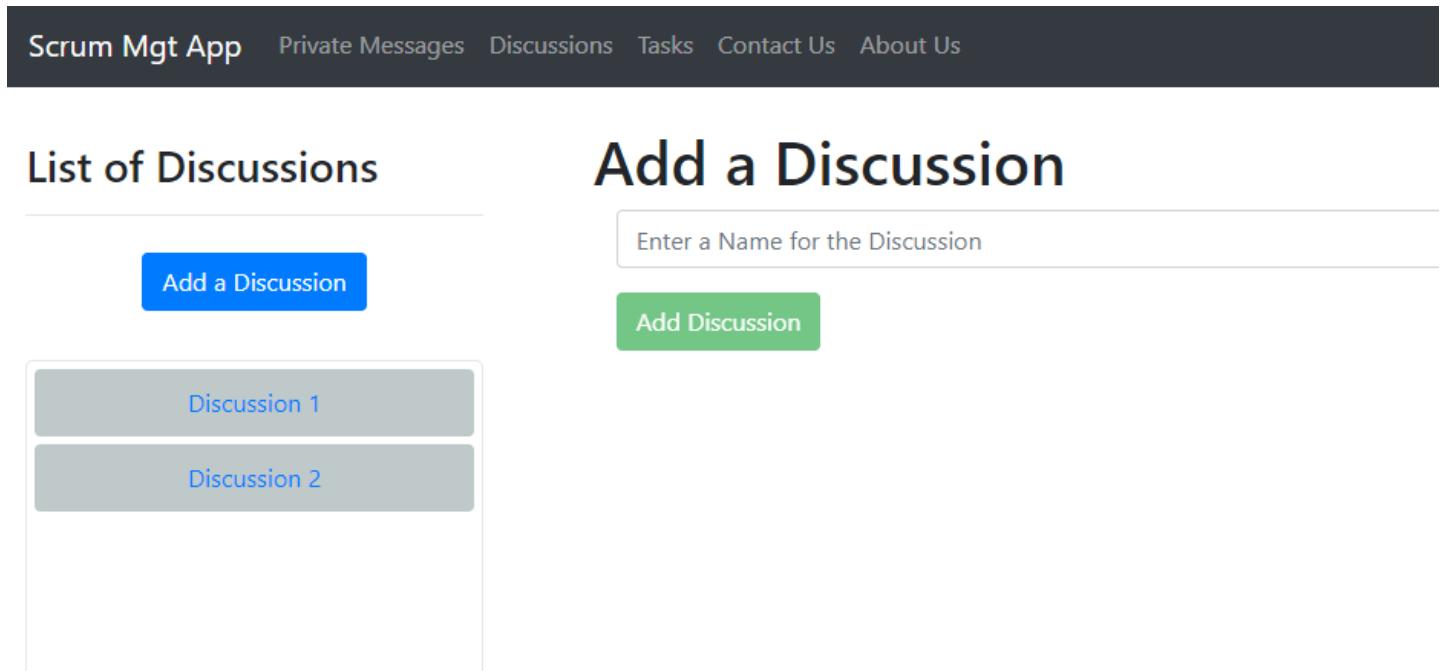
Name	Role	Update Role	Tasks
Test1			Add task

At the bottom of the page are buttons for '+', '<User Code>', and 'Send'.

Figure 36: Discussion Page Design

4.1.6 Discussion List Design

The discussion list is where all the discussions are stored, this way it is easier for the user to view, it allows the user to add a new discussion to their list. The user will be able to click on a discussion name and be redirected to the corresponding discussion page. Regarding the final design, I felt it was not necessary to add the task link to the page as the task bar on the navigation bar is clearer for the user to use.



The image shows two side-by-side screenshots of a discussion application. The left screenshot, titled 'List of Discussions', features a dark header with links for 'Scrum Mgt App', 'Private Messages', 'Discussions', 'Tasks', 'Contact Us', and 'About Us'. Below the header is a blue button labeled 'Add a Discussion'. To its right is a list of discussions with two items: 'Discussion 1' and 'Discussion 2'. The right screenshot, titled 'Add a Discussion', has a similar dark header. It includes a text input field with the placeholder 'Enter a Name for the Discussion' and a green button labeled 'Add Discussion' below it.

Figure 37: Final Discussion List Design



The image shows a wireframe of the initial discussion list design. At the top is a navigation bar with links for 'Homepage', 'Discussions', 'Private Messages', 'All Tasks', 'Contact Us', 'About Us', and a user account section with 'Hi, "Username"', 'Account Details', and 'Logout'. Below the navigation bar is a logo and the text 'Wesbite Name'. A title 'List of Discussions' is followed by a subtitle 'Table of discussions the user is a part of'. A table with three columns—'Discussion Name', 'Role', and 'Tasks'—is shown, though it currently contains no data. At the bottom of the table are three buttons: 'Leave Discussion', 'View Tasks', and 'Join New Discussion'.

Figure 38: Initial Discussion List Design

4.1.7 Private Message Design

This page is for users to engage in a private with another user on the website, this can be accessed either by the navigation page, which will firstly bring up the private message list, the user will then be able to click on each individual chat to then show this page. All messages and files will be saved in this chat, there will also be the tasks displayed for the user so they can easily refer to any details which need discussing. User's will also be able to call each other to discuss topics.

Once the user has finished communicating, they can press the list of messages button to be redirected back to the message list where they can communicate with others. The user can also click the video chat button to start a video call with the other user.

Private Messages

Select User to start a chat

Write your message... No file chosen

Test Two:
 upload-dir\d627070f-3d33-4974-8eab-61370095fd1f\107376204_1137707786671802_3958333698631713090_n.jpg Mar 31, 2021 02:16:03

Test Two:
Hello Mar 31, 2021 02:08:40

Me:
Hi Mar 31, 2021 02:08:31

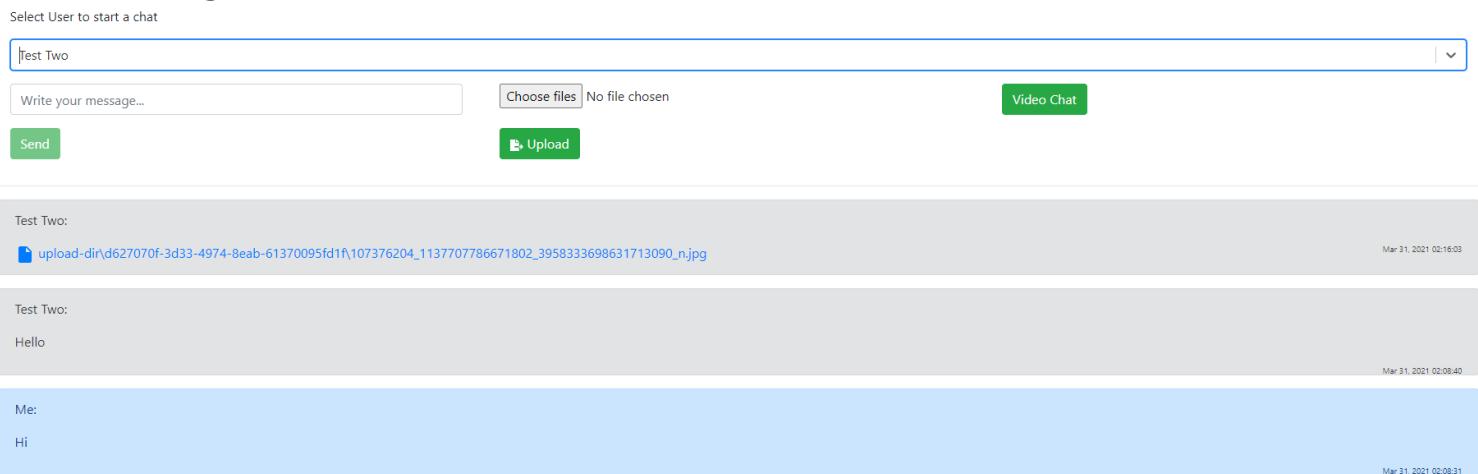


Figure 39: Final Private Message Design

Navigation Bar: Homepage Discussions Private Messages All Tasks Contact Us About Us Hi, "Username" Account Details **Logout**

Logo Website Name

Private Messages

List of Messages	Name of User	Call

Tasks:
Name of task Comment

+ <User Code> Send

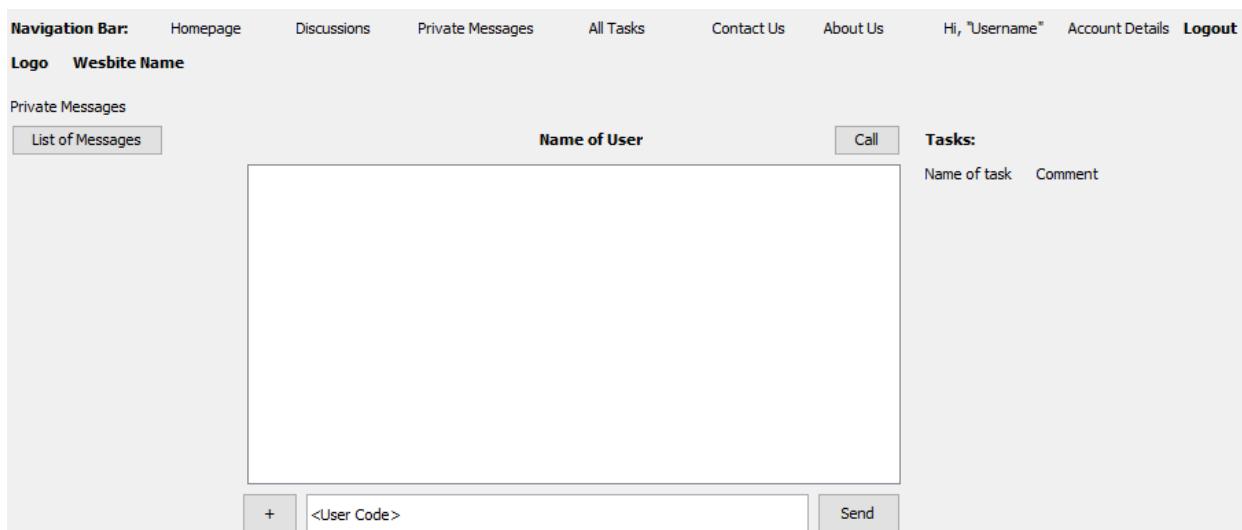


Figure 40: Initial Private Message Design

4.1.8 Call Design

The call page is where the user is re-directed from the discussions page or private messages. The final design of the call page was not as complex as the initial design due to time constraints. I was unable to implement the muting of microphones, also unable to implement the chat alongside the call.

Once the user clicks the video chat button, it will redirect them to another window within their chosen browser, this is where they will be able to view the other person's video output. The user can leave the call by pressing the 'close chat' button.

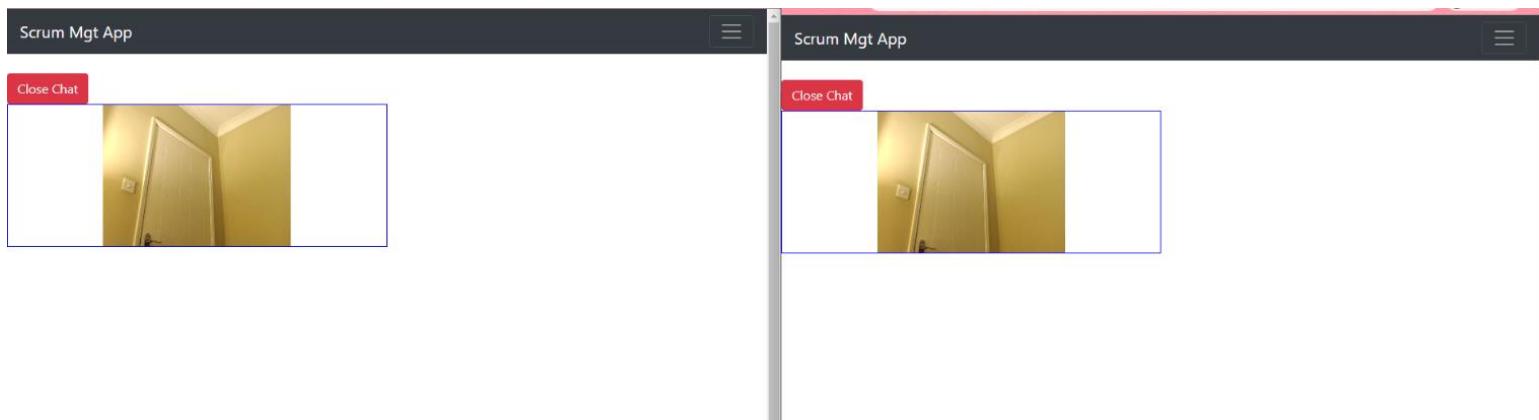


Figure 42: Final Video Call Design

A detailed wireframe of a web-based video call interface. At the top, there is a navigation bar with links for 'Homepage', 'Discussions', 'Private Messages', 'All Tasks', 'Contact Us', and 'About Us'. It also includes a 'Hi, 'Username'' greeting, 'Account Details', and a 'Logout' link. Below the navigation bar, there is a section for 'Team Members In Call:' with a table showing 'Name' and 'Role'. In the center, there is a large video preview area with a 'Call' button above it. To the right of the video area are buttons for 'Leave Call' and 'Discussion Chat'. At the bottom, there are buttons for 'Mute', 'Camera On/Off', a '+' sign, an input field for '<User Code>', and a 'Send' button.

Figure 41: Initial Call Design

4.1.9 About Us Design

The message areas represent where I will be writing information on the site to give the users insight and more context to the sites uses. I will also be going over the legal concerns with using the website, what data is stored and what their company can view if they choose to sign up.

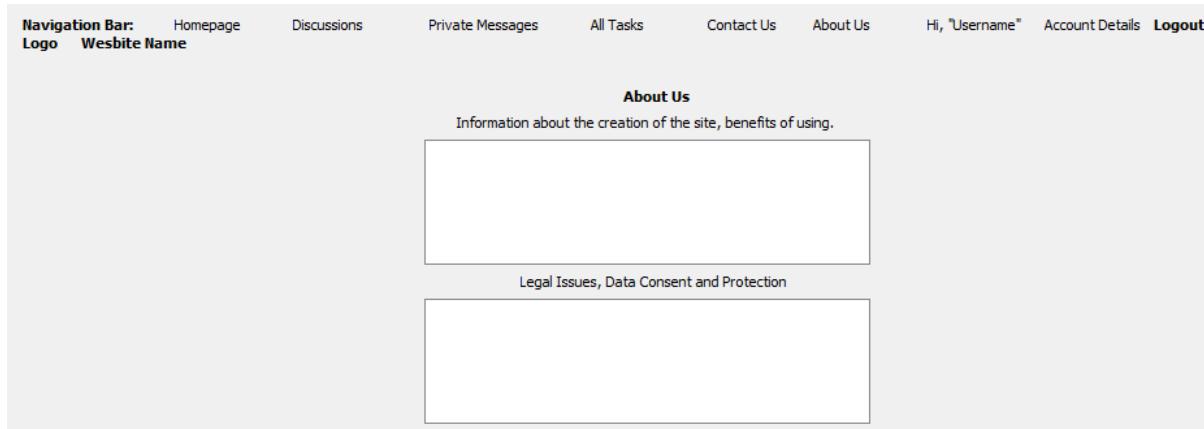


Figure 44: Initial About Us Design

A screenshot of the final 'About Us' page. The top navigation bar is dark with white text, containing links for 'Scrum Mgt App', 'Private Messages', 'Discussions', 'Tasks', 'Contact Us', and 'About Us'. The main content area has a light gray background. A large, bold heading 'About Us' is centered at the top. Below it is a subheading 'Information about the site.' followed by a paragraph: 'This is a scrum management tool, to help Scrum teams move all work over to an effective and efficient application.' Underneath this paragraph is the text: 'Everything for a scrum team is found here!' Then there is a section titled 'This includes the following:' followed by a bulleted list of features:

- Ability for users to register their own personal account, and login
- Please note, users can change their account details within the account details tab
- Ability for Scrum team leaders to assign roles:
- An admin role, recommended only for Scrum masters
- A regular user
- Ability to send private messages and video chat with another user
- Ability to create group discussions, in order to split up various tasks
- Ability for self task assignation, which is specified for each individual task
- All tasks are stored within the 'task' tab on the navigation bar

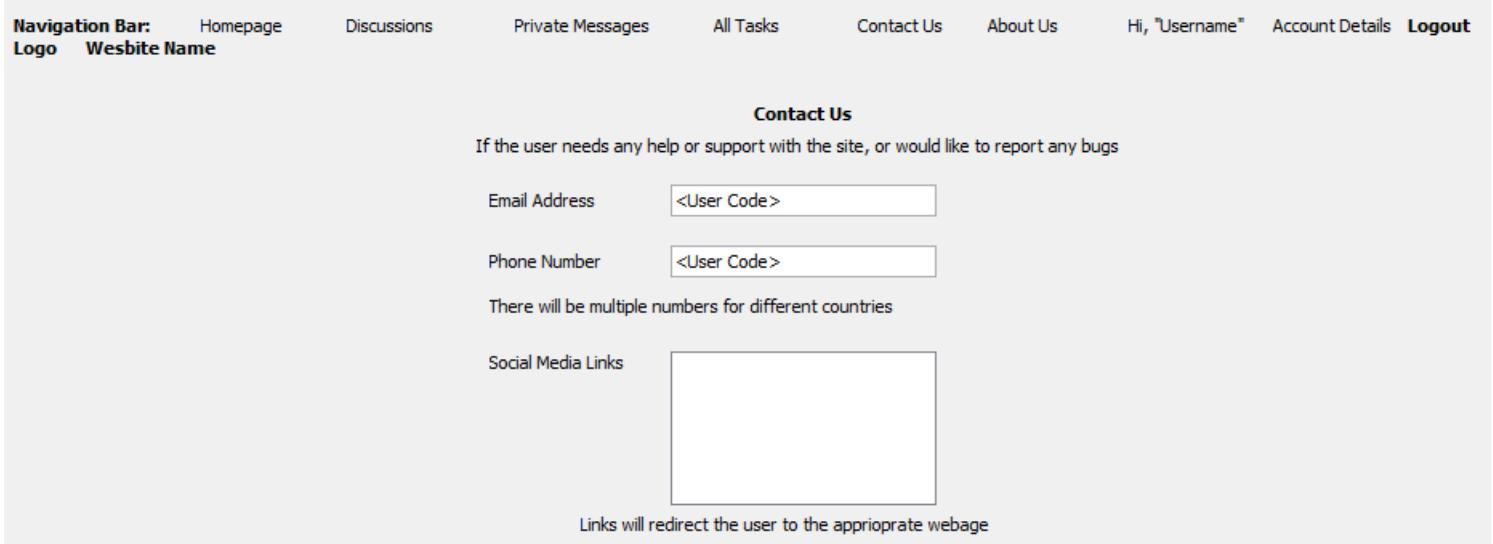
For help or support, please contact the following:

Phone Number: 07719439697
Email: scrummgtapp@gmail.com

Figure 43: Final About Us Design

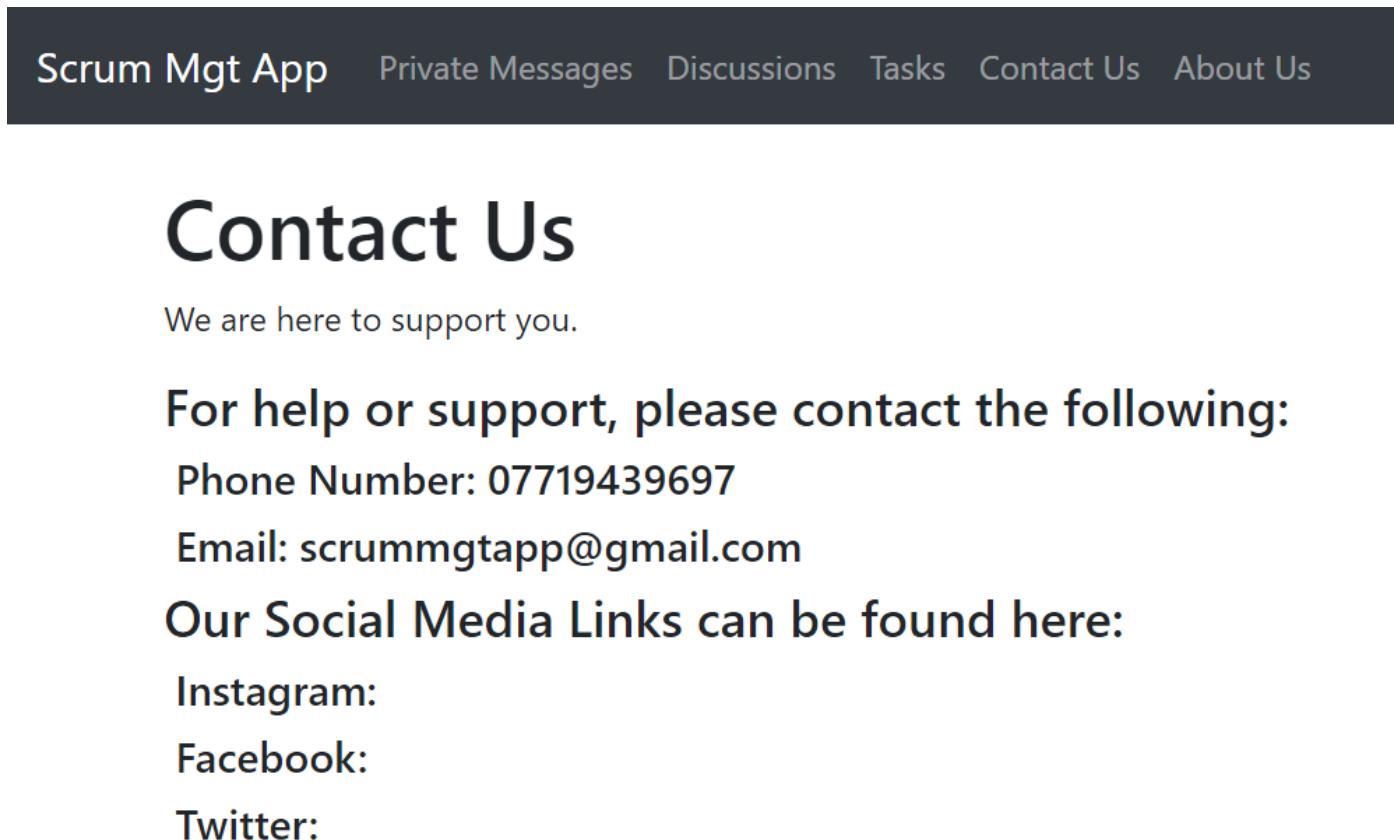
4.1.10 Contact Us Design

The contact page will be where all the contact information will be shown, this is in case a user discovers any bugs with the website or has any other concerns. I will include a business email, phone numbers, supported by different countries. I will also have social media links, for example, Facebook, Instagram, and twitter. These links will redirect the user to the appropriate webpage.



A screenshot of a web application's contact page. At the top, there is a navigation bar with links: Homepage, Discussions, Private Messages, All Tasks, Contact Us, About Us, Hi, "Username", Account Details, and Logout. Below the navigation bar, the page title is "Contact Us". A sub-instruction says, "If the user needs any help or support with the site, or would like to report any bugs". There are two input fields: "Email Address" and "Phone Number", both containing placeholder text "<User Code>". A note below states, "There will be multiple numbers for different countries". To the right of these fields is a large, empty rectangular box labeled "Social Media Links". At the bottom of the page, a note says, "Links will redirect the user to the appropriate webpage".

Figure 46: Initial Contact Us Design



A screenshot of the final contact page design. At the top, there is a dark navigation bar with links: Scrum Mgt App, Private Messages, Discussions, Tasks, Contact Us, and About Us. The main title is "Contact Us" in a large, bold font. Below the title, a sub-instruction says, "We are here to support you." A section titled "For help or support, please contact the following:" lists a phone number (07719439697) and an email address (scrummgtapp@gmail.com). Another section titled "Our Social Media Links can be found here:" lists three social media platforms: Instagram, Facebook, and Twitter.

Figure 45: Final Contact Us Design

4.1.11 Account Details Design

Account details page is for when users would like to change any aspect of their personal details. This is validated to make sure the user enters valid inputs; the user will also have the option to delete their entire account if they no longer have purpose for the website. When changing current details, the user will have to re-enter their password to confirm the changes. This will also apply if the user would like to delete their account.

Account Details

Username :	Test1
First Name :	Test
Last Name :	One
Email :	test1@hotmail.com
Phone Number :	23425432453

Change Password

New Password
Current password to save changes :	<input type="text" value="Current password"/> <input type="button" value="Save Changes"/>

Delete Account

Warning: Once you deleted
you won't be able to revert!

Figure 47: Final Account Detail Design

Navigation Bar:	Homepage	Discussions	Private Messages	All Tasks	Contact Us	About Us	Hi, "Username"	Account Details	Logout
Logo	Website Name								
Account Details									
Username	<input type="text" value="<User Code>"/>								
First Name	<input type="text" value="<User Code>"/>								
Second Name	<input type="text" value="<User Code>"/>								
Email	<input type="text" value="<User Code>"/>								
Phone Number	<input type="text" value="<User Code>"/>								
Password	<input type="text" value="<User Code>"/>								
Re-enter current password to save changes: <input type="text" value="<User Code>"/> <input type="button" value="Submit Changes"/> <input type="button" value="Delete Account"/>									

Figure 48: Initial Account Details Design

4.1.12 All Tasks Design

This is the page to display all the tasks the user has; these can be seen from all discussions and it acts as a general place where the user can review current responsibilities. The user can mark these tasks as completed, which will then remove them from the list. The task name and the ID will be shown so it is easier to distinguish what the task is about briefly. The user can also click the completed tasks to view all tasks which were completed by themselves or other users.

List of Task

Completed Tasks

ID	Task Name	Task Description	Discussion	Assigned User	
3	Add task	This is a task	Discussion 1	Test1	<button>Complete</button>
4	Complete task 2	Another task	Discussion 1		<button>Complete</button>

Figure 49: Final All tasks Design

The image shows a wireframe of a web application's interface. At the top, there is a navigation bar with links: 'Homepage', 'Discussions', 'Private Messages', 'All Tasks' (which is the active link), 'Contact Us', 'About Us', 'Hi, "Username"', 'Account Details', and 'Logout'. Below the navigation bar is a section titled 'Task Table List' with a table header row containing 'Task Name, ID', 'Description', and 'Discussion'. A large empty rectangular box represents the table body. At the bottom right of this section is a button labeled 'Completed'.

Figure 50: Initial All Tasks Design

4.2.1 Database Design

I have chosen a database that would be suitable to use for my website. The website needs a user's table to store all their information, which will be saved so the user's details can easily be accessed and viewed. Because the user will be sending messages, I have made a table for messages within a discussion and privately, this is to easily distinguish between them.

I have a tasks table to store the task ID, name, and description for each user's tasks, as these will need to be stored for every time a user logs in. I have also made a comment table so it can easily store them for all users to see. Logs table is also needed as it stores all the activities of a user in a discussion or private chat. This is important for admins to see as they need to see their team's progress. The final table I created are for the files and images sent to the discussions and private messages. Important files like tables and charts can be stored here, so will need to be accessible by every user.

Below I have created an entity relationship diagram, which shows how all the tables connect, showing all the relationships between each table.

4.2.2 Entity Relationship Diagram

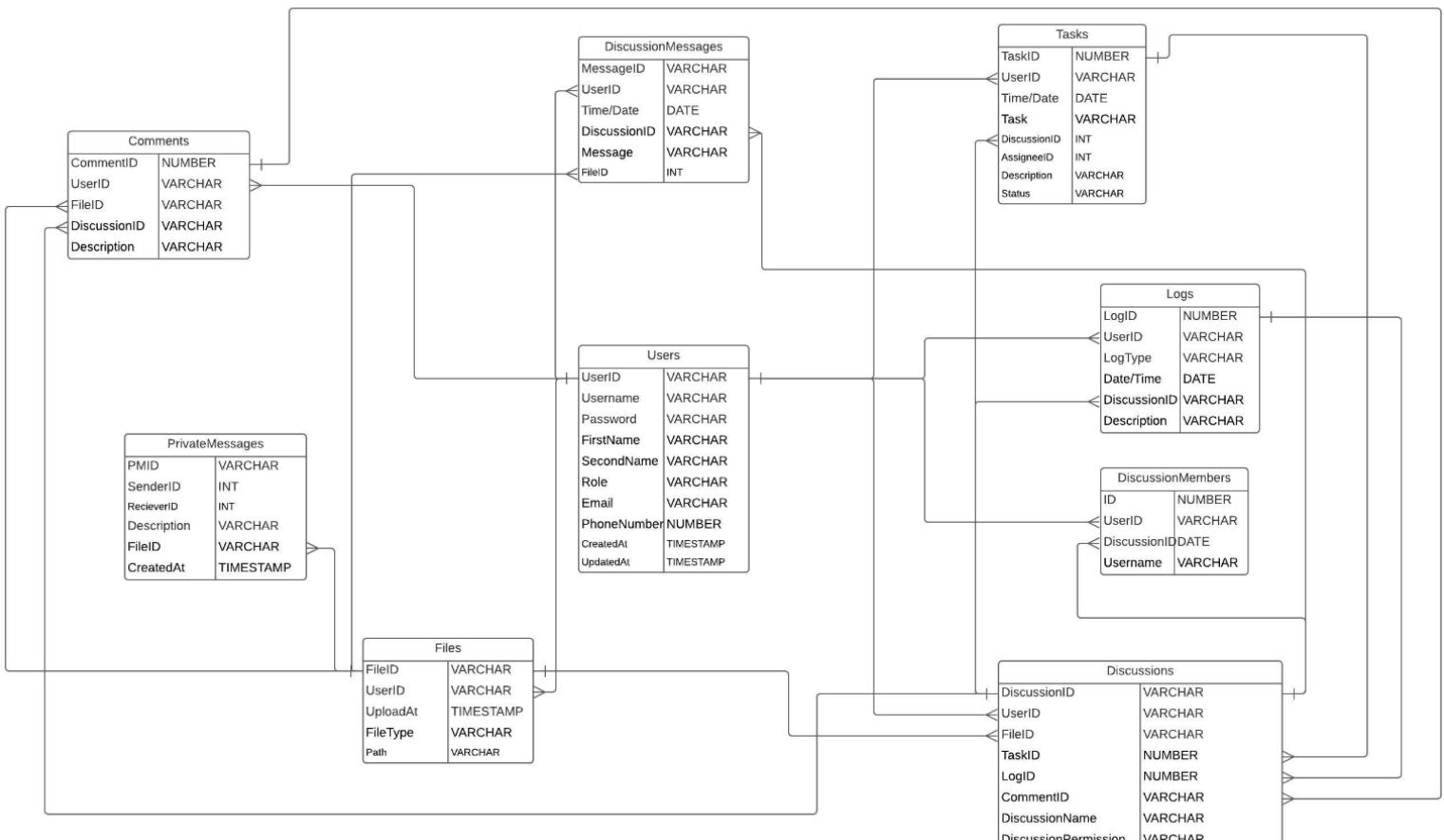


Figure 51: ER Relationship diagram

4.2.3 Data Model Diagram

This data model diagram was the initial design for how the database was originally planned, how the tables are structured and connected.

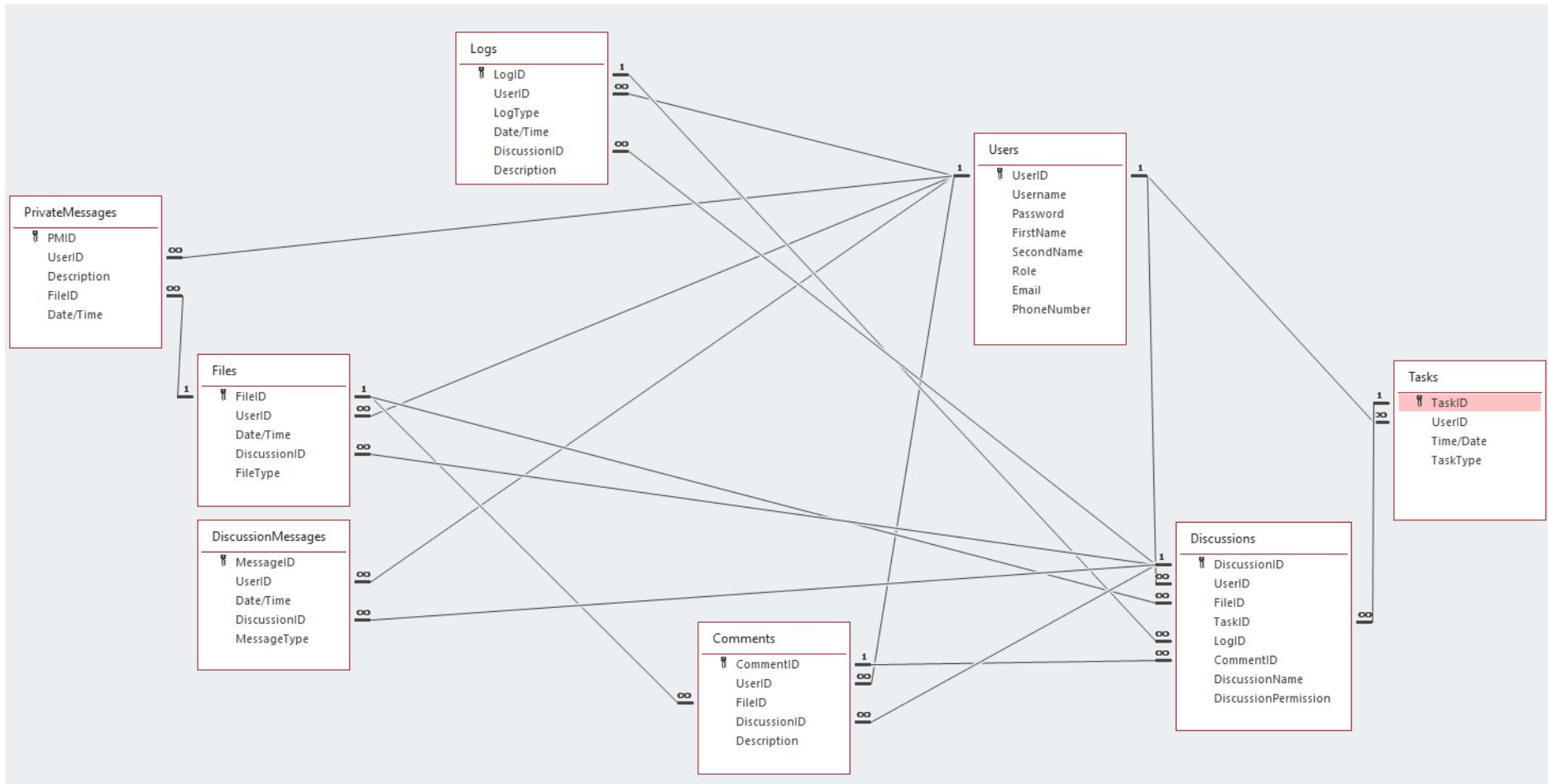


Figure 52: Data Model diagram of my database

4.2.4 Data Dictionaries

Table: Users

This table is used for storing all the users which have registered to the website, this includes the login credentials.

Table 19: User Table

Data	Data Type	Max field Size	PK/FK
UserID	VARCHAR	10	PK
Username	VARCHAR	10	
Password	VARCHAR	255	
FirstName	VARCHAR	50	
SecondName	VARCHAR	50	
Role	VARCHAR	20	
Email	VARCHAR	255	
PhoneNumber	NUMBER	11	
CreatedAt	TIMESTAMP	15	
UpdatedAt	TIMESTAMP	15	

Table: Tasks

This table is where all the tasks will be stored for each user.

Table 20: Task Table

Data	Data Type	Max field Size	PK/FK	Format
TaskID	NUMBER	8	PK	
UserID	VARCHAR	8	FK	
Time/Date	DATE			DDMMYYYY hh:mm:ss
Task	VARCHAR	10		
DiscussionID	INT	10	FK	
AssigneeID	INT	10		
Description	VARCHAR	255		
Status	VARCHAR	45		

Table: Discussion Messages

This table will be for storing all the messages between users within the discussions.

Table 21: Messages Table

Data	Data Type	Max field Size	PK/FK	Format
MessageID	VARCHAR	10	PK	
UserID	VARCHAR	10	FK	
Time/Date	DATE			DDMMYYYY hh:mm:ss
DiscussionID	VARCHAR	10	FK	
MessageType	VARCHAR	10		
FileID	INT	10	FK	

Table: Comments

This table will be for comments which users can leave on files sent in discussions.

Table 22: Comments Table

Data	Data Type	Max field Size	PK/FK
CommentID	NUMBER	10	PK
UserID	VARCHAR	10	FK
FileID	VARCHAR	10	FK
DiscussionID	VARCHAR	10	FK
Description	VARCHAR	255	

Table: Discussions

This table will store all the discussions.

Table 23: Discussions Table

Data	Data Type	Max field Size	PK/FK
DiscussionID	VARCHAR	10	PK
UserID	VARCHAR	10	FK
FileID	VARCHAR	10	FK
TaskID	NUMBER	10	FK
LogID	NUMBER	10	FK
CommentID	VARCHAR	10	FK
DiscussionName	VARCHAR	50	
DiscussionPermission	VARCHAR	50	

Table: Logs

This table is for logging any changes within the discussions.

Table 24: Logs Table

Data	Data Type	Max field Size	PK/FK	Format
LogID	NUMBER	10	PK	
UserID	VARCHAR	10	FK	
LogType	VARCHAR	30		
Date/Time	DATE			DDMMYYYY hh:mm:ss
DiscussionID	VARCHAR	10	FK	
Description	VARCHAR	50		

Table: Files

This table is to store all the files which are sent between users and discussions.

Table 25: Files Table

Data	Data Type	Max field Size	PK/FK	Format
FileID	VARCHAR	10	PK	
UserID	VARCHAR	10	FK	
UploadAt	DATE			DDMMYYYY hh:mm:ss
FileType	VARCHAR	10		

Table: PrivateMessages

This table will be to store all the user's private messages.

Table 26: Private Messages Table

Data	Data Type	Max field Size	PK/FK	Format
PMID	INT	10	PK	
SenderID	INT	10		
RecieverID	INT	10		
FileID	VARCHAR	10	FK	
CreatedAt	DATE			DDMMYYYY hh:mm:ss
Description	VARCHAR	255		

Table: DiscussionMembers

This table is where all a discussion's members are stored.

Table 27: Discussion Members Table

Data	Data Type	Max field Size	PK/FK	Format
ID	INT	10	PK	
UserID	INT	10	FK	
DiscussionID	INT	10	FK	
Username	VARCHAR	45		

4.3 Overall Program Architecture

Within the application, there are 2 projects that must be opened, these are Scrum App Master and Scrum System Master. App Master deals with all the webpages of the website and System master deals with all backend implementations including database.

This section of the report will show how the projects are put together and all files which are contained inside them.

4.4 Project Folder Structure

This is the application folder once it has been downloaded from WebLearn. It should contain:

- Javaee_app_db This is the database SQL scripts which will be needed to setup the database by creating tables.
- Requirements.txt

Shows the user the most appropriate requirements for their PC and software.

- Scrum-mgt-app-master

This is one of the projects required to run the application. It runs all frontend functions.

- Scrum-mgt-system-master

This is one of the projects required to run the application. It runs all backend functions.

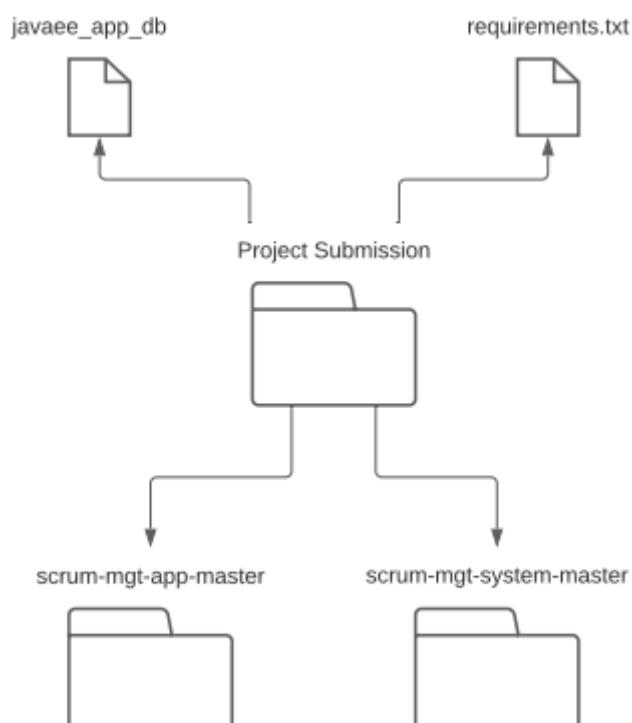


Figure 53: Project Folder Structure

Application Folders

4.5 Scrum Management System Master

Here I will be briefly describing all the files within the first project- Scrum-mgt-system-master. Here is the folder layout within the project. I chose not to include every individual file within the diagram as there is a lot.

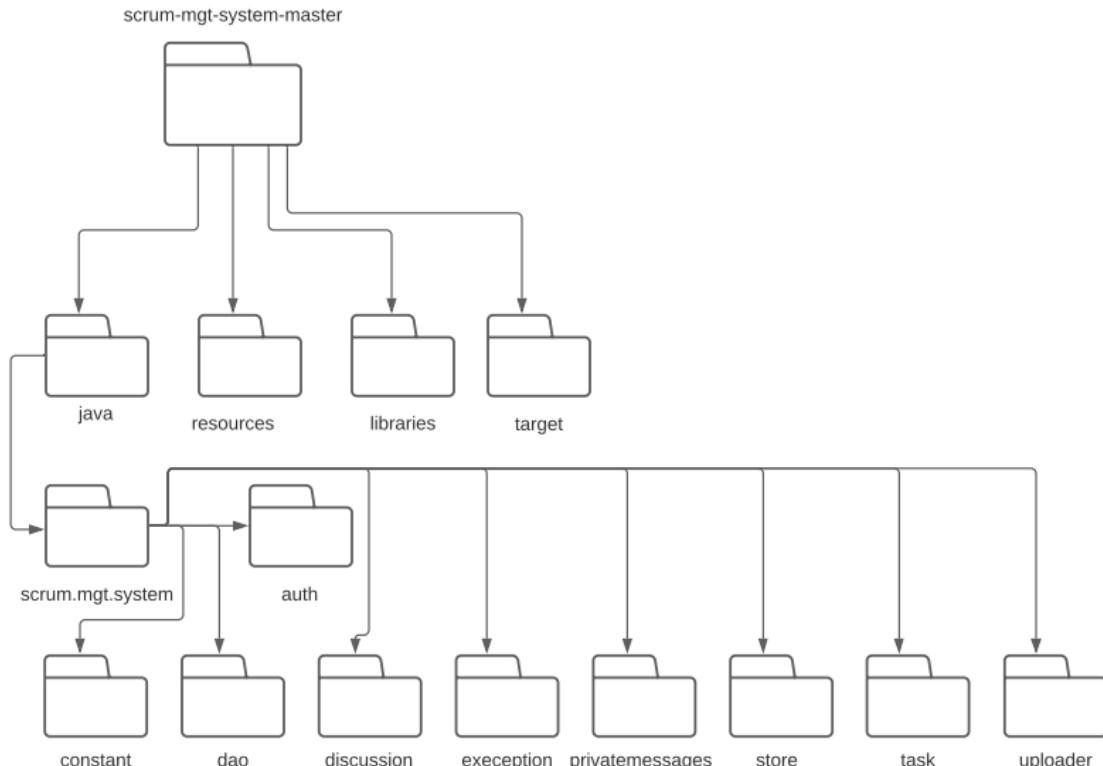


Figure 54: System Master software architecture

- Scrum-mgt-system-master

The whole project containing all files and folders

- Java 'Auth'

This folder contains all the java files of all backend functions. 'Auth' contains 2 folders named 'security' and 'user.'

Security

Security contains the authentication filters and validation rules for the website. Web configuration files for mapping is also defined within this folder.

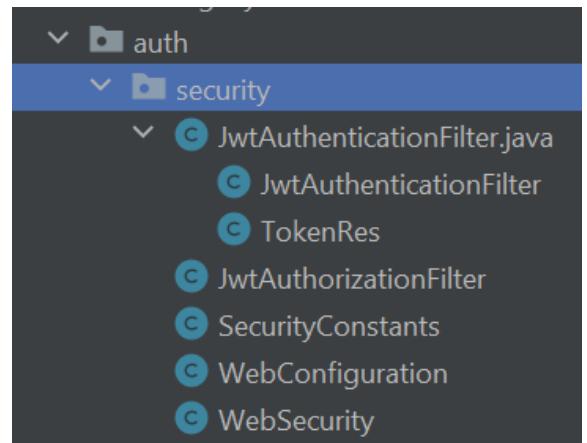


Figure 55: Auth folder with security folder and files

User

In user, there is a file named 'applicationUser.' Within this file, this is where all the getters and setters are implemented for user identification. The next file is named 'Constants.' This is where all the strings are implemented and the roles for admins and regular users. In the file 'UserController,' this is for database functions, like deleting users, encrypting user's passwords, and updating account details. Within the file 'UserDbHandler,' this is where the SQL queries are handled. In the file 'UserDetailsServiceImpl,' this file will receive the user's details by searching in the database. It will search for a record of the selected user when they try to authenticate.

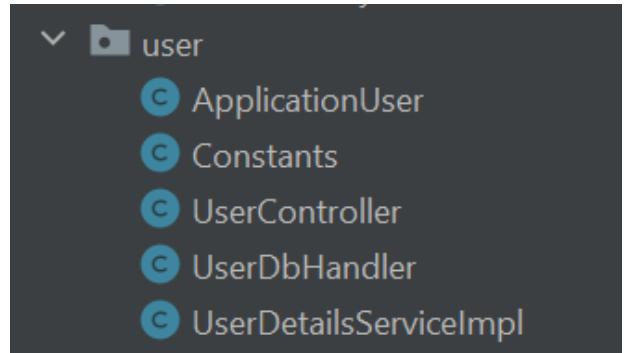


Figure 56: User folder with all files

Constants

This folder contains a single file named 'Constants.' This file just sets string messages which will stay the same throughout the entire project.

Dao

Dao is a folder containing the data access objects, this folder's main purpose is to connect the database to the application using mapping. All the files each map a separate part of the web application, for example, the discussions, private messages, users, and tasks.

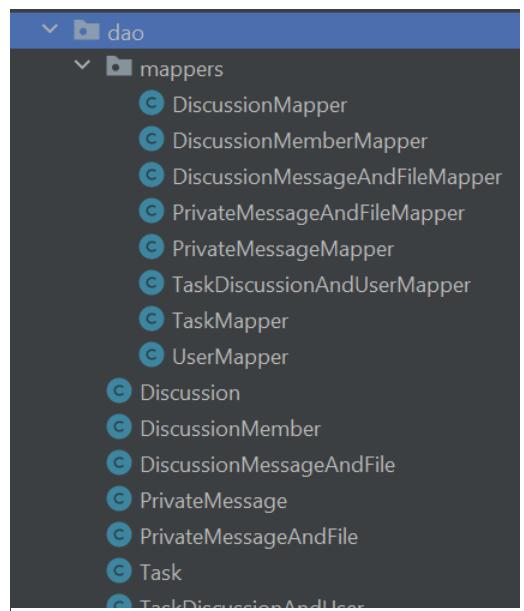


Figure 57: DAO folder with all files within

Discussion

Discussion folder contains files which control all the SQL queries for the discussions on the web application. This includes the members within the discussion and fetching data from the discussions.

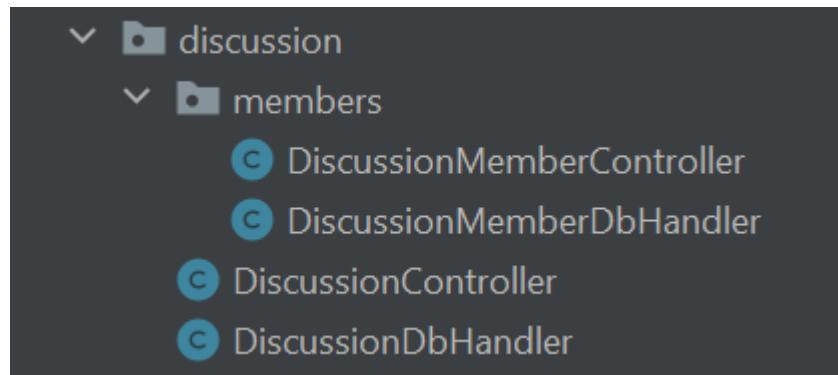


Figure 58: Discussion folder

Exception

Exception's folder is where all exceptions are implemented for the users, data reading, file storage and MySQL.

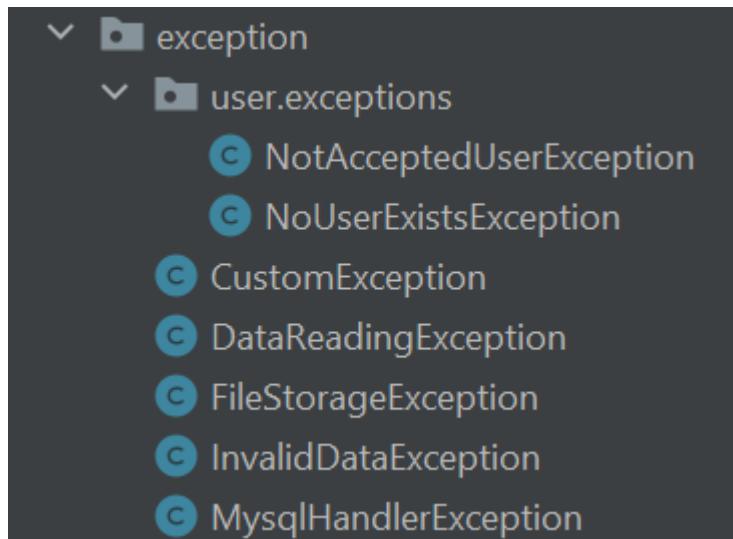


Figure 59: Discussion folder

PrivateMessages

This folder contains all the database controllers and handlers for the private messages. It allows for the fetching of private messages stored within the database and the function to add a file as a private message.

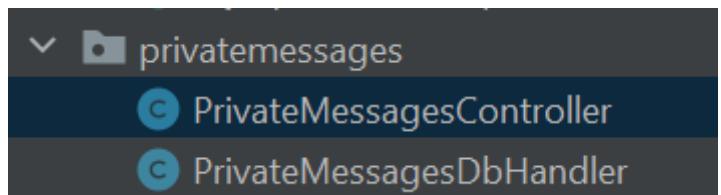


Figure 60: Private messages folder

Store

This folder is where the exceptions for user authorisation are stored. Also, for fetching IDs for tables.

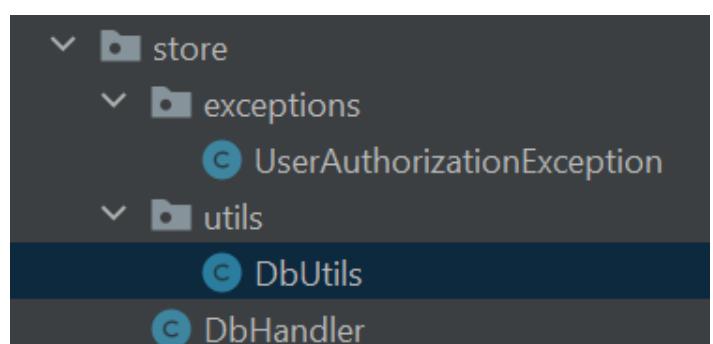


Figure 61: Store folder

Tasks

The task folder is specifically for creating the SQL queries for the tasks. This includes the ability to insert a task into a discussion, fetching the not completed tasks from the database and updating tasks which have been marked as complete.

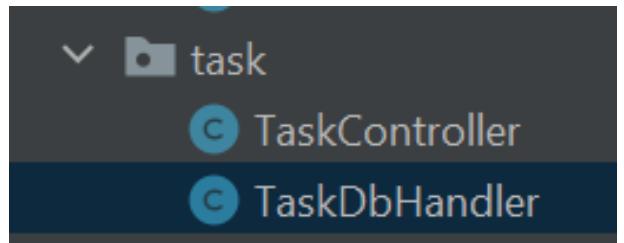


Figure 62: Task folder

Uploader

This folder is for storing files and attachments which have been sent within messages, being able to read files.

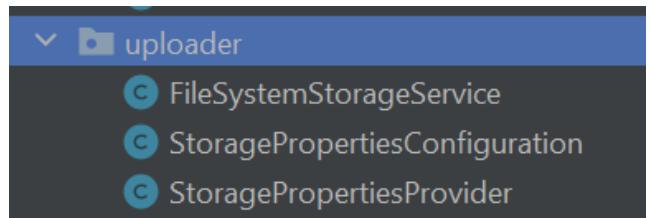


Figure 63: Uploader folder

- Resources

This folder is where the application properties are located. Within the '**application.properties**' file this is where the database URL is stored and the username and password for the MySQL login. Also, this is where the project file location is stored.

- Libraries

This is where Maven spring framework files are stored.

- Target

This is where the resources are stored, and generated sources are located.

4.6 Scrum Management App Master

I will also be describing the other project- Scrum-mgt-app-master.

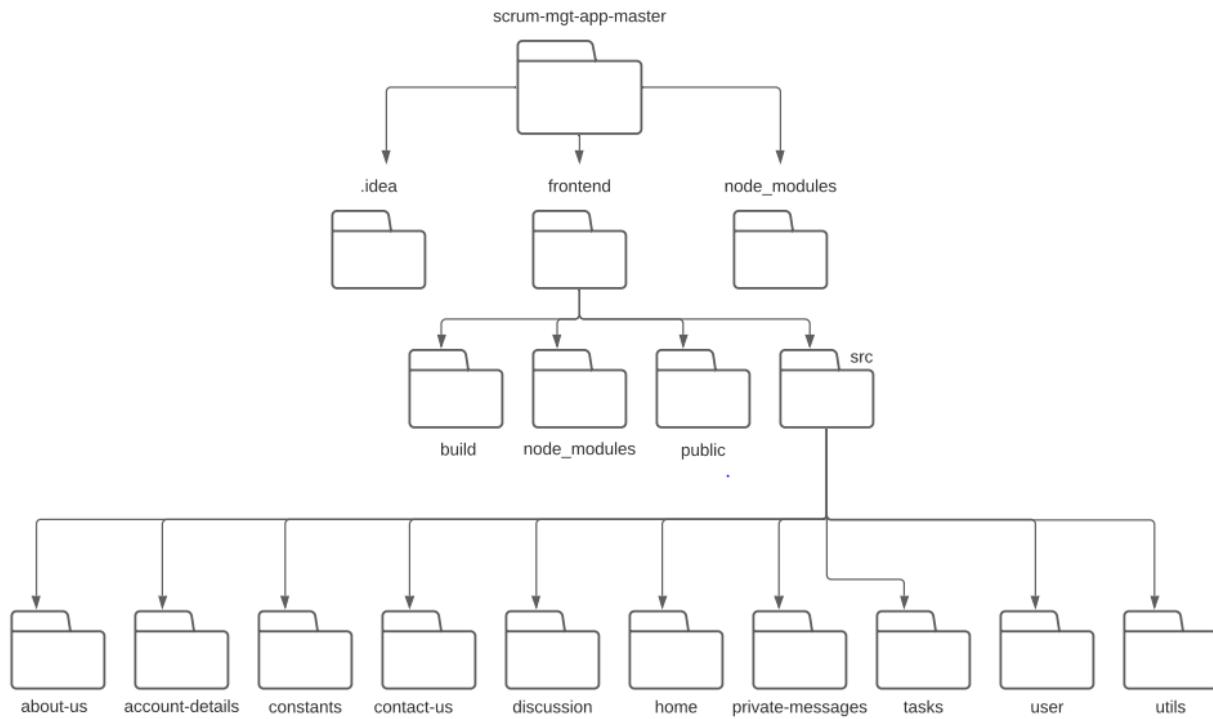


Figure 64: App Master Software Architecture

- .idea

This is where xml files are stored for all general configuration and settings within the frontend project.

- Node_modules

All the folders and files for node.js are stored within this folder.

- Frontend

Build

Within build, all images which are included on the website are stored, like the background image for the login page and register. This is also the same for the '**public**' folder.

Src

Within the source folder, this is where all the JavaScript for the frontend is stored. In 'app.js,' this is where the logout function is created, the page for the video calling and the index. These are stored outside of the folders which I have created. Each folder has a separate java script file in, creating

each individual page. All the JavaScript is what builds the frontend parts of the web application, which the user interacts with. This is where all the designing of the website is implemented, importing react for a cleaner look.

This is a screenshot of all my JavaScript files within Scrum-mgt-app-master.

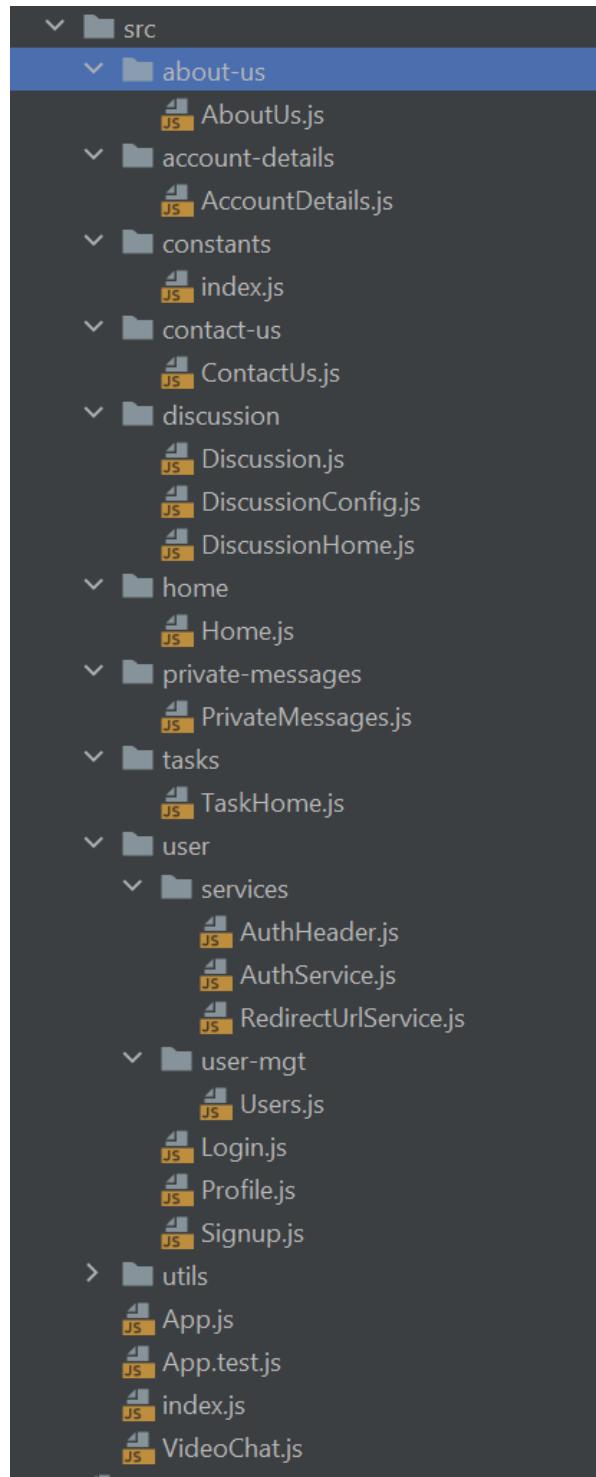


Figure 65: Source JavaScript folders and files for each webpage

Chapter 5: Implementation of the application

Within this chapter, minimum systems will be covered, to ensure that a user has insight on what is needed to successfully run the application. Also, software requirements are also listed to ensure a user has the correct software to allow successful deployment.

5.1 Minimum System Requirements

Disk Space: Minimum 1GB HDD or SSD

Processor:

Intel: i3 Processor, or higher

AMD: Ryzen 3 1300X or higher

Operating System:

Windows 7 or preferably the latest version, Windows 10

Browser: Google Chrome

Internet Connection: **Required**

5.2 Software System Requirements

User's will need these required software's installed on their PC:

- Node.JS, JavaScript Engine
- IntelliJ Community version, IDE platform
- MySQL, backend database connection

5.3 Installing

1. First, the user must download all necessary software to successfully run the project. This is listed within ‘5.2 Software System Requirements.’
2. Once all software is downloaded, the user must download the Project file from my Weblearn submission, within, will contain the 2 separate projects. App master, used for the frontend, System master for managing backend processes. Also, the SQL script will be included within the Project files.
3. Once all project files are downloaded onto the user’s PC, they must then open IntelliJ. Once IDE is open, the must open both projects within separate windows. The user is now ready to run the program.

5.4 Running the Program

5.4.1 Configuring MySQL Database

1. First, the SQL Script will need to be executed within MySQL to create all the tables and load all the test data within the database I have created. First, open MySQL and create a new connection. Then, give a suitable name for the connection, also there will be a prompt to enter the MySQL password created throughout the installation process. Once completed, test the connection, if successful, move onto the next step.

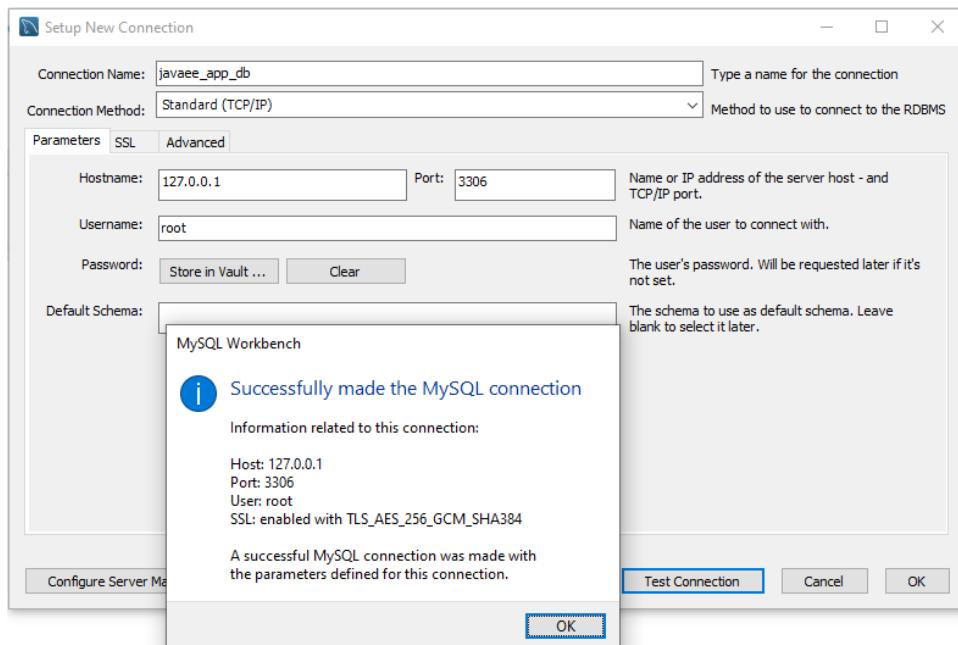
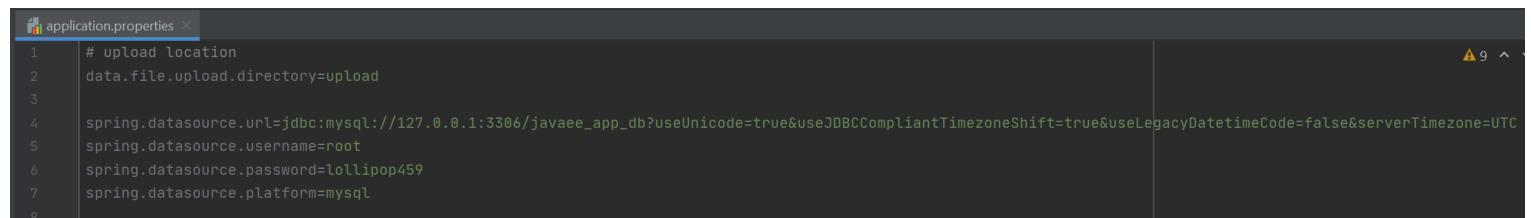


Figure 66: MySQL: Setting up new connection

2. Open IntelliJ, direct to the resources folder within the **System master** application, click on '**application.properties.**' Within 'Spring.datasource.username' and 'password', ensure that the correct username and password are entered. Also, ensure the correct database port is used within 'Spring.datasource.url'. Also, ensure that within 'Spring.datasource.url,' the database name is correct once you have created one on MySQL and executed the schema.



```
application.properties
1 # upload location
2 data.file.upload.directory=upload
3
4 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/javaee_app_db?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC
5 spring.datasource.username=root
6 spring.datasource.password=lollipop459
7 spring.datasource.platform=mysql
```

Figure 67: Properties file #1

3. Staying in the project file '**application.properties**,' within 'spring.resources.static-locations,' it must be set to the file location of where the project has been downloaded on the PC.

```
# upload dir static location
spring.resources.static-locations=file:C:\\\\Users\\\\Lauren\\\\Desktop\\\\Scrum App 7\\\\scrum-mgt-system-master
```

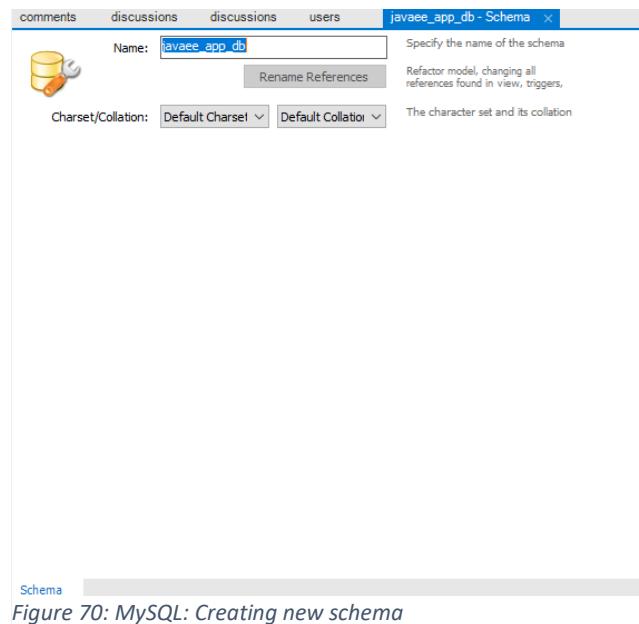
Figure 68: Properties file #2

4. Finally, staying in the project file '**application.properties**,' Ensure that the 'server.port' is set to 9090.

```
# serving port
server.port=9090
```

Figure 69: Properties file #3

- The backend system master will now be ready to be connected to database. Open MySQL and open the newly created connection. Once done so, create a new schema and give an appropriate name.



- Once new schema has been created, open the MySQL schema located within the project files, then copy the whole file and paste into a new query within the MySQL schema. Once executed, apply all changes to allow all tables and test data to be inserted within the database.

A screenshot of the MySQL Query Editor. The tabs at the top are "comments", "discussions", "discussions", and "users". The main area shows a numbered MySQL script:

```

1 •   SELECT * FROM javaee_app_db.comments;
2
3 -- Table structure for table `comments`
4
5
6 -- DROP TABLE IF EXISTS `comments`;
7 •  /*!40101 SET @saved_cs_client = @@character_set_client */;
8 •  /*!40101 SET character_set_client = utf8 */;
9 •  CREATE TABLE `comments` (
10    `commentID` int(10) NOT NULL

```

Figure 71: MySQL: Executing MySQL Script

5.4.2 Configuring Software

- Once both app master and system master are opened within IntelliJ, first open the system master, run the application. Open the ‘java’ file to find ‘Run Application’ at the bottom, or alternatively, press Control + Shift + F10.

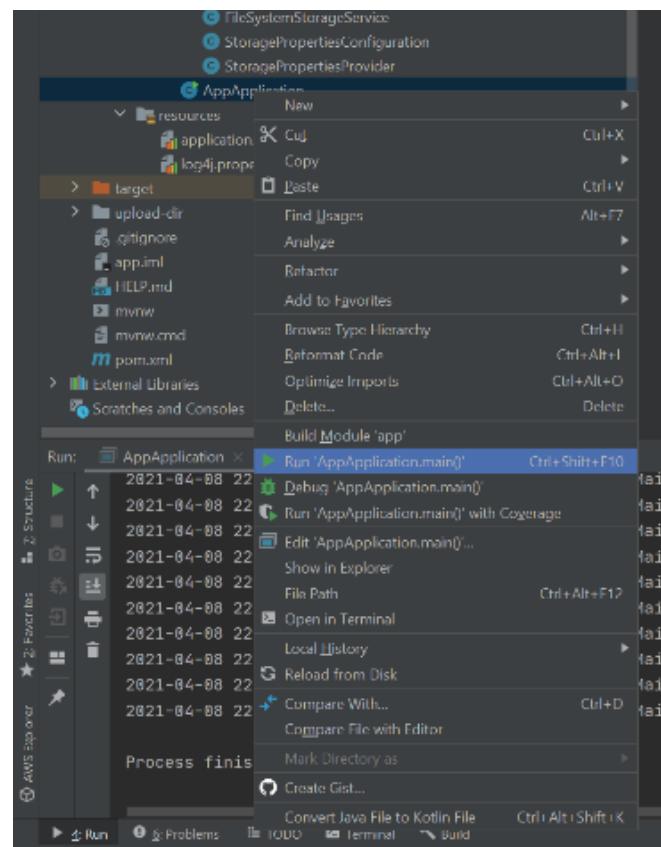


Figure 72: Running application

- The application will be successfully running when the terminal displays that both Tomcat and the server have been started.

```
Run: AppApplication
2021-04-08 23:24:16.923 INFO 23800 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JDBC repositories in DEFAULT mode.
2021-04-08 23:24:16.952 INFO 23800 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 25ms. Found 0 JDBC repository interfaces.
2021-04-08 23:24:17.448 INFO 23800 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 9090 (http)
2021-04-08 23:24:17.448 INFO 23800 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-04-08 23:24:17.448 INFO 23800 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.33]
2021-04-08 23:24:17.512 INFO 23800 --- [ restartedMain] o.a.c.c.Tomcat@localhost:[/] : Initializing Spring embedded WebApplicationContext
2021-04-08 23:24:17.513 INFO 23800 --- [ restartedMain] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1122 ms
2021-04-08 23:24:17.792 INFO 23800 --- [ restartedMain] o.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter, org.springframework.security.web.context.SecurityContextPersistenceFilter, org.springframework.security.web.header.HeaderWriterFilter, org.springframework.security.web.authentication.logout.LogoutFilter, org.springframework.security.web.savedrequest.SavedRequestHeaderFilter, org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter, org.springframework.security.web.authentication.www.BasicAuthenticationFilter, org.springframework.security.web.session.SessionManagementFilter, org.springframework.security.web.access.AccessControlFilter, org.springframework.security.web.servletapi.HttpServletHeaderFilter, org.springframework.security.web.savedrequest.HttpSessionRequestCacheFilter]
2021-04-08 23:24:17.981 INFO 23800 --- [ restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2021-04-08 23:24:18.132 INFO 23800 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2021-04-08 23:24:18.167 INFO 23800 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9090 (http) with context path ''
2021-04-08 23:24:18.169 INFO 23800 --- [ restartedMain] scrum.mgt.system.AppApplication : Started AppApplication in 2.066 seconds (JVM running for 2.451)
```

Figure 73: Starting Java Spring system master

- Once System master is running, open app master, 2 different commands will need to be executed to run Java React. These commands are as follows:

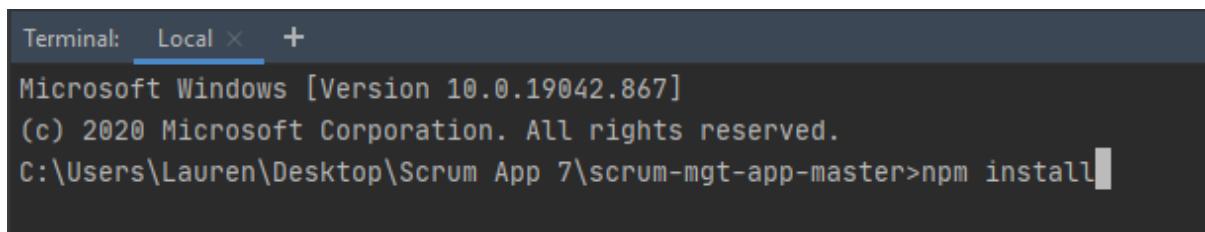
- **npm install**

This command installs all the necessary files and packages Java React is required to successfully start the frontend of the web application.

- **npm start**

This command will start the front-end application and deploy it to the default browser.

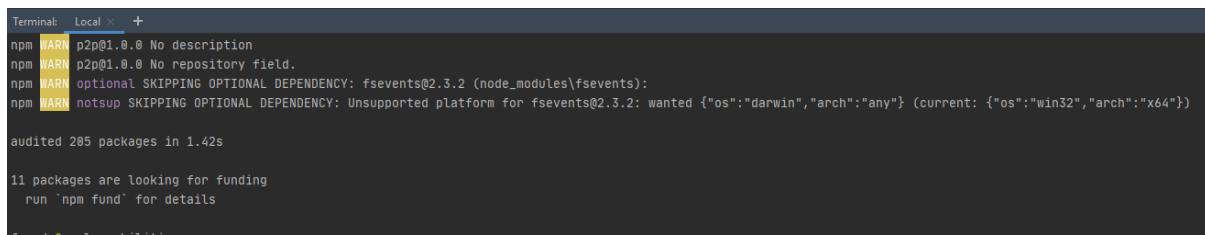
Please note, '**Npm install**,' will need to be executed first, after '**npm start**.' Also, 'npm install' only needs to be ran when the project is being first installed onto the PC. Therefore, only 'npm start,' will need to execute in future deploys of the application.



```
Terminal: Local +  
Microsoft Windows [Version 10.0.19042.867]  
(c) 2020 Microsoft Corporation. All rights reserved.  
C:\Users\Lauren\Desktop\Scrum App 7\scrum-mgt-app-master>npm install
```

Figure 74: npm install on local 1

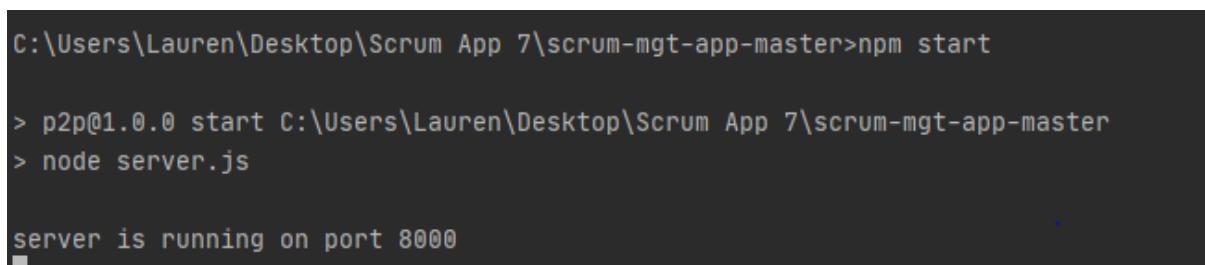
This screenshot shows the command '**npm install**' being typed into the terminal.



```
Terminal: Local +  
npm WARN p2p@1.0.0 No description  
npm WARN p2p@1.0.0 No repository field.  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):  
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})  
audited 205 packages in 1.42s  
11 packages are looking for funding  
  run 'npm fund' for details  
Found 0 vulnerabilities
```

Figure 75: npm install running

This screenshot shows the execution and result of '**npm install**' being successfully executed.



```
C:\Users\Lauren\Desktop\Scrum App 7\scrum-mgt-app-master>npm start  
> p2p@1.0.0 start C:\Users\Lauren\Desktop\Scrum App 7\scrum-mgt-app-master  
> node server.js  
  
server is running on port 8000
```

Figure 76: Successful execution of npm install

This screenshot shows ‘**npm start**,’ being executed and the server successfully running on port ‘8000.’

- Once server is started, another terminal will need to be opened. Type in ‘cd frontend,’ and execute. Then execute the 2 previous **npm** commands from step 3. This is to enable the video calling feature run successfully.

```
Terminal: Local x Local (2) x +  
  
C:\Users\Lauren\Desktop\Scrum App 7\scrum-mgt-app-master\frontend>npm start  
  
> my-app@0.1.0 start C:\Users\Lauren\Desktop\Scrum App 7\scrum-mgt-app-master\frontend  
> react-scripts start  
  
i [wds]: Project is running at http://192.168.1.69/  
i [wds]: webpack output is served from  
i [wds]: Content not from webpack is served from C:\Users\Lauren\Desktop\Scrum App 7\scrum-mgt-app-master\frontend\public  
i [wds]: 404s will fallback to /  
Starting the development server...
```

Figure 77: Executing cd frontend and npm start

This screenshot shows the application successfully being deployed. Once the server has been deployed, the default browser will open with the login page shown.

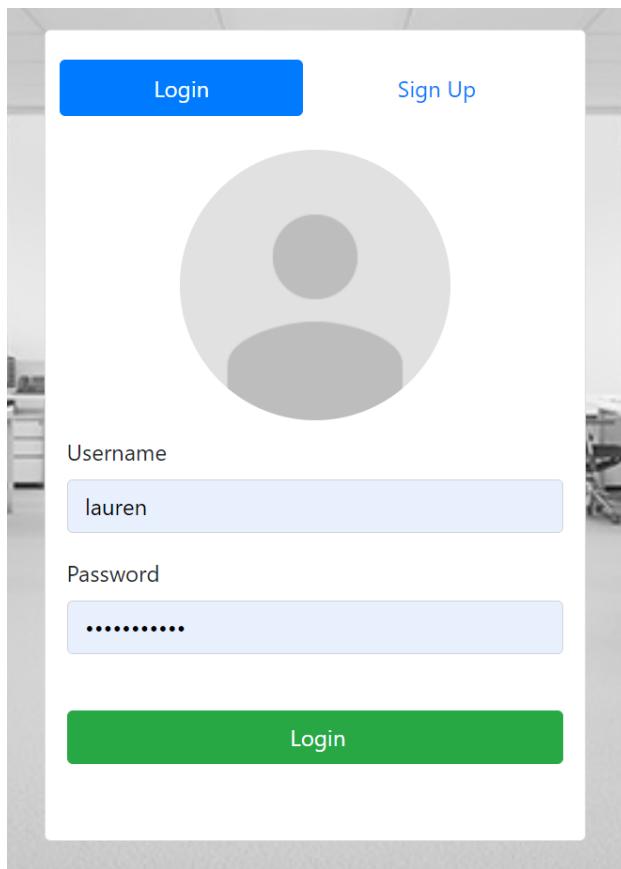


Figure 78: Login page UI

Chapter 6: Testing the Program

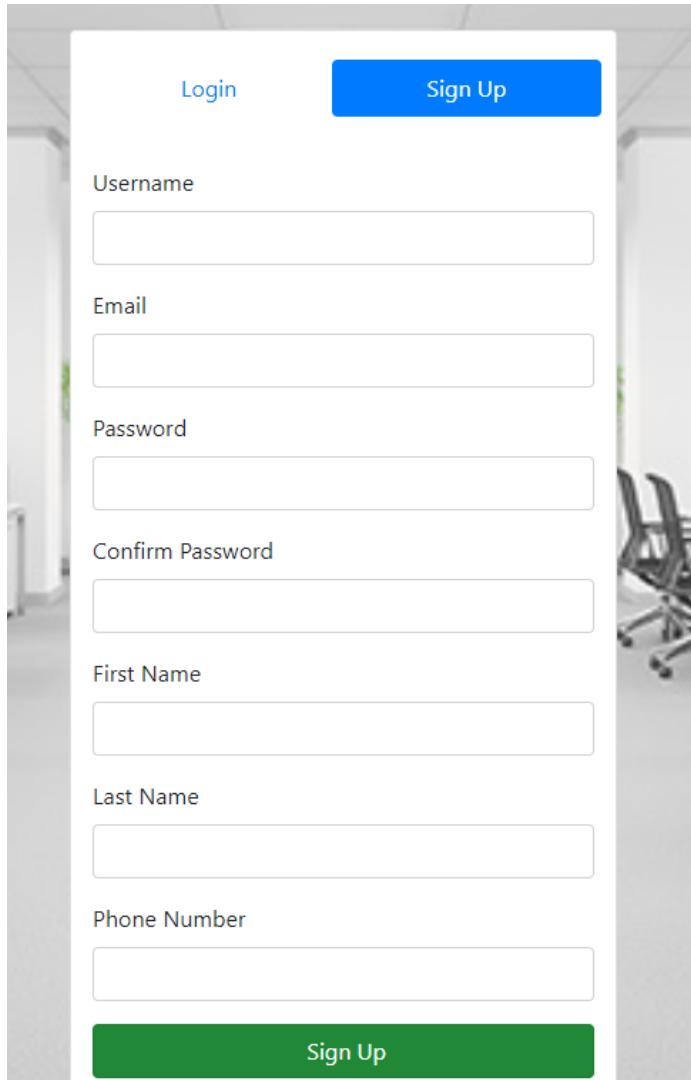
Testing is also shown within chapter 5, to fully demonstrate the applications capabilities and the expected outcomes of using certain functionalities, this also includes necessary screenshots.

6.1 Registration

First, I will be showing the registration page. User's will be able to click 'Sign up,' or 'Login.' Registration will allow for new users to create a profile by filling out all the required personal credentials. I will be showing some test's done within this page.

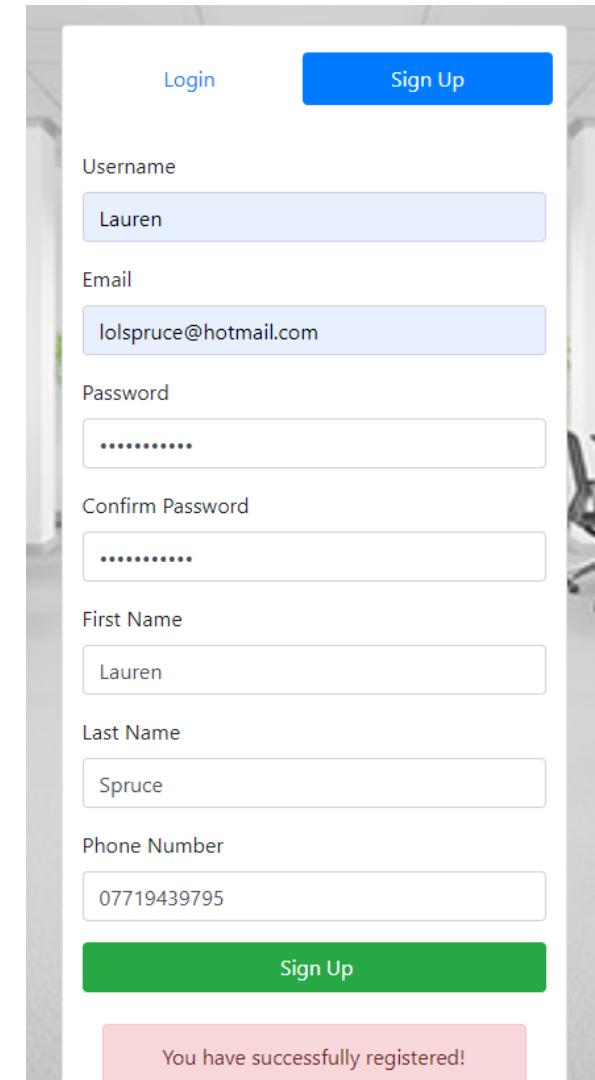
Below is the UI design of the Registration form. User's must enter all field's correctly, otherwise the following error messages will be displayed.

Once user has successfully entered all information correctly, the following dialogue box will appear on the below screenshot. The user's credentials will then be updated within the database on MySQL.



The image shows the initial state of a mobile application's registration screen. At the top, there are two buttons: 'Login' on the left and a larger blue 'Sign Up' button on the right. Below these are seven input fields with labels: 'Username', 'Email', 'Password', 'Confirm Password', 'First Name', 'Last Name', and 'Phone Number'. Each input field is represented by a white rectangular box with a thin gray border. At the bottom of the screen is a large green 'Sign Up' button with white text.

Figure 79: Registration page UI



The image shows the registration screen after a successful submission. The 'Sign Up' button at the top is now white with blue text. The input fields have been populated with sample data: 'Username' is 'Lauren', 'Email' is 'lolspruce@hotmail.com', 'Password' and 'Confirm Password' both show a series of six dots ('.....'), 'First Name' is 'Lauren', 'Last Name' is 'Spruce', and 'Phone Number' is '07719439795'. A green 'Sign Up' button with white text is at the bottom. A pink toast notification at the bottom right reads 'You have successfully registered! Please login to use the App.'

Figure 80: Successful registration

This screenshot shows the MySQL view of all current registered users.

Result Grid				Filter Rows:	Edit:	Export/Import:
	id	username	password	firstName	secondName	
▶	1	Test1	\$2a\$10\$xmpAE1gspW4nl5pXZK8DODad1411p...	Test	One	
*	2	test2	\$2a\$10\$8IIL4QoKyf.xQSzetO1OPOLWjiPrt7FlK...	Test	Two	
	NULL	NULL	NULL	NULL	NULL	

Figure 81: MySQL user's table

6.1.1 Password and Confirming password not matching

If the user does not enter the same password within each field, the following error dialogue message will be shown to the user. This prompts the user to retry until successful.

The screenshot shows a sign-up form with fields for Username, Email, Password, Confirm Password, First Name, Last Name, and Phone Number. The 'Password' and 'Confirm Password' fields both contain '.....'. The 'Sign Up' button is green. A red error message box at the bottom right says 'Password & Confirm Password in not matching!'.

Username	test4
Email	test4@hotmail.com
Password
Confirm Password
First Name	test
Last Name	4
Phone Number	43234234235

Sign Up

Password & Confirm Password in not matching!

Figure 82: Password validation

6.1.2 Leaving a field blank

If the user leaves any field blank the following error dialogue message will appear. This is due to all fields being required to create the account. For the screenshot below, I leave the phone number blank while trying to sign up.

The screenshot shows a 'Sign Up' form. At the top right are two buttons: 'Login' in blue and 'Sign Up' in white on a blue background. The 'Sign Up' button is larger and has a rounded rectangle border. Below the buttons are six input fields:

- Username:** A text input field containing 'Lozza'.
- Email:** A text input field containing 'lolspruce@hotmail.com'.
- Password:** A text input field containing a series of dots ('.....').
- Confirm Password:** A text input field containing a series of dots ('.....').
- First Name:** A text input field containing 'Lauren'.
- Last Name:** A text input field containing 'Spruce'.

Below the input fields is a large green 'Sign Up' button. At the bottom of the page, there is a pink rectangular callout box with the text 'Phone Number is required!' in red.

Figure 83: Field validation

6.1.3 Incorrect email format

If the user does not use the correct format for the email field it will display the following error message.

The form consists of the following fields:

- Username: Lozza
- Email: lauren (highlighted in light blue)
- Password: (redacted)
- Confirm Password: (redacted)
- First Name: Lauren
- Last Name: Spruce
- Phone Number: 0771323443

Sign Up button (green)

Error message: Email is invalid!

Figure 84: Email validation

6.2 Login

User's which have already successfully created an account will be able to use the login page. They must enter their Username and password they chose while registering to the site. Once the user has successfully logged in, it will redirect them to the homepage.

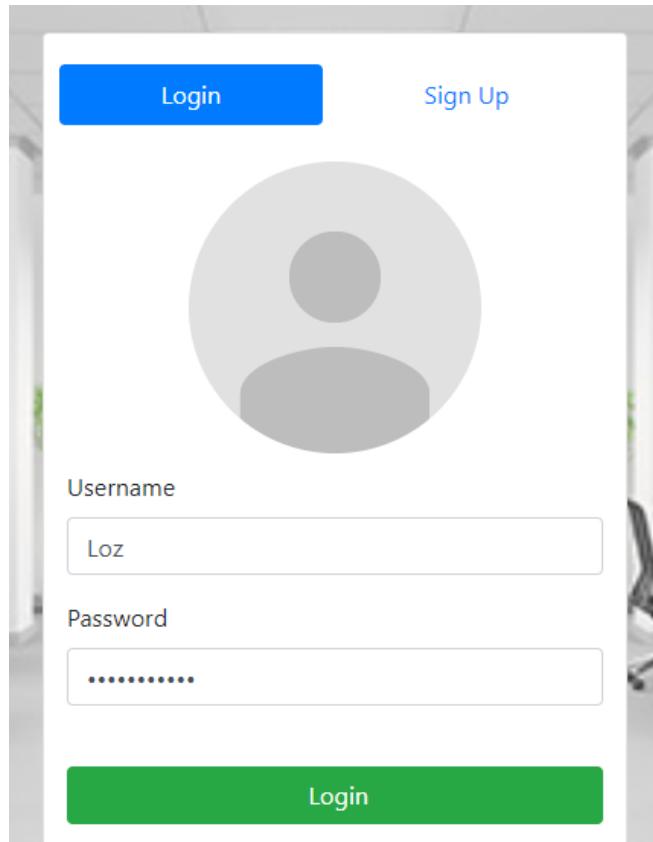


Figure 85: Login UI

6.2.1 Incorrect Username or Password

If the user types in the incorrect username or password the following error message will be displayed, prompting the user to try again.

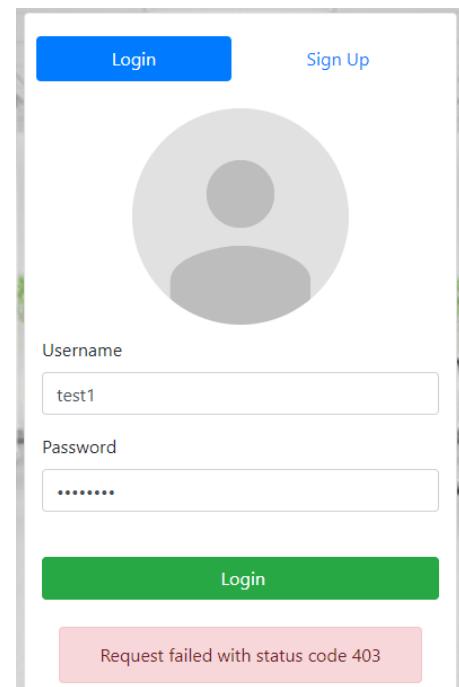


Figure 86: Username and password validation

6.3 Private Messages

To send a private message, click 'Private Messages' on the navigation bar. This will open the messages page. To send a message to a user, click on the drop-down box, this will display all the active users on the site. Select a user to send a message to, this will open the selected user's private chat. The 'write your message...' box will allow users to send messages to one another.

Private Messages

Select User to start a chat

Test Two

Write your message... Choose files No file chosen Video Chat

Send Upload

Test Two:
 upload-dir\d627070f-3d33-4974-8eab-61370095fd1f\107376204_1137707786671802_3958333698631713090_n.jpg Mar 31, 2021 02:16:03

Test Two:
Hello Mar 31, 2021 02:08:40

Me:
Hi Mar 31, 2021 02:08:31

Figure 87: Private message UI

Attachments can also be sent by clicking the choose files button. Once a file is selected, click the upload button to send the message. The attachment can be opened by clicking on the link in the chat, this will then open another window with selected file present.



Figure 88: Private message file attachment view

6.4 Discussions

To create a discussion, navigate to the ‘discussions’ page on the navigations tab, this will open the discussions page. A list of current discussions will be present on the side showing all created discussions. To create a discussion, the user will need to type in a suitable name for the discussion, then click on the ‘add discussion’ button.

A screenshot of the 'Scrum Mgt App' interface. At the top, there is a navigation bar with links for 'Private Messages', 'Discussions', 'Tasks', 'Contact Us', and 'About Us'. On the right, it shows 'Account Details' and 'Test1 [admin]'. The main area has two sections: 'List of Discussions' on the left and 'Add a Discussion' on the right. The 'List of Discussions' section is currently empty. The 'Add a Discussion' section contains a text input field labeled 'Enter a Name for the Discussion' and a green 'Add Discussion' button.

Figure 89: Discussions UI

This will then update the list of discussions to contain the newly created discussion. Once updated, click on the discussion and this will open the discussion page.

List of Discussions

Add a Discussion

Discussion 1

Figure 90: Discussion List

The discussion page will show the group chat, list of discussions, tasks assigned to that discussion and the team members list. To join the discussion, click on the ‘join discussion’ button to begin messaging and sending attachments.

The screenshot shows the 'Scrum Mgt App' interface. At the top, there's a navigation bar with links for 'Private Messages', 'Discussions', 'Tasks', 'Contact Us', and 'About Us'. On the right, it shows 'Account Details' and 'Test1 [admin]'. The main content area has three main sections: 'List of Discussions' on the left, 'Discussion 1' in the center, and 'Team Members' and 'Tasks' on the right.

List of Discussions: Contains a button 'Add a Discussion' and a list item 'Discussion 1'.

Discussion 1: A large central panel titled 'Discussion 1'. It features a message from 'Test1' saying 'Hello' at 'Mar 31, 2021 02:13:21'. Below this is a text input field with placeholder 'Write your message...', a file upload section with 'Choose files' and 'No file chosen', and a 'Send' button.

Team Members: A table with columns 'ID', 'Member', and 'Task'. One row is shown: ID 2, Member Test Two, Task Select... (with a dropdown arrow).

Tasks: A table with columns 'ID', 'Task Name', 'Status', and 'Assignee'. One task is listed: ID 1, Task Name Do paperwork, Status Completed, Assignee test2.

Figure 91: Join discussion #1

Once joined, it will update the team members list with the user's name and ID. Now the user will be able to message and add new tasks.

The screenshot shows the 'Scrum Mgt App' interface. At the top, there is a navigation bar with links for 'Private Messages', 'Discussions', 'Tasks', 'Contact Us', and 'About Us'. On the far right, it shows 'Account Details' and 'Test1 [admin]'. Below the navigation bar, the main area has a left sidebar titled 'List of Discussions' with a 'Add a Discussion' button. The main content area is titled 'Discussion 1' with 'Exit Discussion' and 'Delete Discussion' buttons. It displays a message from 'Test1' saying 'Hello' on 'Mar 31, 2021 02:13:21'. To the right, there is a 'Team Members' table and a 'Tasks' table. The 'Team Members' table has two rows: 'Test Two' and 'Test1'. The 'Tasks' table has one row: 'Do paperwork' assigned to 'test2'. At the bottom, there is a message input field, a file upload section, and 'Send' and 'Upload' buttons.

ID	Member	Task
2	Test Two	Select...
1	Test1	Select...

ID	Task Name	Status	Assignee
1	Do paperwork	Completed	test2

Figure 92: Join discussion #2

Attachments can also be sent within the group chats, where all members can view and save any important documents or images.

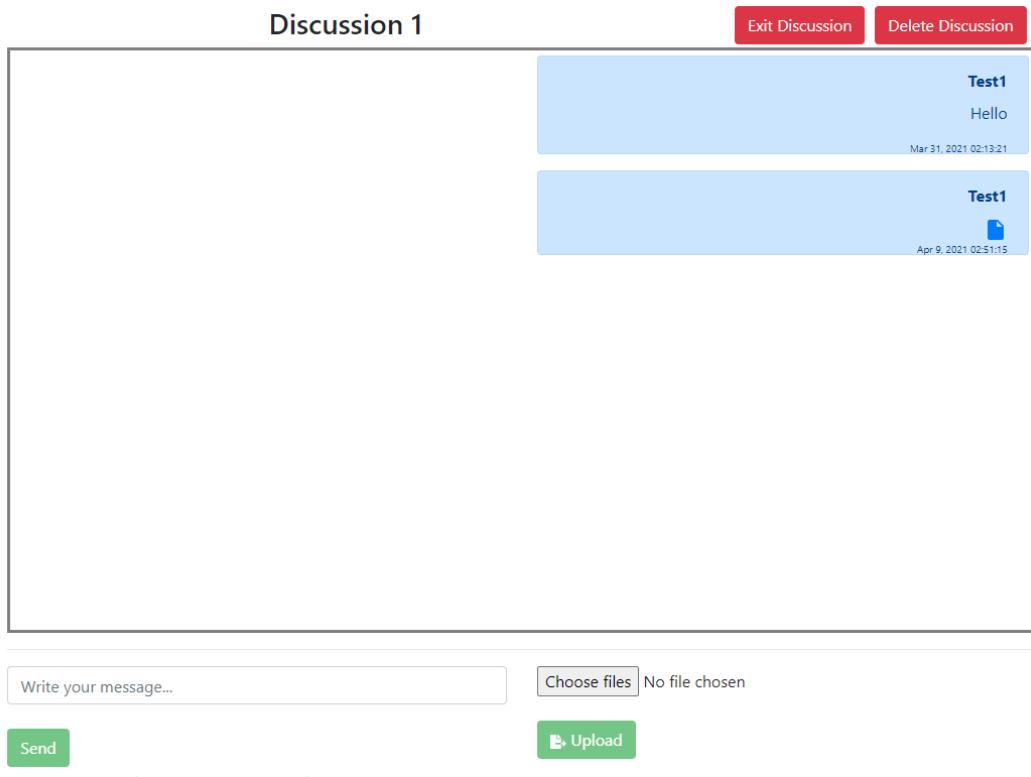


Figure 93: Sending messages in discussions

To exit a discussion, the user must click on the 'Exit discussion' button and this will remove them from the discussion.

Team Members

ID	Member	Task
2	Test Two	Select... ▾

Figure 94: Discussion members

Showing user 'Test1' no longer within the discussion.

6.4.1 Deleting a discussion

To delete a discussion, only admins can do so. This is done by clicking the ‘delete discussion’ button.

List of Discussions

Add a Discussion

Figure 95: Deleting the discussion

Refreshes the list of discussions back to 0.

Only admins can delete discussions, a regular user will not be able to access the delete discussion button, this is to ensure the preservation of vital information.

The screenshot shows the Scrum Mgt App interface. At the top, there is a navigation bar with links for Scrum Mgt App, Private Messages, Discussions, Tasks, Contact Us, and About Us. On the right, it shows Account Details for 'test2 [user]'. The main area is titled 'List of Discussions' and contains a blue 'Add a Discussion' button. Below this, a list item for 'Discussion 1' is shown, which has been deleted. The 'Discussion 1' section is empty. To the right of the list, there is a 'Team Members' table and a 'Tasks' table. The 'Team Members' table has one row with ID 2 and Member 'Test Two'. The 'Tasks' table has one row with ID 1, Task Name 'Do paperwork', Status 'Completed', and Assignee 'test2'. At the bottom, there is a message input field with placeholder 'Write your message...', a file upload field with 'Choose files' and 'No file chosen', and a 'Send' button.

Figure 96: User view of discussion

This screenshot shows Test2, a regular user, without the ‘delete discussion’ button.

6.5 Tasks

To create a task, user's must be active in a discussion to be able to create one. The plus button must be pressed, this will show the fields to insert a task name and description. All current tasks will be listed below with the status of the task.

The image shows a user interface for managing tasks. At the top, there is a blue header bar with the text "Add Tasks" and a blue plus sign button. Below this is a "Tasks" section with a table containing two rows of data. The table has columns for ID, Task Name, Status, and Assignee. The first row shows an ID of 1, Task Name "Do paperwork", Status "Completed", and Assignee "test2". The second row shows an ID of 2, Task Name "New Task", Status "Not Started", and Assignee "None". At the bottom of the interface are two buttons: "Add" and "Cancel".

ID	Task Name	Status	Assignee
1	Do paperwork	Completed	test2
2	New Task	Not Started	None

Figure 97: Task UI

Once all fields are filled out, the user must click the 'add' button which will update the list of tasks.

The image shows the same user interface as Figure 97, but with an additional row of data in the table. The table now has three rows: the first row with ID 1, Task Name "Do paperwork", Status "Completed", and Assignee "test2"; the second row with ID 2, Task Name "New Task", Status "Not Started", and Assignee "None"; and a third row with ID 3, Task Name "New Task", Status "Not Started", and Assignee "None".

ID	Task Name	Status	Assignee
1	Do paperwork	Completed	test2
2	New Task	Not Started	None
3	New Task	Not Started	None

Figure 98: Adding tasks

The task can be viewed from the tasks page on the navigation bar, this will be shown in the non-completed tasks and assigned user will be blank, as no user has self-assigned this task.

List of Task					Completed Tasks
ID	Task Name	Task Description	Discussion	Assigned User	
2	New Task	This is a description	Discussion 1		Complete

Figure 99: Task table UI

I will now show a user self-assigning the task. On the team members list, if the ‘select’ drop down list is clicked, the task will be shown here. Once clicked, it will show the task list to be updated and the status saying ‘new,’ along with the assignee.

Team Members

ID	Member	Task
2	Test Two	<input type="button" value="Select..."/>
1	Test One	<input type="button" value="Select..."/>

Add Tasks
+

Tasks

ID	Task Name	Status	Assignee
1	Do paperwork	Completed	test2
2	New Task	New	Test1

Figure 100: Assigning a task

Once the user wants to mark the task as completed, they must go over to the task page, they must click the ‘complete’ button. The user will be shown a dialogue message asking them to confirm this option.

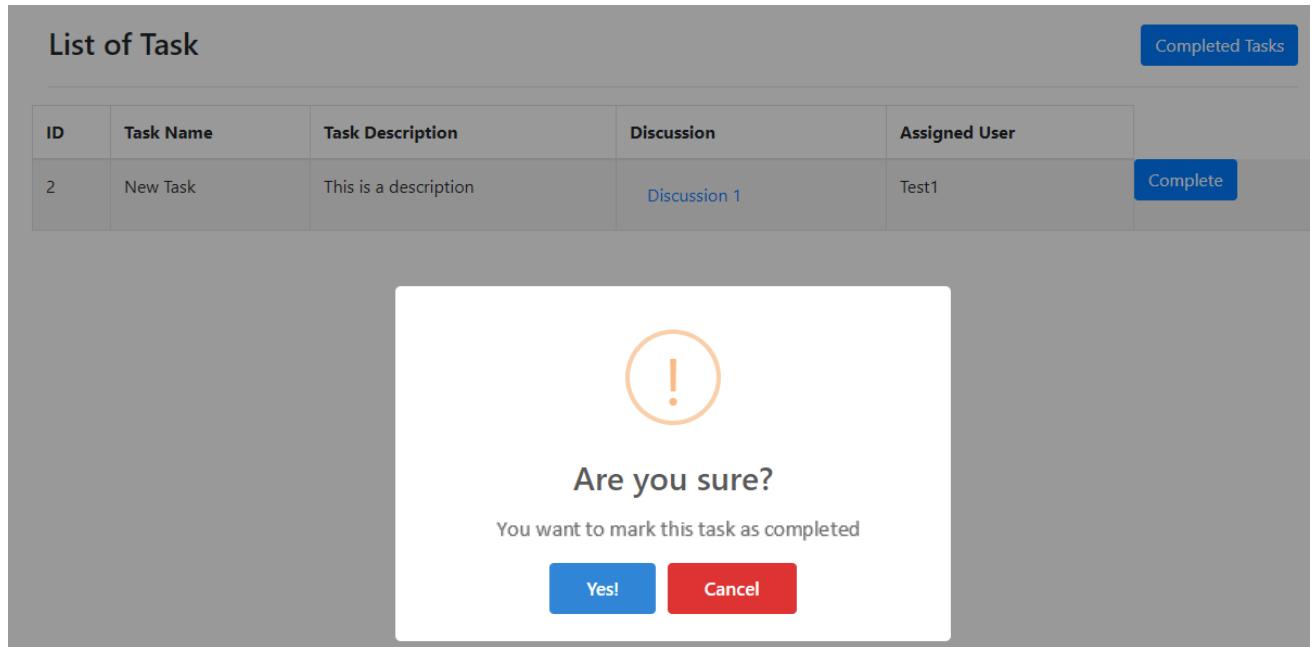


Figure 101: Marking a task as completed

Once confirmed, the user will need to click on the ‘completed tasks’ button to view their completed task within the list.

ID	Task Name	Task Description	Discussion	completed by
1	Do paperwork	just do it	Discussion 1	test2
2	New Task	This is a description	Discussion 1	Test1

Figure 102: Completed tasks

6.6 Private Video Calling

To privately call another user, first the user must go to the private messages and select 'Video chat' button. Once clicked, it will redirect the user to a separate browser page. It will ask if the user wants to enable their microphone and camera. Once both users are on the video chat it will show the opposite user's video.

Once the users are both finished in the video chat, they can close the page or click the 'close chat' button.

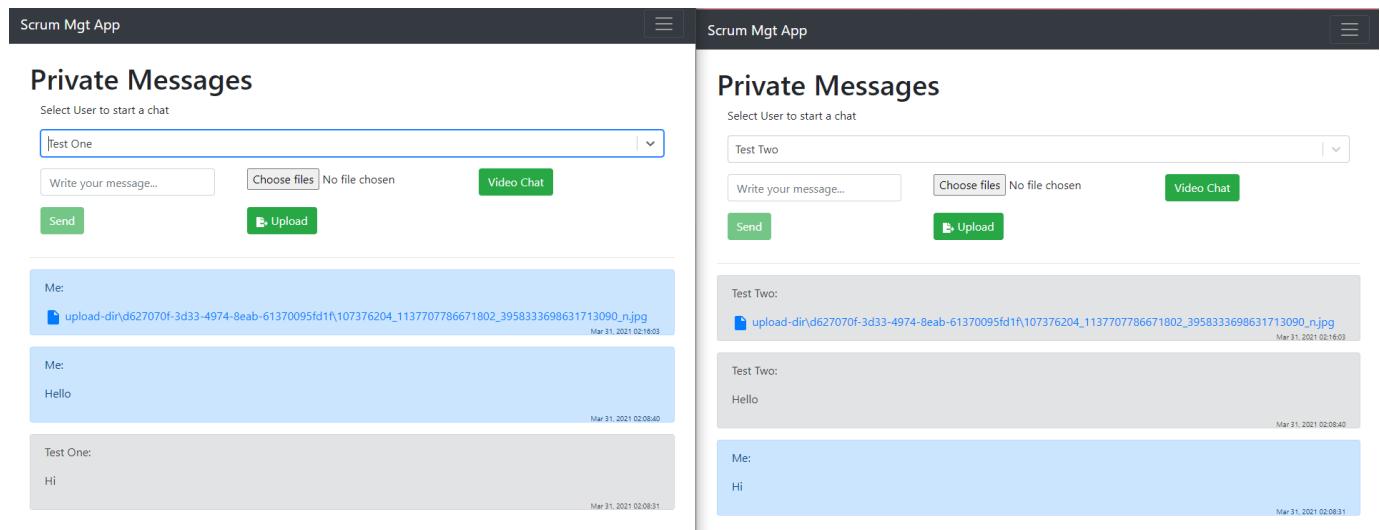


Figure 103: Private Message UI with private video calling

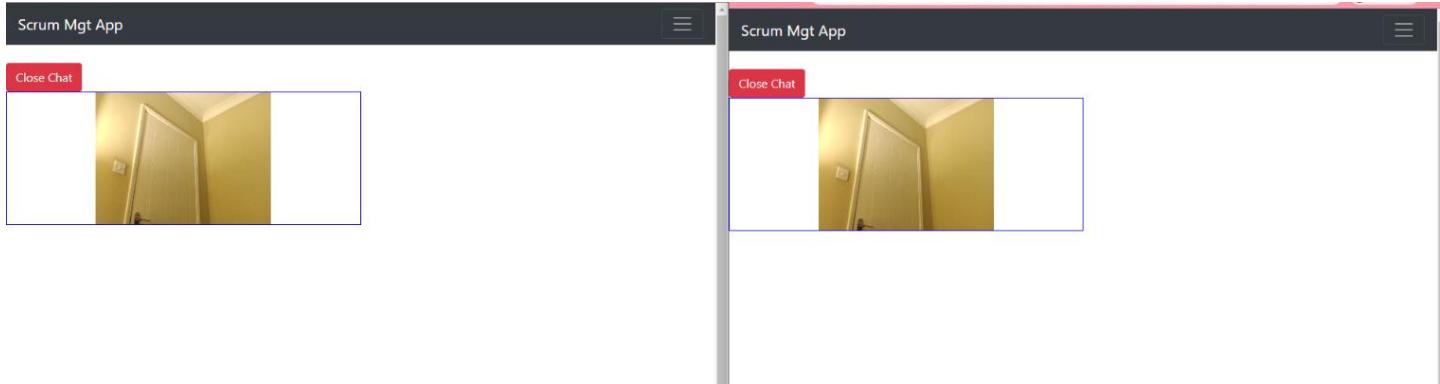


Figure 104: Private video call UI

6.7 Roles

Within roles, I let there be 2 roles. These include a regular user and an admin. The difference is that admins can delete discussions whereas regular users cannot. I plan to add more functionality within this role towards the final submission.

6.7.1 Changing role to admin

To change a role to admin, the user must do this within the MySQL database. I chose to keep it this way so therefore no non-authorised users can grant themselves access, improving the website's security. I may add a way to complete this within the website for the final submission.

First, the account will always be set to a regular user as default, as seen in this screenshot. To check what permission a user has this can be done in the 'Users' table in the database, or simply on the website in the top right-hand corner next to the user's name.

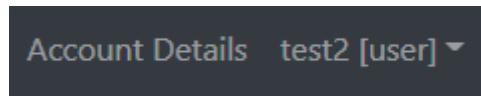


Figure 105: User details

To change to admin, go to MySQL and select the correct database. Within this side panel all tables will be shown. Click on the user's table and select the table view.

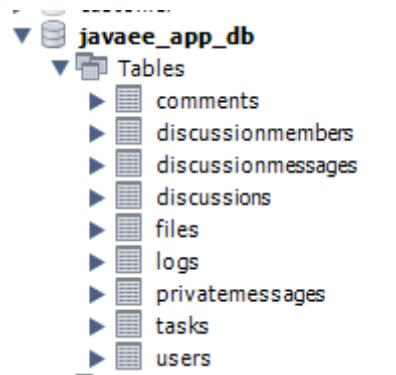


Figure 106: MySQL table view

Here is the view of all the users currently registered in the database. The role data must be changed from 'user' to 'admin,' this is done by editing the data and typing it in.

		firstName	secondName	role	email	phoneNumber	createdOn
▶	ODad1411lp...	Test	One	admin	test1@hotmail.com	23425432453	202
...	\LWjiPrt7FlK...	Test	Two	user	test2@hotmail.com	435345435534	202
*		NULL	NULL	NULL	NULL	NULL	NULL

Figure 107: MySQL: User data

Once admin has been changed, you must apply the changes by clicking ‘apply’ in the bottom right-hand corner.

Figure 108: MySQL: Changing user to admin

MySQL will tell you to review the changes, simply click apply to confirm changes.



Figure 109: MySQL: Confirming applied changes #1

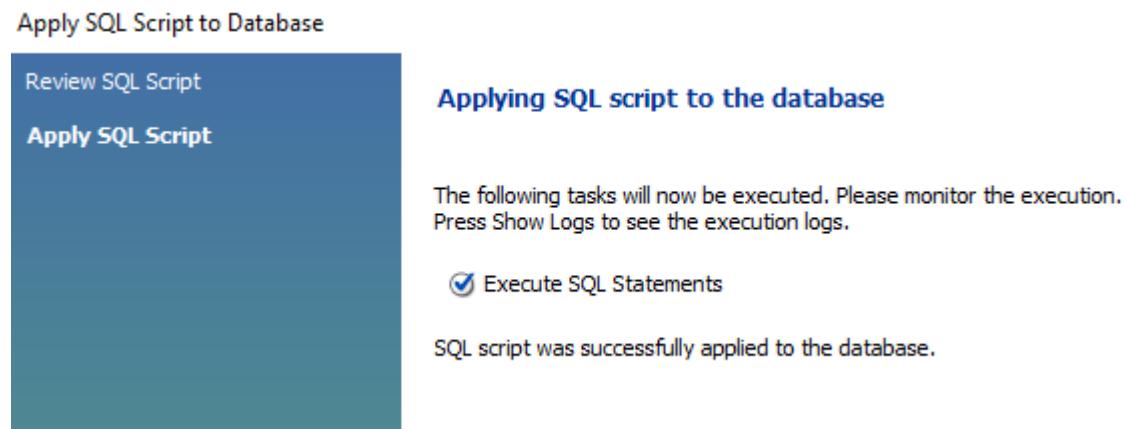


Figure 110: MySQL: Confirming applied changes #2

To test that the user role has been changed, you must log out of the website and log in again for changes to take effect. Once logged in again, the user's permission will now be set to admin.

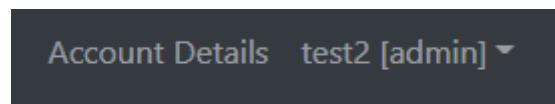


Figure 111: Updated role for user

6.8 Account Details

The user will also be able to change their account details at any time. This is done by clicking on the account details page on the navigation bar. Here they will see the following UI which will allow them to re-enter any of their details, they will need to re-enter their password to save changes.

Account Details

Username :	test2
First Name :	Test
Last Name :	Two
Email :	test2@hotmail.com
Phone Number :	435345435534

Change Password

New Password	Enter new password
Current password to save changes :	Current password
<input type="button" value="Save Changes"/>	
<input type="button" value="Delete Account"/> <small>Warning: Once you deleted you won't able to revert!</small>	

Figure 112: Account details UI

6.8.1 Entering password incorrectly

The following error message dialogue message will be shown if the confirming password is incorrect. The user will be prompted to try again.

Account Details									
Username :	test2								
First Name :	Test								
Last Name :	Two								
Email :	test2@hotmail.com								
Phone Number :									
 <h3>Change Password</h3> <table border="1"><tr><td>New Password</td><td>Enter new password</td></tr><tr><td>Current password to save changes :</td><td>Current password</td></tr><tr><td colspan="2"><input type="button" value="Save Changes"/></td></tr><tr><td colspan="2"><input type="button" value="Delete Account"/> <small>Warning: Once you deleted you won't able to revert!</small></td></tr></table>		New Password	Enter new password	Current password to save changes :	Current password	<input type="button" value="Save Changes"/>		<input type="button" value="Delete Account"/> <small>Warning: Once you deleted you won't able to revert!</small>	
New Password	Enter new password								
Current password to save changes :	Current password								
<input type="button" value="Save Changes"/>									
<input type="button" value="Delete Account"/> <small>Warning: Once you deleted you won't able to revert!</small>									


Something Went Wrong!
Entered password Incorrect

Figure 113: Account details Password validation

6.8.2 Deleting Account

Within this test, I will create a new account named 'Test3.' Here are all of the account details displayed.

Account Details

Username :	test3
First Name :	user
Last Name :	3
Email :	test3@hotmail.com
Phone Number :	32334234234

Change Password

New Password
Current password to save changes :	<input type="text" value="Current password"/>
	<input type="button" value="Save Changes"/>

Delete Account
Warning: Once you deleted
you won't be able to revert!

Figure 114: Deleting account

Result Grid						Filter Rows:	Edit:	Export/Import:	Result Grid
	id	username	password	firstName	secondName				Form Editor
	1	Test1	\$2a\$10\$xmpAE1gspW4nl5pXZK8DODad1411lp...	Test	One				
	2	test2	\$2a\$10\$8IIL4QoKyf.xQSzetO1OPOLWjiPrt7Flk...	Test	Two				
	3	test3	\$2a\$10\$3qKQOQ4KWXnlO0xPFKj6Wu1ifd1qh8...	test	3				
	NULL	NULL	NULL	NULL	NULL				

Figure 115: MySQL: Deleting account #1

Here is the view of the newly created user within the database.

To delete the user, the 'delete account' button must be clicked. This will display the following message for user's to double check their decision.

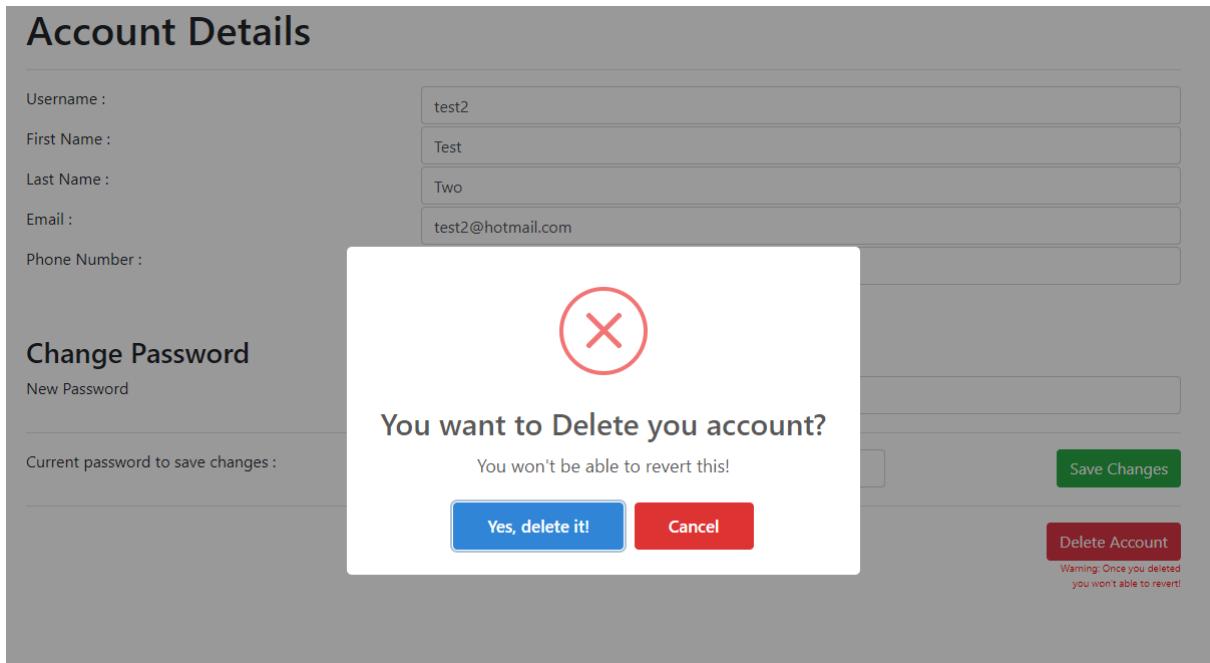


Figure 116: Deleting account confirmation dialogue message

Once deleted, the user will be returned to the login page and the 'Test3' account is therefore deleted in the database.

The screenshot shows a MySQL Workbench interface with a "Result Grid" window. The grid displays the following data:

	id	username	password	firstName	secondName
	1	Test1	\$2a\$10\$xmpAE1gspW4nl5spXZK8DODad141lp...	Test	One
▶*	2	test2	\$2a\$10\$8IIL4QoKyf.xQSzetO1OPOLWjiPrt7FlK...	Test	Two

Figure 117: MySQL: Deleting account #2

Chapter 7: Evaluation of Results

7.1 Checklist of requirements

Here, I will be detailing all the functions which were expected from my software. This is a summary of what was completed, including features which could have been completed and what was necessary.

Table 28: List of requirements for application

Function	Must include	Could not	Did not meet	Met
User creating an account	○			✓
User logging with account	○			✓
Changing account details: -Username -Password -Email Address -First Name -Last name -Phone Number	○ ○ ○ ○ ○ ○	○ ○ ○		✓
Ability for user to logout	○			✓
Permanently delete accounts		○		✓
Join and leave discussions		○		✓
Have a private chat with another user		○		✓
Video call with 2 users		○		✓
Creation of tasks	○			✓
Assigning tasks	○			✓
Completing a task	○			✓
Marking as not complete for a task		○		✓
Ability to assign different roles	○			✓
Add files to discussion	○			✓
Add files to private message		○		✓
Discussion List	○			✓
Task List	○			✓
Functional navigation bar	○			✓

7.2 Comparison of results to Case Studies

To fully understand the results of the overall project, it is essential to compare the results to the previous case studies researched within Chapter 2. I will be fully reviewing the positives and negatives of the 3 Scrum management tools previously stated.

Here is a brief overview of all the functionalities within the scrum application, these core features, were the most desirable features listed from the early development stages. It shows what has been completed in the final product, and how it competes to other on market software.

Each software will be compared, giving positives and negatives of each, then compared with my final project outcome.

Table 29: Scrum applications desired functionalities

Software	Private Messaging	Boards	Video Calls	Audio Calls	File/ Content Sharing	Role/Task Assignment
My project	✓	✓	✓	✓	✓	Both

7.2.1 QuickScrum

QuickScrum was the very first software which was researched. Some of the best features include:

- **Tracking of activities**
- **Tracking to view individual's progress**
- **Drag and drop implemented into the application**
- **Charts**
- **Third party integrations**

Some of the worst features include:

- **Takes a long time to load some pages**
- **Overall organisation of the page**
- **Customer support for the site is unreliable**
- **Does not include private messaging or video calling**

Overall, I feel like I was able to complete all the main features from QuickScrum, I was able to implement video and audio calling unlike QuickScrum. This allows for a higher quality of communication between team members. I feel like QuickScrum could be optimised a lot better, as the pages take a while to load. With my project, it has quick loading times, only downside is running the application, as it takes a few more seconds to deploy the website. I feel that QuickScrum's user interface is a lot more professional than mine, however it lacks the key organisation of tasks and discussions, which my project has successfully implemented.

To conclude, my project managed to exceed all functions that are currently available within QuickScrum, this can be viewed from the comparison table below.

Table 30: Features included within QuickScrum

Software	Private Messaging	Boards	Video Calls	Audio Calls	File/ Content Sharing	Role/Task Assignment
My project	✓	✓	✓	✓	✓	Both
QuickScrum		✓			✓	Both

7.2.2 nTask

The second software which was researched was nTask. I will list the main features which are desirable:

- **Tasks**
- **Meetings**
- **Timesheets**
- **Risks and issues within the workspace**
- **Third-party integrations**
- **Issue tracking**
- **Project portfolio**
- **Daily scrum management**
- **Team Collaboration**

Some of the worst features include:

- **Some key customisation and formatting options not included**
- **Updates take too long and disengage customers**
- **Cannot easily attach files**
- **Not user friendly**

Overall, I feel that nTask is an advanced scrum management tool, as it does not offer consistent user-friendly functions throughout the site. It also does not provide private messaging or video calling, which is something I wanted to change within my project. Due to the complexity of the site, it could draw away from Scrum teams to use it, therefore I kept my software simple and easily navigable. To conclude, I feel as though my software exceeds nTasks features by providing more of a communication platform, not primarily just task assignment. The table below compares the project alongside nTask's features.

Table 31: Features included within nTask

Software	Private Messaging	Boards	Video Calls	Audio Calls	File/ Content Sharing	Role/Task Assignment
My project	✓	✓	✓	✓	✓	Both
nTask		✓			✓	Both

7.2.3 VivifyScrum

Another scrum management tool that was researched within Chapter 2, was VivifyScrum. This software has a sleek and clean UI, keeping all information organised, increasing the workflow for teams. This is also useful for teams transferring over to scrum principles.

The best features of VivifyScrum include:

- **A project calendar with tasks and important dates integrated**
- **Time tracking of the project**
- **Scrum board built in**
- **Roles and permissions**
- **Use of charts**
- **The use of third-party integrations**

The worst features of VivifyScrum include:

- **Search feature is unreliable**
- **Tasks completed are hard to filter through**
- **Not as fully developed as previous scrum management tools**

Compared to my project, VivifyScrum fails to implement the video calling and messaging. Also, VivifyScrum seems a lot less polished compared to nTask and QuickScrum, with problems integrating features like GitLab. However, it includes charts and the ability to create them, something my project lacks. There is also a calendar implemented within VivifyScrum, another feature that my project does not include. Overall, I feel like the developers at VivifyScrum made a lot more experimental developments compared to my own project and other case studies. However, it lacks fully testing and updating these functionalities.

Another comparison table is included for the features of my project and VivifyScrum.

Table 32: Features included in VivifyScrum

Software	Private Messaging	Boards	Video Calls	Audio Calls	File/ Content Sharing	Role/Task Assignment
My project	✓	✓	✓	✓	✓	Both
VivifyScrum		✓			✓	Both

To conclude, the project has fully met all features available on the current market's software, while implementing more communication-based features, like private messaging and video calling. The project is a mix of both features from within communication web applications and scrum management tools.

7.3 Use Case Tools

Within this section, it will outline all the tools which aided the completion of the project.

Table 33: Use case tools

Case Tool	How it was used	Positives
Microsoft Visio	Visio was used to create ERD diagrams.	Lots of features and templates are available to compare ERD diagrams.
Lucid Charts	Lucid charts were used to create flow charts and the use case diagram.	Lots of symbols and templates are included to aid in the designing of diagrams.
Microsoft Access	Access was used to create the data model diagram for the database.	Allowed to fully edit the data model and design the initial database design and planning.
Microsoft Word	All the report was structured and written on here.	Easy to format and organise all the different sections within the report, ability to add figure tables and easily combine all the references with Harvard referencing style.
Notepad++	All the SQL was copied and pasted on here, to submit as an SQL file.	Easy to organise SQL code, easier to read through compared to notepad, as notepad++ can format code in different programming languages.
MySQL workbench 8.0 CE	The database is deployable on MySQL, also used within the coding on IntelliJ.	Easy to implement within IntelliJ, also easier to edit tables and test data.
Google Chrome	All of research was carried out within google chrome.	Made overall research concise and efficient. I also used chrome to view the applications webpages.
Paint	Paint was used to edit and format screenshots of software design.	Produced good quality diagrams and easy to use.
Snipping tool	Used to crop relevant screenshots.	Easily and quickly snippet information diagrams needed for the report, useful for testing the application.
IntelliJ IDEA community edition	Used to code the frontend and backend parts of the software. All design was also done within.	Contains easy database integration, and well-structured and organisation folders containing all the project within.
NetBeans IDE 8.2	The original design ideas were completed within a Java JFrame.	Allowed for quick templates of the webpages design to be completed, easy to use with many features within JFrame.

Chapter 8: Conclusions

8.1 A Summary of what has been achieved within the project

To conclude my final project report, I feel as though I successfully carried out the projects goal: Work Capacity Management System to Facilitate Self Organisation of a Scrum.

Here is a summary of all completed objectives hit after the successful completion of the project:

- A fully deployed web application with backend database

This is significant as it allows for all user details and interactions to be stored, data is never lost throughout the user's interactions. This is important as many users will be interacting with the site, showing the importance of the negative impacts on data loss.

- A login and register page which allows users to access the application, user authentication

This is significant as it allows for every user to have their own access to the site, eliminating the risk of personal data being stolen, or other team members to view information intended for scrum masters only.

- Creation of tasks which users can assign and mark as complete

This is significant as each user has their own responsibility, this is viewable to all team members and ensures that everyone is up to date with the progression of their own team's project.

- Table of tasks available to user, can switch between completed and not completed

This is significant as it allows for more efficient workload, users can keep up to date with their tasks, and update them in real time to ensure that their scrum team leader acknowledges each individual team members contributions.

- Ability to private message other users registered to the web application

This is significant as certain information may only need certain individuals to access, private files can be sent to another team member, to keep privacy and stop third parties from viewing conversations or files.

- Ability to create and join and leave discussions and message within them, ability to send attachments

This is important, as it allows users to manage their discussions properly, if a discussion is no longer important, they are free to leave, decluttering their feed.

- List of discussions all users are a part of

This allows for users to ensure that they can view which discussions they are a part of in case they are missing crucial discussions to involve them in team discussions.

- Attachments able to be sent within private messages and within discussions

This is importance as attachments of crucial documents may need to be circulated around for all team members to view and develop on.

- Logout feature

Allows for better testing, also ensuring maximum security. This is because the user might access the software from a public PC, this reduces the chances of unauthorised parties attaining valuable organisation data.

- Ability to change account details and permanently delete accounts

Sometimes, users may need to update their email address or contact number, therefore account details are crucial to be changed. This also allows for scrum team leaders an insight on how to contact the team member if they cannot reach them from within the software.

- Ability to set roles consisting of admin and regular user

This was crucial as Scrum team leaders would need an admin role to fully control the regular users' actions, be able to delete discussions or any users which may not be involved in the software any longer.

- Allow for 2 users' to privately video call one another

This was crucial as it allows for a more 1 to 1 private environment. Video calling is also more useful for better collaboration, as messaging does not allow for tone and quicker collaboration in certain scenarios.

The program was completed to the highest standard, 100% of functionalities are completed and working. All functionalities have been tested vigorously, in preparation to the viva. This ensures that the application is functional, reducing the chance of encountering errors within the code. The code is fully polished up and efficient, to ensure it is easily readable and error free. Finally, the report outlines and covers the entirety of thought processes and decisions made, backed up with evidence. All improvements to the program have been documented, as well as compared to previous ideas.

8.2 Reflections of lessons learned

Completing all my research within Scrum's and the ways teams co-operate daily has given me much clearer insight on what I needed to implement within the website.

It has given me the ability to dissect and evaluate all the current market solutions available, making me question all the chosen features within.

Designing the database and interface has made me more aware of certain rules which apply to successful applications, this has also provided deeper understanding from concepts I have learnt in the previous years at the University. I feel as though this project has helped me gain many new and refined skills needed to become a successful developer within a competitive market.

Within my project's progression, overcame a lot of obstacles to achieve the highest quality report and application within the time allocated. One of the most important lessons I learnt was to always stay on top of all progression which I had made, reflecting on application progress, and ensuring that all possible improvements were completed and refined before submission deadlines.

Another lesson I learnt was not to delay any progress relating to my application, as towards deadlines it made it difficult to complete all tasks as well as other module assignments, therefore it was vital for me to complete all progress on the application before submissions, this enabled me to have more time for revisions and able to refine work to a higher quality standard.

8.3 Legal, Social, Ethical and Professional Issues

Within this section, legal issues regarding current team management systems on the market will be discussed.

8.3.1 Data Protection Regarding Project

To use the application, users will have to enter and provide personal information which will be stored within the database. If this website were ever put on the market, consideration would have to take place regarding the current laws associated with Data Protection. For example, Microsoft teams have plenty of policies and security implemented, like Data Loss Prevention (DLP) and Information Barriers. *“These policies can impact users in 1:1 chat, group chats, or at a team-level.”* (Microsoft, 2020) This protects users from their information being accessed by unwanted users and used for malicious activities.

If this website were to be deployed onto the market, the following features would need to be added. First, providing the user the option to delete all their data from the website, to ensure it was complying with the UK law, the General Data Protection Regulation. Secondly, to ask the user before a successful registering of their account if they want their data to be used for further research and analytics, giving the user a choice. Finally, when registering, the user would be provided information on how their data will be stored and used in the site, giving them a clear understanding and chance to back out if they did not consent.

Another issue that could arise is that the company may log the user’s activities, this is due to all messages and files being saved for all team members to see. If the application went on the market, some companies may want to have access to their employee’s private messages on the application. This is good, since if someone were being bullied or harassed, the company would have a record of the messages and keep it as evidence against the individual being disruptive. This would then allow the company to take the suitable course of action to prevent it from happening again, the company could also give suggestions on how to improve the application to deal with this kind of behaviour. First, a report button function would be implemented, to allow anyone receiving abuse to report the user, this could be to the company or to the developer of the site directly. This report would be reviewed, and the appropriate action would be carried out to eliminate the negative behaviour.

Another issue is that the company may be able to watch each employee’s screen. This breaches their own personal space. To ensure this would not happen on the platform, addition of terms and rules would be included on the navigation page, to tell employers their rights on the platform.

8.4 Future Work

Within this section, I will be outlining some of the future work I will be looking into, considering all the skills improved during the entirety of this module.

After the completion of the project, I will be preparing myself for the viva, and making sure that it completed to the best of my ability. I will be listing the short-term goals and long-term goals which were covered and planned for after the modules completion.

Short Term Goals

Within the project, it has aided in helping construct a path for future goals that might be achieved after the successful completion of this module.

First, this project has helped develop many more coding skills, especially relating in Java. Throughout my university journey, I learnt a lot about basic Java coding and how to implement within simple programs. Due to this project, it has allowed me to complete extensive research into complex functionalities, like implementing a video calling system, to designing a more professional user interface using Java React. As a short-term goal, I would like to research more into the available functionalities within Java, possibly expanding my coding knowledge into a new programming language like C#.

Another short-term goal is to achieve the highest possible marks for this module, this project has been a key motivator in achieving a first-class degree, this is due to fully exploring what it may be like to fully develop a program as a full stack developer. This will aid into job searching, which is a long-term goal of mine. This module has also been my favourite throughout final year, it allowed me to fully complete my own research and design a program with complete freedom.

Another short-term goal within this project was to eliminate my procrastination tendencies. I felt like with the constant deadlines and submissions, this gave me motivation to try my hardest to achieve the highest marks I could possibly attain. The Gantt chart helped greatly as it gave me a clear outline of all features and tasks needed to complete the project. I never once felt stranded within the project as I always had support from my classmates and supervisor, which I cannot thank enough.

Long Term Goals

Throughout this project, a long-term goal was to organise myself as a person. This aided in giving me better efficiency within other deadlines I had to complete, I used the methodologies within other module coursework's to allow myself to stay on track with the progress. It gave me important life management skills, to attain control in certain situations. This helped in both academic life and real work scenarios involving work.

Another long-term goal was to achieve a better job at the end of my degree. Due to the nature of this project, I feel as though it was given me the relevant skills to complete a master's degree, which

is a possibility I am currently looking into. I will also be following my interest in full stack development due to the success of the project, to enable me to find a job suitable to my acquired skills.

Appendices

1.1 Original Project Plan

This is my original plan from the project proposal:

No	Activities	Estimate Duration	Activity Description
1	Analysing sources and technical reports	4 weeks	Gathering resources and current management to improve my project.
2	Source review	4 weeks	Reviewing all the reports and books and choosing what should be in my final report.
3	Researching scrum software	4 weeks	Researching all the current software's scrum teams are using to evaluate the positives and negatives, to also evaluate what needs changing.
3	Interviewing scrum teams + scrum masters	3 weeks	Emailing teams and individuals who have worked in scrum in the past, asking questions about their experiences using scrum.
5	Concept models	3 weeks	Designing prototypes of possible designs of UI's to be used for the management system, comparing to current management systems used by scrum teams
6	Finalising design choice	3 weeks	Checking my design to see if it is suitable for my project requirements, comparing to the other on market management systems.
7	Reviewing UI designs	4 weeks	Reviewing all my current prototype designs and evaluating their strengths and weaknesses.
8	Implementing code with MySQL+ Eclipse	16 weeks	Writing up the code, implementing the different features and the visual content aspects.
9	Code clean-up	4 weeks	Adding comments, condensing the code, looking for any improvements, consulting with supervisor.
10	Prototype testing	4 weeks	Checking for errors, touching up, making sure it is fully functioning.
11	Final testing and adjustments (Final submission)	2 weeks	Finalising prototype and report, spell-checking, formatting the report to improve presentation.

Figure 118: Original Project Plan

1.2 Review of Project Progress

After completing the application, the overall project's aims and objectives were successfully met. All work completed was carried out critically, making sure to implement all features agreed with the supervisor from the original proposal.

After reviewing the original project plan, all the work was carried out efficiently and properly, this is due to ensuring the plan was the backbone of all decisions made. Improvements within the project were also compared to this plan, to ensure that it followed the core principles.

Amending the Gantt chart within the interim report helped reflect the actual carried out, it gave an insight on the realistic work effort needed to achieve to complete the highest quality of work within the minimum amount of time given. Within the original Gantt chart, it did not provide clear and concise information within each task. By updating it, this allowed for more insight for both completing the project and for supervisors to have a better insight on the project's intentions and aims.

All research has been successfully completed within software engineering and agile methodologies, giving examples. This is to ensure that all options have been fully explored, that the right methodology was used to allow for the most efficient competition.

The UI and database have been successfully implemented, this was due to referring and researching the current markets solutions. This is evidently backed up by chapter 3's requirement analysis, the UML diagram and all the high-level use cases and descriptions.

Throughout the implementation of the software, a change was made to the IDE, changing from Eclipse to IntelliJ. This was altered due to IntelliJ providing more relevant functionality needed for this software. It was easier to implement, due easier database integration, and the use for more plugins.

In conclusion, the overall software was implemented to the highest quality, this was due to full exploration of all options relating to functionality, and the IDE's used within other similar software. It combines important functionalities needed for Scrum Management teams to successfully utilise, eliminating the need for additional software. This software fully exceeds all expectations set out in the proposal, by adding additional features and fully exploring all requirements.

A table will be shown, listing all the report's chapters and vital information, outlining sections that went well, including troubles faced and how they were overcome.

Table 34: Review of report structure

Specific stage	Positives	Negatives
Introduction	<ul style="list-style-type: none"> From the beginning, the main objectives and purpose of application were identified. The report properly identifies the initial problem, which was the sole purpose of completing the project. 	<ul style="list-style-type: none"> Completing insightful research into which methodologies should be used to complete the project. This was completed by asking the supervisor for their opinion early on within the project's lifecycle.
Background Research	<ul style="list-style-type: none"> Lots of time and effort went into analysing all current on-market solutions for Scrum software. Comparisons were made between the 4 case studies. 	<ul style="list-style-type: none"> This section took the longest to complete, leaving little time to polish up other sections. This was overcome by using references which were up to date and from a reliable source, filtering out irrelevant information.
Requirements Analysis	<ul style="list-style-type: none"> The program is fully explained, giving insightful information on how to run and install the program. Also lists all the folders used and explains in detail. 	<ul style="list-style-type: none"> It was time consuming including all screenshots and explaining each step. This was overcome by giving additional time to fully complete this chapter, ready for the final report.
Design	<ul style="list-style-type: none"> The software has an overall professional feel to it, by using Java React. The original designs from NetBeans were reflected upon throughout the entirety of designing process. 	<ul style="list-style-type: none"> It was difficult to implement a new JavaScript library, therefore extensive research was completed to fully explore all possible implementations within the software.
Implementation	<ul style="list-style-type: none"> The software fully exceeds all expectations of what was originally required. Additional functionality including changing account details were implemented. 	<ul style="list-style-type: none"> This was challenging to complete, to overcome this, research was completed to implement the calling feature. YouTube videos and various websites were used to grasp the core ideas and code needed to complete.
Evaluation of Results	<ul style="list-style-type: none"> The evaluation was well documented, referring to the original case studies and concluding the similarities and differences. 	<ul style="list-style-type: none"> Identifying what was learnt throughout the project was challenging, this was overcome by reviewing each section of the software and documenting in concise sections.
Conclusions	<ul style="list-style-type: none"> Overall, the conclusion ties all work completed together and illustrates positives and negatives faced. 	<ul style="list-style-type: none"> Like the evaluation, it was hard to cover everything, this was overcome by reviewing all report sections and code.

1.3 Amendments to the Original Plan of Project

This section will outline the current completion of the project. To conclude, the project has exceeded original expectations.

I will be reviewing the completeness of my project plan that was created within the interim report.

Table 35: Amendments to Original WBS plan

Stage	Estimated duration	Actual Duration	Difference	Completion
Analysing sources and technical reports	4 weeks	4 weeks	0 weeks	Complete
Source Review	4 weeks	3 weeks	+1 week	Complete
Researching scrum software	4 weeks	4 weeks	0 weeks	Complete
Interviewing scrum teams and scrum masters	3 weeks	2 weeks	+1 week	Complete
Concept Models	3 weeks	2 weeks	+1 week	Complete
Finalising design choice	3 weeks	1 week	+2 week	Complete
Reviewing UI designs	4 weeks	5 weeks	-1 week	Complete
Implementing code	16 weeks	17 weeks	-1 week	Complete
Code Clean up	4 weeks	4 weeks	0 weeks	Complete
Prototype Testing	4 weeks	5 weeks	-1 week	Complete
Final Testing and Adjustments	2 weeks	3 weeks	-1 week	Complete

It shows that the source reviewing and interviewing took a week less to complete. This was due to increased motivation and workflow. These sections were crucial to the project as they aided in designing the application suitable to scrum needs.

Concept models took a week less to complete; this was due to having a basic plan in mind before the project was started. These concept plans originally took place in the summer before third year, to ensure there was a solid idea in place ready to implement.

The most challenging parts of the project were the implementation and testing. This was due to the extensive number of features implemented, it took time and practice to refine all functionalities. Also, due to the COVID pandemic, it was hard to find individuals to test the program, to overcome this I allowed trusted individuals to remotely connect to the PC, saving time reinstalling and configuring the application within the testing stages.

Final adjustments and testing took a week longer due to a backup of implementation work let to complete, this was overcome by dedicating days to polish up the report and code. Overall, the project plan was carried out successfully and efficiently, a few stages were completed rather slowly compared to others, however this was successfully completed due to dedicated time assigned to complete any left-over tasks.

1.4 Lessons Learnt in Project Management

Within this section, the lessons learnt from the overall completion of the project will be listed. These lessons are regarding the project management completed, how the overall project was carried out over the given timeline.

First, the most important lesson was to ensure that the projects aim, and objectives were constantly motivated. This was complete by consistently relating the project back to the original planning and aims documented within the proposal. Due to the length of the project, the original principles could often be overlooked, reviewing all work completed ensured that no points were forgotten about within finalising of the report. Leading up to final deadlines, it was often felt like not much work was being completed, by keeping motivation, this aided a lot in those crucial days.

Another lesson learnt was never to give up on an idea, rather to keep trying. This was important as most ideas led into more refined versions of the software. It is also key to work within a quiet and relaxed environment, this was to reduce stress and procrastination. It also led for an increased workflow as distractions were not present. Trying relates to motivation, as motivation was the key driving point in ensuring to never give up with any difficulties faced.

This ties into another important lesson, keeping on schedule. This was easier due to completing milestones and Gantt charts within the proposal stage, it always was used for referring to the key outlines of the structure, also ensuring that focus and concentration were present throughout the project's lifecycle.

Another crucial lesson learnt was to never skip a task. Organisation is key within completing a project, so therefore leaving steps out for future work always led to failure. This was due to certain features being overlooked and ultimately leading to confusion. This concludes this importance of having a structured and organised workflow.

Finally, the last important lesson was to ensure that references were cited. Most sources which were come across were unreliable or outdated, to ensure that all information used was relevant, it was crucial to use websites which were well known and up to date. Therefore, multiple references were looked at and evaluated before incorporating into the final report.

1.5 Survey Responses

Within this section, I will be asking family members and classmates to review my applications features, giving a rating 1-10. 1 is for not very good, and 10 is for exceptional work. They will also be asked on what they like about each component, and any feedback on improvements which could be added after submission.

The first person I asked to complete a survey, was my boyfriend Callum. Here are his responses listed below.

Callum's response

Question 1: How well did you understand what the website was about? (1-10)

10

Question 2: Why was that number scored? (Give reasons)

The website has clear purpose and has all relevant information in the correct places.

Question 3: Did you find that the navigation bar was useful? (1-10, Give reasons)

9

Question 4: Is the discussions page easy to interact with? (Yes/No, Give reasons)

Yes, layout is completed well, listing is easy to interact with, within discussions lists all users within the current discussion and has good structure.

Question 5: Is the task assignation easy to complete? (Yes/No, give reasons)

Yes, it is all within discussions webpage so therefore is easier to assign a task to the current user.

Question 6: What are your thoughts on overall user interface?

It is good, could use more pictures and maybe more use of fonts.

Question 7: Would you recommend this web application to someone interested in creating a Scrum team collaboration? (Yes/No)

Yes

Question 8: What improvements would you suggest making? (Give suggestions)

More user interface designing, maybe improve the call feature in future to add a microphone mute option.

The second person I asked to complete a survey, was my classmate Kevin.

Kevin's response

Question 1: How well did you understand what the website was about? (1-10)

9

Question 2: Why was that number scored? (Give reasons)

Because the website has a simple layout which is easy to use, which make it very easy to understand what purpose it is built for. It also provides a user for people within a team to communicate.

Question 3: Did you find that the navigation bar was useful? (1-10, Give reasons)

9

Question 4: Is the discussions page easy to interact with? (Yes/No, Give reasons)

Yes, as it is clearly layered out where all the functions, also how to use it such as how to assign a task and how to open an attachment.

Question 5: Is the task assignation easy to complete? (Yes/No, give reasons)

Yes, as it is clearly shown how to add a user to a task, also creating one is simple as it lists all tasks which have been made.

Question 6: What are your thoughts on overall user interface?

Could use more images and more information to fill the site up. I like the layout and structure of all webpages.

Question 7: Would you recommend this web application to someone interested in creating a Scrum team collaboration? (Yes/No)

Yes

Question 8: What improvements would you suggest making? (Give suggestions)

Add more information on each page to clearly show a user how to interact with the page.

The third person I asked to complete a survey, was another one of my classmates, Emmanuel.

Emmanuel's response

Question 1: How well did you understand what the website was about? (1-10)

9

Question 2: Why was that number scored? (Give reasons)

It is having a good layout; I like the navigation bar and the ability to change user details

Question 3: Did you find that the navigation bar was useful? (1-10, Give reasons)

10

Question 4: Is the discussions page easy to interact with? (Yes/No, Give reasons)

Yes, it allows for better organisation, I like the use of the different buttons and the ability to join a discussion and leave it. Also, how it updates in real time.

Question 5: Is the task assignation easy to complete? (Yes/No, give reasons)

I like how it was combined into the same page as discussions, also how you can give each task a description.

Question 6: What are your thoughts on overall user interface?

The user interface is great, I like the use of the space within the private messaging and discussion pages.

Question 7: Would you recommend this web application to someone interested in creating a Scrum team collaboration? (Yes/No)

Yes

Question 8: What improvements would you suggest making? (Give suggestions)

More features to be added on the video calling, maybe a feature to include group video calling to allow for better communication.

The final person I asked to complete a survey was my classmate, Beaulah.

Beaulah's response

Question 1: How well did you understand what the website was about? (1-10)

10

Question 2: Why was that number scored? (Give reasons)

Everything is structured in a good format; it provides details and a lot of useful functions for a scrum team.

Question 3: Did you find that the navigation bar was useful? (1-10, Give reasons)

8

Question 4: Is the discussions page easy to interact with? (Yes/No, Give reasons)

Yes, I feel like it would help a user be able to efficiently look for specific chat logs.

Question 5: Is the task assignation easy to complete? (Yes/No, give reasons)

Yes, as it only takes a few clicks to assign a task

Question 6: What are your thoughts on overall user interface?

The space is used well, the login page and register page are well designed and I also like the listing of tasks with the completed/not completed buttons.

Question 7: Would you recommend this web application to someone interested in creating a Scrum team collaboration? (Yes/No)

Yes

Question 8: What improvements would you suggest making? (Give suggestions)

N/A

I feel as though the surveys went well, it offered me a lot of feedback and room to improve on future projects. Also, it gave crucial practice of showcasing my application, ready for my viva.

1.5 New Project Gantt Chart

This is my second version of the Gantt chart, purple shows tasks, yellow shows report writing, and the diamond shape represents milestones.

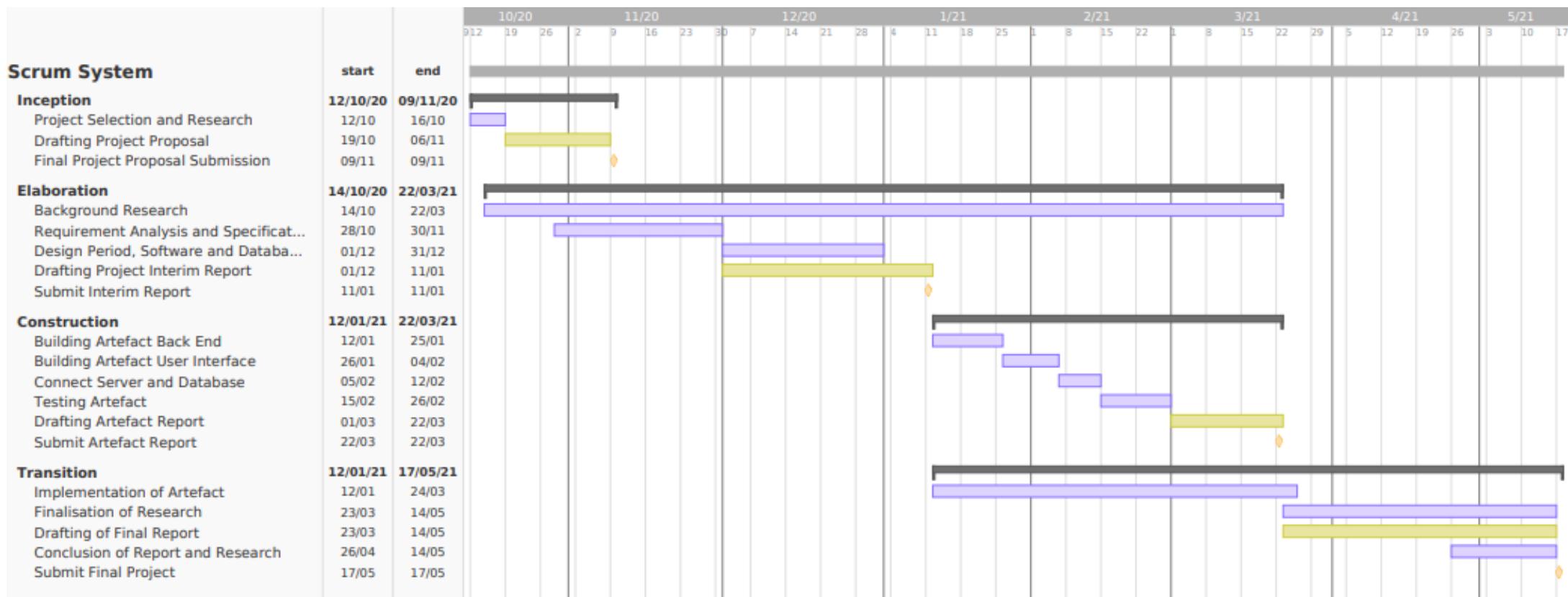


Figure 119: Updated and Amended Gantt Chart

References

- Beck, K. (2012). *Extreme Programming Explained*. John Wiley.
- Bellis, M. (2019, February 27). *The History of HTML and How It Revolutionized the Internet*. Retrieved from ThoughtCo.: <https://www.thoughtco.com/history-of-html-1991418>
- Botello, L. (2021). *Eclipse: A Brief History*. Retrieved from IRI:
<https://www.iri.com/blog/iri/business/brief-history-of-eclipse/#:~:text=Eclipse%20was%20originally%20started%20by,platform%20for%20all%20IBM%20products.&text=In%202001%2C%20to%20facilitate%20developer,source%2C%20royalty%2Dfree%20license.>
- Chen, J. (2020, August 5). *Important Instagram stats you need to know for 2020*. Retrieved from sproutsocial: <https://sproutsocial.com/insights/instagram-stats/>
- Faraz, M. (2020, March 29). *The Success Story of Instagram*. Retrieved from medium: <https://medium.com/@iMohammadFaraz/the-success-story-of-instagram-5635b2e9c7ff>
- Fowler, M. (2013, July 13). *Extreme Programming*. Retrieved from martinfowler:
[https://martinfowler.com/bliki/ExtremeProgramming.html#:~:text=Extreme%20Programming%20\(XP\)%20is%20a,dominant%20as%20the%20noughties%20passed.](https://martinfowler.com/bliki/ExtremeProgramming.html#:~:text=Extreme%20Programming%20(XP)%20is%20a,dominant%20as%20the%20noughties%20passed.)
- Hafidz, M. U., & Sensuse, D. I. (2019). *The Effect of Knowledge Management System*. IEEE.
- Highsmith, J. (2013). *Adaptive Software Development: A collaborative approach to managing complex systems*. Addison-Wesley.
- Hoffmann, J. (2017, October 22). *A Look Back at the History of CSS*. Retrieved from CSS-Tricks: <https://css-tricks.com/look-back-history-css/>
- ISO. (n.d.). *MANAGEMENT SYSTEM STANDARDS*. Retrieved from ISO: <https://www.iso.org/management-system-standards.html>
- Javatpoint. (2018). *History of Java*. Retrieved from Javatpoint: [https://www.javatpoint.com/history-of-java#:~:text=Java%20was%20developed%20by%20James,project%20in%20the%20early%20'90s.&text=1\)%20James%20Gosling%2C%20Mike%20Sheridan,sun%20engineers%20called%20Green%20Team.](https://www.javatpoint.com/history-of-java#:~:text=Java%20was%20developed%20by%20James,project%20in%20the%20early%20'90s.&text=1)%20James%20Gosling%2C%20Mike%20Sheridan,sun%20engineers%20called%20Green%20Team.)
- JetBrains. (2021). *IntelliJ IDEA*. Retrieved from JetBrains: <https://www.jetbrains.com/idea/>
- Microsoft. (2020, 11 12). *Security and compliance in Microsoft Teams*. Retrieved from Microsoft: <https://docs.microsoft.com/en-us/microsoftteams/security-compliance-overview>
- Mircosoft. (2021). *Microsoft Teams*. Retrieved from Microsoft: <https://www.microsoft.com/en-gb/microsoft-teams/group-chat-software>
- MySQL. (2021). *MySQL*. Retrieved from MySQL: <https://www.mysql.com/>

- NetBeans. (2021). *A brief history of NetBeans*. Retrieved from NetBeans:
<https://netbeans.org/about/history.html#:~:text=Jaroslav%20Tulach%2C%20who%20designed%20the,the%20way%20to%20deliver%20them>.
- Oracle. (2017). *Introduction to Java EE*. Retrieved from avaae.github.io:
<https://javaee.github.io/tutorial/overview001.html#:~:text=The%20aim%20of%20the%20Java,responsible%20for%20all%20Java%20technologies>.
- Oracle. (2021). *Java EE FAQ*. Retrieved from Oracle:
<https://www.oracle.com/java/technologies/javaee-faq.html>
- Oracle. (2021). *Oracle Java*. Retrieved from Oracle: <https://www.oracle.com/java/>
- Oracle. (2021). *Oracle JDBC FAQ*. Retrieved from Oracle :
<https://www.oracle.com/uk/database/technologies/faq-jdbc.html>
- Pressman, R. S. (2019). *Software Engineering A practitioner's approach*. McGraw-Hil.
- ProductPlan. (2021). *What is Adaptive Software Development (ASD)*? Retrieved from ProductPlan:
[https://www.productplan.com/glossary/adaptive-software-development/#:~:text=Project%20managers%20John%20Highsmith%20and,Development%20\(RAD\)%20agile%20framework](https://www.productplan.com/glossary/adaptive-software-development/#:~:text=Project%20managers%20John%20Highsmith%20and,Development%20(RAD)%20agile%20framework).
- React. (2021). *React A JavaScript library for building user interfaces*. Retrieved from Reactjs:
<https://reactjs.org/>
- Santos, L. S., L'Erario, A., Pagotto, T., Fabri, J. A., Oliveira, F. S., & Camilo, J. R. (2018). *A Scrum-based process to distributed projects in multidisciplinary teams*. IEEE.
- Saran, C. (2020, May 29). *The software innovation impact of coronavirus*. Retrieved from computerweekly: <https://www.computerweekly.com/news/252483854/The-software-innovation-impact-of-Coronavirus>
- Slack. (2021). *Slack makes it downright pleasant to work together*. Retrieved from Slack:
<https://slack.com/intl/en-gb/>
- Spring. (2021). *Spring makes Java*. Retrieved from Spring.io: <https://spring.io/>
- Sutherland, J., & Schwaber, K. (2020). *The Scrum Guide*.
- Syal, R. (2013, September 18). *Abandoned NHS IT system has cost £10bn so far*. Retrieved from The Guardian: <https://www.theguardian.com/society/2013/sep/18/nhs-records-system-10bn>
- Trello. (2021). *Trello helps teams move work forward*. Retrieved from Trello:
https://trello.com/?&aceid=&adposition=&adgroup=127256029088&campaign=12908495645&creative=518177140446&device=c&keyword=trello&matchtype=e&network=g&placement=&ds_kids=p63375551074&ds_e=GOOGLE&ds_eid=700000001557344&ds_e1=GOOGLE&gclid=CjwKCAjw-qeFBhAsEiw
- Tyson, M. (2019, April 11). *What is JDBC? Introduction to Java Database Connectivity*. Retrieved from InfoWorld: <https://www.infoworld.com/article/3388036/what-is-jdbc-introduction-to-java-database-connectivity.html>

VisualParadigm. (2021). *Requirement Analysis Techniques*. Retrieved from VisualParadigm :
<https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/#:~:text=Requirement%20Analysis%2C%20also%20known%20as,requirements%20gathering%20or%20requirements%20capturing>.

Zoom. (2019). *About Us* . Retrieved from Zoom:
https://explore.zoom.us/about?_ga=2.222333723.1445902788.1608057201-245763263.1608057201

Bibliography

- AC Communications Systems Corporation. (2000). *The Scrum software development process for small teams*. IEEE.
- Anderson, J. (2019, June 25). *How Does Technology Impact Politics?* Retrieved from Accquia: <https://www.acquia.com/blog/how-does-technology-impact-politics>
- Ann McIver McHoes, I. M. (2011). *Understanding Operating Systems Sixth Edition*. Boston: Course Technology.
- Beck, K. (2012). *Extreme Programming Explained*. John Wait.
- Bellis, M. (2019, February 27). *The History of HTML and How It Revolutionized the Internet*. Retrieved from ThoughtCo.: <https://www.thoughtco.com/history-of-html-1991418>
- Botello, L. (2021). *Eclipse: A Brief History*. Retrieved from IRI: <https://www.iri.com/blog/iri/business/brief-history-of-eclipse/#:~:text=Eclipse%20was%20originally%20started%20by,platform%20for%20all%20IBM%20products.&text=In%202001%2C%20to%20facilitate%20developer,source%2C%20royalty%2Dfree%20license.>
- Castells, M. (2013). *The Impact of the Internet on Society: A Global Perspective*. Retrieved from BBVA Open Mind: <https://www.bbvaopenmind.com/en/articles/the-impact-of-the-internet-on-society-a-global-perspective/>
- Chakrabarti, S. (2018, January 2018). *Hard Questions: What Effect Does Social Media Have on Democracy?* Retrieved from About Facebook: <https://about.fb.com/news/2018/01/effect-social-media-democracy/>
- Chen, J. (2020, August 5). *Important Instagram stats you need to know for 2020*. Retrieved from sproutsocial: <https://sproutsocial.com/insights/instagram-stats/>
- Clement, J. (2020, April 24). *Global digital population as of April 2020*. Retrieved from statista: <https://www.statista.com/statistics/617136/digital-population-worldwide/>
- Dmitriy. (2019, May 8). *Technology's impact on democracy: between positive results and added difficulties*. Retrieved from Chatham House: <https://demtech.chathamhouse.org/submission/technologys-impact-on-democracy-between-positive-results-and-added-difficulties/>
- Faraz, M. (2020, March 29). *The Success Story of Instagram*. Retrieved from medium: <https://medium.com/@iMohammadFaraz/the-success-story-of-instagram-5635b2e9c7ff>
- Fowler, M. (2013, July 13). *ExtremeProgramming*. Retrieved from martinfowler: [https://martinfowler.com/bliki/ExtremeProgramming.html#:~:text=Extreme%20Programming%20\(XP\)%20is%20a,dominant%20as%20the%20noughties%20passed.](https://martinfowler.com/bliki/ExtremeProgramming.html#:~:text=Extreme%20Programming%20(XP)%20is%20a,dominant%20as%20the%20noughties%20passed.)
- Fukuyama, F. (2020, January). *30 Years of World Politics: What Has Changed?* Retrieved from Journal Of Democracy: <https://www.journalofdemocracy.org/articles/30-years-of-world-politics-what-has-changed/>
- Hafidz, M. U., & Sensuse, D. I. (2019). *The Effect of Knowledge Management System*. IEEE.

- Hidalgo, E. S. (2019, March). *Adapting the scrum framework for agile project management in science*. Retrieved from sciencedirect:
<https://www.sciencedirect.com/science/article/pii/S2405844018340635>
- Highsmith, J. (2013). *Adaptive Software Development: A collaborative approach to managing complex systems*. Addison-Wesley.
- Hoffmann, J. (2017, October 22). *A Look Back at the History of CSS*. Retrieved from CSS-Tricks:
<https://css-tricks.com/look-back-history-css/>
- ISO. (n.d.). *MANAGEMENT SYSTEM STANDARDS*. Retrieved from ISO:
<https://www.iso.org/management-system-standards.html>
- Javatpoint. (2018). *History of Java*. Retrieved from Javatpoint: [https://www.javatpoint.com/history-of-java#:~:text=Java%20was%20developed%20by%20James,project%20in%20the%20early%2090s.&text=1\)%20James%20Gosling%2C%20Mike%20Sheridan,sun%20engineers%20called%20Green%20Team.](https://www.javatpoint.com/history-of-java#:~:text=Java%20was%20developed%20by%20James,project%20in%20the%20early%2090s.&text=1)%20James%20Gosling%2C%20Mike%20Sheridan,sun%20engineers%20called%20Green%20Team.)
- Jeff Sutherland, K. S. (2011). *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*.
- Josh Wester, J. T. (2019, November 15). *How Is Technology Changing Democracy?* Retrieved from Providence : <https://providencemag.com/2019/11/how-is-technology-changing-democracy/>
- Kaltenecker, S. (2015). *Leading Self-Organising Teams*. C4Media.
- Microsoft. (2020, 11 12). *Security and compliance in Microsoft Teams*. Retrieved from Microsoft:
<https://docs.microsoft.com/en-us/microsoftteams/security-compliance-overview>
- NetBeans. (2021). *A brief history of NetBeans*. Retrieved from NetBeans:
<https://netbeans.org/about/history.html#:~:text=Jaroslav%20Tulach%2C%20who%20designed%20the,the%20way%20to%20deliver%20them.>
- Oracle. (2017). *Introduction to Java EE*. Retrieved from avaee.github.io:
<https://javaee.github.io/tutorial/overview001.html#:~:text=The%20aim%20of%20the%20Java,responsible%20for%20all%20Java%20technologies.>
- Oracle. (2021). *Java EE FAQ*. Retrieved from Oracle:
<https://www.oracle.com/java/technologies/javaee-faq.html>
- Oracle. (2021). *Oracle Java*. Retrieved from Oracle: <https://www.oracle.com/java/>
- Oracle. (2021). *Oracle JDBC FAQ*. Retrieved from Oracle :
<https://www.oracle.com/uk/database/technologies/faq-jdbc.html>
- Pressman, R. S. (2019). *Software Engineering A practitioner's approach*. McGraw-Hil.
- ProductPlan. (2021). *What is Adaptive Software Development (ASD)?* Retrieved from ProductPlan:
[https://www.productplan.com/glossary/adaptive-software-development/#:~:text=Project%20managers%20John%20Highsmith%20and,Development%20\(RAD\)%20agile%20framework.](https://www.productplan.com/glossary/adaptive-software-development/#:~:text=Project%20managers%20John%20Highsmith%20and,Development%20(RAD)%20agile%20framework.)
- Santos, L. S., L'Erario, A., Pagotto, T., Fabri, J. A., Oliveira, F. S., & Camilo, J. R. (2018). *A Scrum-based process to distributed projects in multidisciplinary teams*. IEEE.

- Saran, C. (2020, May 29). *The software innovation impact of coronavirus*. Retrieved from computerweekly: <https://www.computerweekly.com/news/252483854/The-software-innovation-impact-of-Coronavirus>
- Sequence Diagram Tutorial: Complete Guide with Examples.* (2019, June 13). Retrieved from creatively: <https://creately.com/blog/diagrams/sequence-diagram-tutorial/>
- Sutherland, J., & Schwaber, K. (2020). *The Scrum Guide*.
- Syal, R. (2013, September 18). *Abandoned NHS IT system has cost £10bn so far*. Retrieved from The Guardian: <https://www.theguardian.com/society/2013/sep/18/nhs-records-system-10bn>
- Tyson, M. (2019, April 11). *What is JDBC? Introduction to Java Database Connectivity*. Retrieved from InfoWorld: <https://www.infoworld.com/article/3388036/what-is-jdbc-introduction-to-java-database-connectivity.html>
- UML Class Diagram Tutorial*. (n.d.). Retrieved from LucidCharts: <https://www.lucidchart.com/pages/uml-class-diagram>
- UML Use Case Diagram Tutorial*. (n.d.). Retrieved from LucidChart: <https://www.lucidchart.com/pages/uml-use-case-diagram>
- VisualParadigm. (2021). *Requirement Analysis Techniques*. Retrieved from VisualParadigm : <https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/#:~:text=Requirement%20Analysis%2C%20also%20known%20as,requirements%20gathering%20or%20requirements%20capturing>.
- What is UML Collaboration Diagram?* (n.d.). Retrieved from Visual Paradigm: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml-collaboration-diagram/>
- Zoom. (2019). *About Us* . Retrieved from Zoom: https://explore.zoom.us/about?_ga=2.222333723.1445902788.1608057201-245763263.1608057201