FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

School of *Computer Science & Mathematics*

BSc DEGREE
IN
Computer Science

PROJECT PROPOSAL

Name: Ioannis Armakollas

ID Number: K1527447

Project Title: Teamwork Collaboration Tool (TeCoT)

Project Type: Build

Date:7/10/2018

Supervisor: Graeme Jones

Kingston University London

Did you discuss and agree the viability of your project idea with your supervisor?	Yes
Did you submit a draft of your proposal to your supervisor?	Yes
Did you receive feedback from your supervisor on any submitted draft?	Yes

Declaration

The following declaration should be signed and dated and inserted directly after the title page of your report:

Declaration

I have read and understood the University regulations on plagiarism and I understand the meaning of the word *plagiarism*. I declare that this report is entirely my own work. Any other sources are duly acknowledged and referenced according to the requirements of the School of Computer Science and Mathematics. All verbatim citations are indicated by double quotation marks ("…"). Neither in part nor in its entirety have I made use of another student's work and pretended that it is my own. I have not asked anybody to contribute to this project in the form of code, text or drawings. I did not allow and will not allow anyone to copy my work with the intention of presenting it as their own work.

Date 29/04/2019

Signature : Ioannis Armakollas

Acknowledgement

This has been a great journey, one with a lot of ups and downs and a lot of learning and not learning. If there was one person that I should thank about this experience, it would be professor Graeme Jones, who is one of the people that I would bother non-stop every year since he was both my supervisor any my tutor. In reality, it would have been way more difficult to go through this journey without his assistance and know-how and if there is one thing that I wish I did different, it would be to bother him even more about every single thing because he has always been there and willing to help.

Contents

Declaration 2
Acknowledgement 3
Abstract 7
1.Introduction and Background 8
1.1Problem and Solution 8
1.2 Aims and Objectives 8
1.3 What will the project deliver? 8
1.4 Project Methodology 9
1.5 Work Plan 9
2.Literature Review 10
2.1Introduction 10
2.2Database Security & Threats 10
2.3 GDPR 11
3. Requirements Chapter 12
3.1 Introduction 12
3.2 SWOT Analysis 12
3.2.1 Recommendation 13
3.2.2 Risk Mitigation 13
3.3 Stakeholder analysis 13
3.3.1 Power/Interest Grid 14
3.4 Use Case Analysis 14
3.4.1 Use Case Diagram 14
3.4.2 User Stories 15
3.4.3 Use Case Text 16
3.5 Requirements 17
3.5.1 Functional Requirements 17
3.5.2 Non-Functional Requirements 17
3.6 MoSCoW 17
3.7 Conclusion 18
4 Design Chapter 18
4.1 Introduction 18
4.2 Database Design 19
4.2.1 Entity Relationship Diagram (ERD) 19
4.2.2 Data Dictionary 21
4.3 User Interface Design 21

4.3.1 Introduction 21
4.3.2 Wireframes 22
4.3.3 Navigation 24
4.3.4 The TeCoT Logo 25
4.4 Conclusion 25
5 Implementation Chapter 25
5.1 Introduction 25
5.2 Technologies 25
5.3 Architecture 26
5.4 Database Implementation 27
5.4.1 Designing the Database 27
5.4.2 Creating the Database 27
5.4.3 Populating the Database 28
5.4.4 Generating Data 28
5.4.5 Querying Data 28
5.5 Creating the GUI 29
5.5.1 Task Cards 29
5.5.2 Navbar 31
5.5.3 Expandable Cards 32
5.6 Conclusion 33
6 Validation Chapter 33
6.1 Introduction 33
6.2 Test Tables 33
6.3 Conclusion 41
7 Critical Review 41
7.1 Introduction 41
7.2 Achievements 41
7.3 Problems Encountered and Limitations 42
7.4 Future Plans 42
7.5 Legal, Ethical, Societal and Security Issues 43
7.6 Conclusion 43
References 44
Appendix A -Schedule 45
Appendix B – Initial ERD 47
Appendix C – Volume of Testing Data Expected 48
Appendix E – Use Cases 49

Appendix F – Functional Requirements 51
Appendix G – Non-Functional Requirements 52
Appendix H – MoSCoW 53
Appendix I -Sketches 55
Appendix J – Wireframes 67
Appendix K – Part of the Data CSV 71
Appendix L – Data Dictionary 72
Appendix M – Test Tables 74
Appendix N – Link to the BOX folder 76

Abstract

It is very well known that groupworks are a very big and important part of a university student's life. A lot of the students despise the idea of working together in a group, mainly because there is no guarantee that some of the members will work on the tasks assigned to them or if a team is going to receive feedback to help them improve their project.
To make the groupwork a more pleasant experience for both the students and the lecturers, a tool will be developed that will provide a platform where they can work together, share opinions, ideas and tasks, receive and leave feedback. The tool will aim to increase the productivity and communication within groups by focusing on developing a simple interface with a feedback and an email notification system.

1. Introduction and Background

1.1 Problem and Solution

A university student will be part of multiple groups on different modules during their degree. Those groups will have different goals, different members, different interests and different tools involved.

The main issue is that the students will be introduced to groups and people that they have not talked to before or know. Different groups prefer to use different tools to increase their productivity and communication. Some like communicating through different messaging applications, like Skype or WhatsApp, others prefer emails. All those tools force new users to register into more and more different websites, download and use more tools, share personal details and information. However, the biggest problem with how things are currently done, is the fact that students are unable to share their work and progress with each other and with their supervisors/lecturers to receive feedback.

My client, Professor Graeme Jones, is interested in developing a tool that will allow students to plan and monitor the set of tasks needed to achieve the goal of a groupwork exercise. In addition, lecturers responsible for the exercise need to be able to monitor both the progress of the group as well as the contribution of individual group members to the final project achievements.

The Teamwork Collaboration tool (TeCoT) will focus on improving the communication, productivity and allow students to share their progress with each other and with their supervisors.

1.2 Aims and Objectives

Aims:

- To create a project that will improve the productivity and the communication within groups.
- To improve the way that students receive feedback on a project and its tasks

Objectives:

- Develop a database that will store all the information required for the smooth operation of the system (Users, Projects, Tasks etc)
- Develop a simple and yet concise interface for each different user group.
- Create a feedback system that allows the supervisor to follow a project and its tasks and leave feedback that can be read by anyone that is involved/engaged with it.
- Develop an email notification system for any changes or new feedback on a project to keep users up to date.
- Develop and test the tool on different devices (Personal Computer, Mobile Phone and Tablet).

1.3 What will the project deliver?

A database model that enables students to initiate a project, create and assign tasks and deliverables to group members, as well as capture progress.

Responsive webpage the enables the various user types to monitor, create or modify projects and their available tasks from a variety of platforms.

Allow the students to share their work with their group members and their supervisors and in return receive feedback on it.

A system that will be able to alert users through email and on the system notifications about any changes on any project that they are involved in.

Overall, it will be an all-in-one tool with the goal of decreasing the need of using multiple tools to manage a group's project and will make it so that even the supervisors have an easier time accessing said projects.
The project will also deliver all the data that were gathered during the development process and a prototype with as many of the basic functions as possible implemented.

1.4 Project Methodology

The methodology that is going to be used for the project management will be Kanban. Kanban is an Agile methodology that uses a simple board that is divided into different states of task's progression (e.g. "To-Do", "Doing", "Done"). For this project, the states "Backlog", "In Progress", "Testing" and "Complete" will be used.
The reason Kanban was chosen is because it allows for continuous improvement and the introduction of new things that will need to be worked on. It also gives a very simple to read visual indication of the current progress of the project with the help of the table.
It focuses on doing and finishing with the most important tasks first and reassessing frequently. Making it easy to see what needs to be done and gives a good estimate on the time required to finish a task.

1.5 Work Plan

For the development of this project, an initial plan was created after several discussions with the client that focuses on the step by step deliverables along their deadlines that will result in the product reaching a satisfying stage regarding functionality and documentation.
The full work Plan can be found in Appendix A.

2 Literature Review

2.1 Introduction

The literature review will have as a focus the database security and how an individual could access the data within that database. Furthermore, it will reference and talk about the General Data Protection Regulation (GDPR) and how this law was put in place to ensure that the handling of sensitive and personal data is done properly in order to increase the security and protection of it.

2.2 Database Security & Threats

Even though the storing and handling of data has changed throughout the years, databases are still very popular and are part of the back-end of any service that involves the storage of information. It is also important to note that for the past years, most services have switched to a more digital approach. Banking, shopping, socialising, are only a few of the things that are used by people every day and store an overwhelming amount of personal information in their databases. Banks store names, account balance, transaction history etc. A shop's database will also store past purchases, card details and several more while social media is going to be storing a user's images, names of friends, relationships, messages and many more personal information. In the article "Database Security" (Lipinski, 2011), the writer talks about how simple it is for information to "leak", how a few simple mistakes or a forgetful moment can result in a heap of information being accessed and used in improper ways. It should also be noted that a database breach, depending on the form of access and the use of data afterwards, could go unnoticed for a long period of time. Properly securing data is one of the most challenging steps during the design and development of a new service.

Learning about the possible threats is one of the first step towards correctly implementing a database and securing it. The journal titled "Evaluating Database Security and Cyber Attacks: A Relational Approach."(Bamrara, 2015) does a great job of explaining the forms of attack and the techniques that could help prevent them. The author uses E-Banking as an example since it is one of the areas that has a large amount of breaches and further analyses them. The threats take the form of the attacker gaining access to private data, corrupt, alter or even destroy them. They can also create fake scenarios with fake data and implement malicious procedures. However, an attack does not necessarily have to be from an outside source. A paper titled "A Cloud and In-Memory Based Two-Tier Architecture of a Database Protection System from Insider Attacks" (Moon, Chung, Popovsky, 2014) focuses on insider attacks. It states that this form of attack is equally as important as an external attack and shares some data that points out how common they are. The danger with insider attacks is that there will not be a sign of forced entry since the person will more than likely be or was an employee with access to the data.

The previous source (Bamrara, 2015), does a very good job at compiling and explaining several security techniques which when implemented correctly can ensure that the data is not nearly as easy to access. Passwords are usually the focus of attacks and gaining access to them can result in bigger issues and therefore must be protected. At first, a password was simply a word and a user would only be required to know that word to access a service, also referred to as a "Single Factor Authentication". This method was acceptable for a long time, but it was not as secure as it should be, especially with the volume of data that is currently being stored behind passwords. The next step is something that is being used by many databases that store sensitive data and that is know as "Password Hashing" which is used to replace the contents of a password field with a domain explicit password. Building up on the "Single Factor Authentication", the "Two Step Verification" idea was created and implemented. The goal for this verification method is to ensure that the user who is trying to access a service has a second code which is distributed through other means such as email, a

phone application, a message etc. It adds another layer of protection that is hard to bypass. There are several more techniques that result in the protection of data but are not as known or used as the ones that were mentioned above.
Even though data protection has improved majorly the last few years, there is no guarantee that there never will be unauthorized access to one's data. A user should always be able to oversee their data and have full knowledge and control of who handles the information and why. This was one of the main reasons the EU passed a new regulation, more commonly known as GDPR.

2.3 GDPR

According to the European Commission's website (European Commission, 2019), due to the increase of the popularity of online services and therefore the transition to a more digital age, over 90% of Europeans requested for data protection laws across the EU. On the 24th of May 2016, the General Data Protection Regulation (GDPR) entered into force and did not fully apply up until the 25th of May 2018.The aim of the regulation is to protect the Personal Identity Information (PII) of an individual. James Fisher, a GDPR & Data Protection Consultant, writes in his article (Fisher, 2019) about the risk-based approach of the GDPR. In the context of data handling, risk is defined as physical, material on non-material damage that could result from the processing of data (e.g. identity theft, fraud etc.). It is important to note that the GDPR focuses on only two risk levels and those are "Risk" and "High Risk". Any form of data processing that is seen as "High Risk" will have to perform a Data Protection Impact Assessment (DPIA) which enforces some minimum requirements to the party that handles the data processing (e.g. Disclosing the purpose of the data process). The regulation is way more impactful than it sounds, especially towards the companies that process the data. The article "Weighing the Impact of GDPR" (Greengard, 2018), emphasizes on the penalties that are to be established if a business does not comply with one of the grander penalties being a fine of up to 4% of the company's worldwide annual revenue from the previous fiscal year. A customer is also granted the ability to request the complete removal of their personal data from a company's database, at any given time. This results in several companies voicing their concerns about how this change in data processing is hindering innovation. It greatly affects the progress in artificial intelligence (AI) since it relies heavily on data processing and it also places companies in a position in which they would have to consider using multiple databases, one of which would be handling the EU data with all the requirements placed by the GDPR. It is not clear yet as to how this regulation is going to impact this "digital age". It could affect it greatly and change everything about how data is being handled or it might end up getting written off and replaced by something more efficient.

3. Requirements Chapter

3.1 Introduction

The aim of the requirements chapter is to display the steps taken to identify most if not all the main requirements for the project. The focus of this chapter are the actors and the possible actions that they will be taking while using this tool. By understanding and identifying how the tool works, what are its strengths, its weaknesses, its purpose and the users' expectations, it will result in a thorough list of the requirements that will assist with the development and further improvement of the tool.

3.2 SWOT Analysis

SWOT analysis is a commonly used technique during the project planning process that will assist in finding out which areas of the project will require extra focus and consideration. The goal of the SWOT Analysis is to weigh the Strengths and the Opportunities against the Weaknesses and Threats of the project. That will allow us to decide if we should continue with the development of the tool simply by noticing if the positives outweigh the negatives and if we can do anything to mitigate the risks.

Strengths:
• **Increased productivity by providing the user with the required tools for managing a project.**
• **Easier to communicate with team members and share opinions and feedback**
• **Easier for supervisors to monitor students and projects.**
• **Detailed information regarding the progress on the project and the contribution of each member.**
• **Allows members to clearly divide responsibilities and roles between them.**
• **Accessibility on different types of devices like phones, laptops and tablets.**
• **Assigning, updating and finishing tasks will be fast and easy**
• **Users will be able to let their teammates know about any updates or changes through email notifications or even a personal/group messaging system that will be built into the tool itself.**
• **Each group will be able to set their own unique tasks on their project (even if the whole class has the same exact project).**

Weaknesses:
• **Unable to use if you do not have access to a device that is able to run the tool.**
• **People will be more inclined to work from home with far less face to face interactions between them and their group.**

• **Users will have to use this new tool which might take some time to get used to.**
• **One person working on the project.**

Opportunities:
• **Database structure allows for future additions and improvements.**
• **Potential to branch out and assist not only university students with their projects but possibly more people.**

Threats:
• **System is online and therefore runs the risk of having issues with the host/server.**
• **Servers could be hacked into and have data altered or stolen.**
• **Possible data loss in case of a server issue.**
• **Another company might end up developing a similar tool making this one obsolete.**

Table 1 – SWOT Analysis

3.2.1 Recommendation

Taking the SWOT into consideration, the project would be beneficial for students and lecturers alike. Giving the users the flexibility to work on their groupwork the way they would prefer and will allow lecturers to easily access the groupwork or its tasks and leave some feedback. The main issue that could deter anyone from investing or using the tool is the online-only focus. A lot of users will have to "risk" losing or getting their projects, and/or their data stolen on a possibly vulnerable platform and while it is an understandable worry, it can be solved by following some very standard practises regarding the security of a website and its database.

3.2.2 Risk Mitigation

The tool will have to get properly implemented in the most commonly used device types to enable any user type to access it wherever and whenever they want. The tool needs to be approachable, easy to use with a clear and simple UI to reduce the time required for a user to learn and get used to it which will assist with a smoother transition from previously used software. Regarding the data handling, file encryption along with regular backups and a firewall will be used to ensure that there is no data loss or unauthorised access to the database and the data. Considering any possible competition in the future, one of the aims is to make it easier to upgrade it later by making it possible to edit anything on the website include the databases without that many restrictions.

3.3 Stakeholder analysis

A stakeholder is a person or a group that is involved with the tool and could be affected by all the changes that will come with the new tool. They are also the main influence when the requirements are discussed. This analysis will help identify the importance of each stakeholder group and therefore the requirements that should be included to cover most of their needs.

The main stakeholder group will be the students. A student is going to be directly influenced with the introduction of a tool specifically developed to assist them with all their projects and groupwork. They would have to switch from older methods and then start learning and using the new system.
The lecturers/supervisors are another stakeholder because they will have to be more involved with the projects and the students since they will have to follow, manage and give them feedback. They will have to learn to work with a system that requires them to interact with all the projects and their management directly instead of just waiting for the submission date.
The system will also need administrators who will have to learn how to use and manage the tool and to provide the best possible support to the users.
Lastly, other companies (besides the university) could be considered a stakeholder group since the tool might be outsourced and made accessible to a wider range of users. Therefore, it should be developed with the flexibility to be accessed and used by users that are not in the university.

3.3.1 Power/Interest Grid

The Power/Interest grid or matrix is a simple and yet very helpful tool that clearly displays the position of any current and maybe even future stakeholders based on their power and interest on the tool, its development and any other future possibilities. It will be used to identify the main user groups that the tool should be designed and catered to.

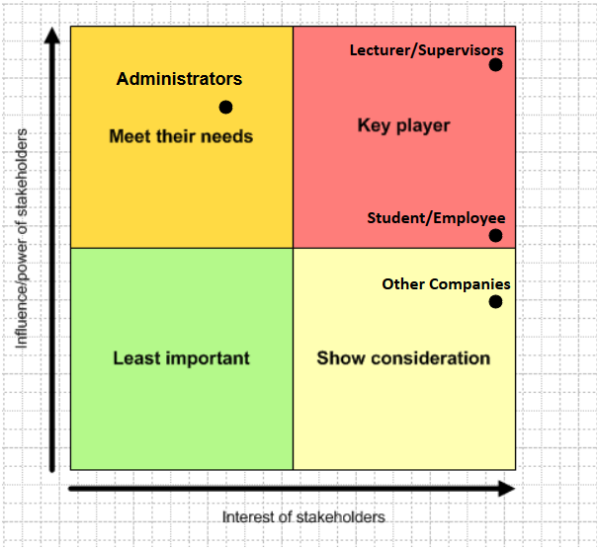

Figure 1 – Power/Interest Grid

3.4 Use Case Analysis

The Use Case Analysis is used to break down the various tasks that a user can perform. There are different approaches and steps that help with this analysis. Steps, such as creating a Use Case Diagram, User Stories and Use Cases will be used in the next section. One thing that helps with the Use Case Analysis is if each point is approached from the user's point of view to further understand what the requirements are going to be.

3.4.1 Use Case Diagram

The Use Case Diagram exists in order to identify the user groups and all the tasks linked to each of them based on the project's Use Cases. The box is used to represent the system and inside it; all the bubbles are used to show all the different possible actions that a user (the people outside of the box) should be able to take.

Figure 2 – Use Case Diagram

3.4.2 User Stories

User Stories are simple sentences that aim to give an insight on what a User might be expecting from the tool. After the identifications of the user groups, interviews or speculations might be conducted to conclude to the stories that will be used to help with the list of requirements.

Student:

- As a student I would like to be notified when there is a new update or feedback on the projects that I am involved in through an email notification system.
- As a student I would like to read the feedback left by my lecturer under the respective task/deliverable.
- As a student I would like to have the option to create and assign my own/unique tasks for an activity to my group members and then be able to see who is responsible for each task and a clear indication of that task's progress.

Lecturer:

- As a lecturer I would like to closely monitor the progress of my students on the activities and their tasks and be able to leave feedback on them.
- As a lecturer I would like the ability to see who is contributing in a project which will help with grading everyone appropriately.

- As a lecturer I would like to give my students the options to send me an email notification when there is a task that they would want me to check and give some feedback on.

3.4.3 Use Case Text

A Use Case Text takes the Use Cases (shown in the diagram above) and expands on them by describing step by step the process, the requirements and the pre and post conditions. That helps identify all the dependencies, the steps that a user will have to take from the beginning, and they are also extremely helpful when it comes down to testing the functionality of the tool itself.

Use case number	**UC1**
Name	Create Groups
Description	Lecturer creates new groups for the new project that they assigned in the lecture
Actor	Lecturer
Pre-Conditions	Lecturer is logged in the System.
Basic Flow	1. Click on “Create new groups” 2. Set Group information 2.1 Select Year 2.2 Select Module 3. Select number of Groups 4. Click “Done” 5. System creates the group 6. System sends an invite notification to the selected students
Extensions	
Post Conditions	Groups created Members assigned

Table 2– Lecturer’s Use Case Text

User case number	**UC5**
Name	Create Task
Description	Student creates a task for the project
Actor	Student
Pre-Conditions	Student is logged in. Student is part of a group. Group has a project.
Basic Flow	1. Click on “Projects” 2. Select the Project 3. Select “Create New Task” 4. Add “Title” and “Description” 5. Add a “Goal” 6. Click “Done”
Extensions	System notifies the rest of the group members about the new task.
Post Conditions	A task has been created and students have to assign themselves to it.

Table 3 – Student’s Use Case Text

3.5 Requirements

The requirements section will document some of the users' expectations from the system by using simple tables and IDs for each requirement so that they can be referenced in a later stage of the development process.
Requirements are divided in two main categories, the functional and the non-functional.

3.5.1 Functional Requirements

The functional requirements are referring to the functionality of the system and describe what is should do. They are actions that are available to the users. Below is part of the functional requirements table, the full table can be seen in Appendix F.

ID	Functional Requirements	Users
1	Register Account	All Users
2	Log In	All Users
3	Log Out	All Users
9	Create Task	Student
10	Assign Task	Student
11	Un-Assign Task	Student
20	Create Activity	Supervisor
21	Delete Activity	Supervisor

Table 4 – Functional Requirements

3.5.2 Non-Functional Requirements

Non-Functional Requirements aim to set certain expectations on how they system works. Efficiency, Accessibility, Security, Stability etc. are all considered non-functional requirements and are also extremely important for the system to work properly. The full NFR table can be seen in Appendix G

ID	Non-Functional Requirements
1	Interface should also be mobile/tablet friendly
2	Sensitive Data should be encrypted in the database
3	System should remain functional under heavy traffic
4	Pages should not take longer than 2 seconds to load
5	System should log the user out after a period of inactivity

Table 5 – Non-Functional Requirements

3.6 MoSCoW

A system has multiple requirements, however, not all of them are equally important. There are a few requirements that are simply quality of life improvements but also some that without them the system would never be able to run as intended. That is why MoSCoW (Must Have, Should Have, Could Have, Won't Have) will be used to assist with the prioritisation of those requirements.

Functional Requirement	Description	Priority

Register Account	A user will have to register/activate their accounts when using the system for the first time.	Must Have
Log In/Log Out	A user will have to be able to log-in and out of the system	Must Have
View Task Progress	A task should have a progress status in which a user responsible for it can set the different progress states.	Should Have
Request to join a group	A Student should be able to show interest and request to join a certain group. It is up to the Supervisor to decide whether the Student can or cannot join a group.	Should Have
Request Feedback	A Group or Student should be able to request Feedback on tasks or the Project in general from their Supervisor whenever needed.	Could Have
Create Announcement	A Supervisor could be able to create an announcement for all the groups involved in a Project	Could Have
Download/Print Projects and its contents	Downloading and printing anything regarding a project will not be required since everything will be editable and stored on the web page.	Won't Have
Students create and join the groups they want	Students will not be able to join the group of their choice and that is a restriction to keep the whole process under the control of the supervisor.	Won't Have

Table 6 – MoSCoW table

3.7 Conclusion

The proper identification and categorisation of the requirements is one of the most important steps to take at the beginning of the process since it sets the foundations that supports the project during its development phase. However, the requirements are not set in stone and there is always a chance that some requirements could be added, deleted or changed. That is one of the benefits of working with an Agile methodology and it allows for even more improvements and changes later in the future.

4 Design Chapter

4.1 Introduction

The design chapter will aim to show the step-by-step process that will transform all the information gathered during the requirements chapter into a product that the user will be able to work with in the future both front-end and the back-end. The chapter will start by talking about the database design of the system which will mainly consist of the Class Diagram, the ERD (Entity Relationship Diagram) and the Data Dictionary. The design stage of the project will also affect some of the previously made assumptions and decisions. The changes will be related to certain issues that will arise after the implementation of the previous steps or things that were not well thought out and therefore missing some parts.

4.2 Database Design

The database is the backbone and a very important component for this tool. It is crucial that the design is complete without any mistakes or with the possibility to edit certain things, relationships and data without having to recreate everything. Therefore, this part of the design is going to emphasize on the step by step process that ended up with the creation, population and implementation of the database and its data.

4.2.1 Entity Relationship Diagram (ERD)

The ERD is used to help visualise the database, all the tables, the columns and the connections that will exist between said tables. The tables and their attributes are represented by tables with a title and a list. The title contains the name of the table and the list underneath the title will contain the names for each attribute within the table. Furthermore, some attributes will contain the letters (PK) or (FK) right next to them and that is used to indicate which of them are the Primary Keys of the table and which are the Foreign Keys that link to a different table. The links between the tables are also visible and used to show the relationship between them. It is important to understand that in order to reach and retrieve certain information, a clear and accessible path has to be established beforehand and the ERD will help visualise that path.

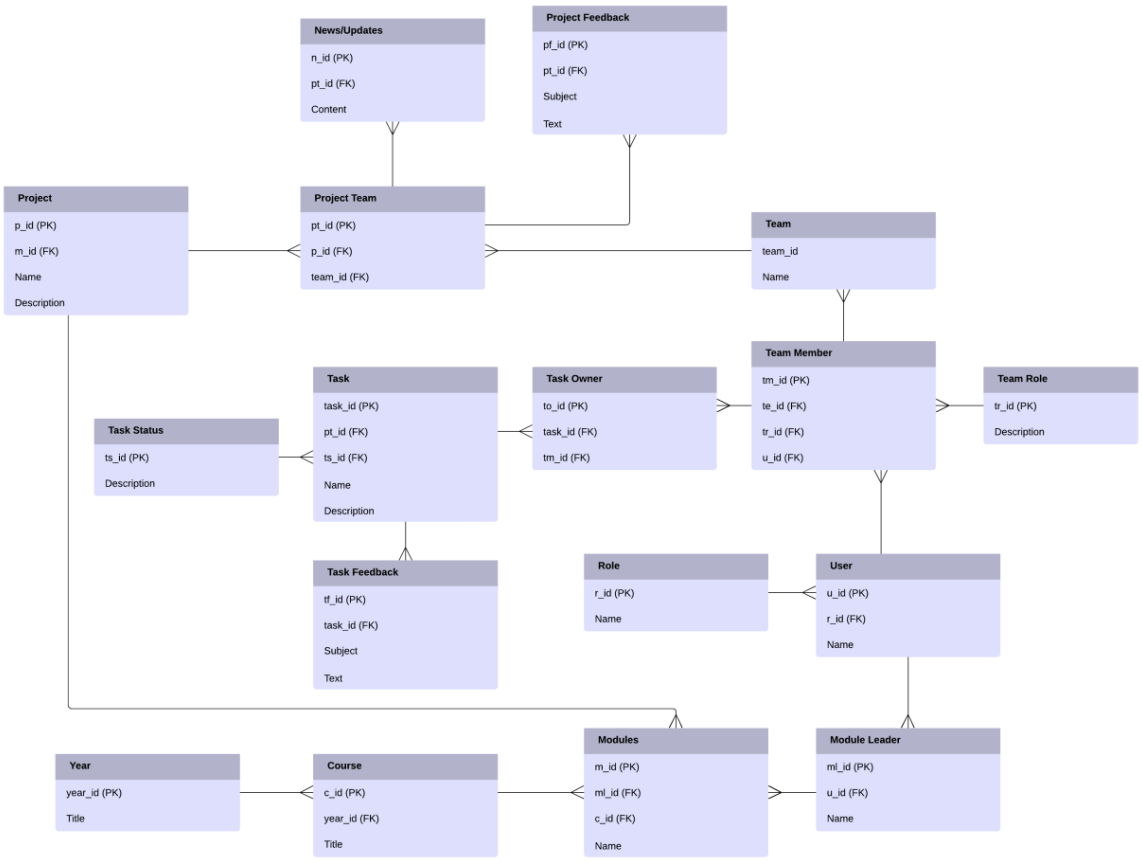

Figure 3 – Entity Relationship Diagram

Figure 1 Contains the full image of the ERD with all the currently necessary connections. Parts for the ERD can be broken down into different sections in order to further explain the design process and choices behind the tables and the relationships between them. The aim of

the next section of this chapter is to explain some of the sections that were created during the break-down with as much detail as possible.

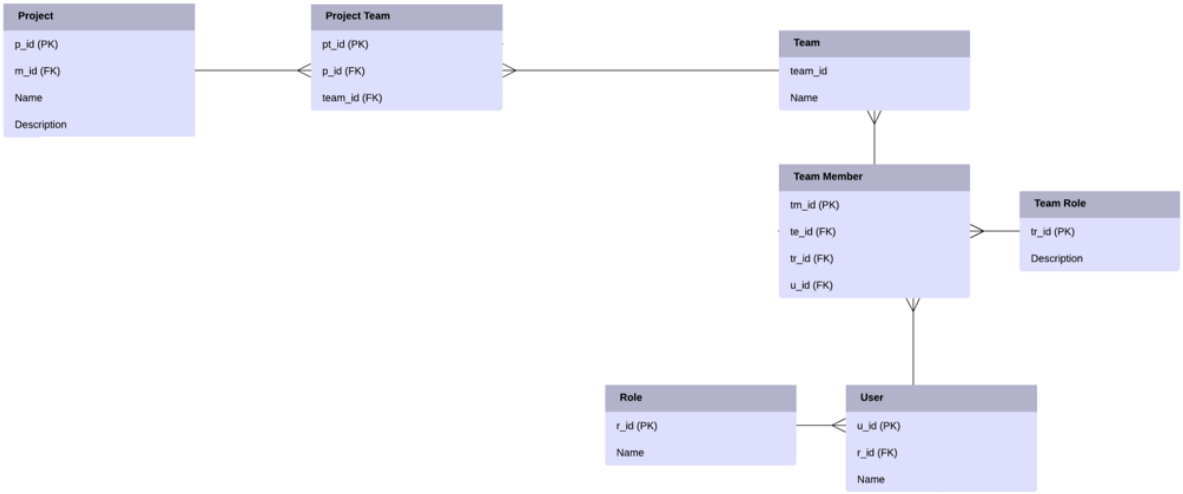

Figure 4 – Team and Project Info

This first section shows the relationship between the “Users”, the “Teams” they could be part of and then the “Projects” that are assigned to those “Teams”. As previously mentioned, the way the tables are connected in the ERD and in the Database will help with understanding the relationship between them and then retrieve the information required by a table that is not directly connected to the table that connects said information. In this case, a user can have projects assigned to them and they could also check all the details. That is possible mainly because a “User” is part of a “Team” and a “Team” can have a project assigned to it and so, any “User” that is related to the “Team” with the “Project” will therefore have that “Project” assigned to them.

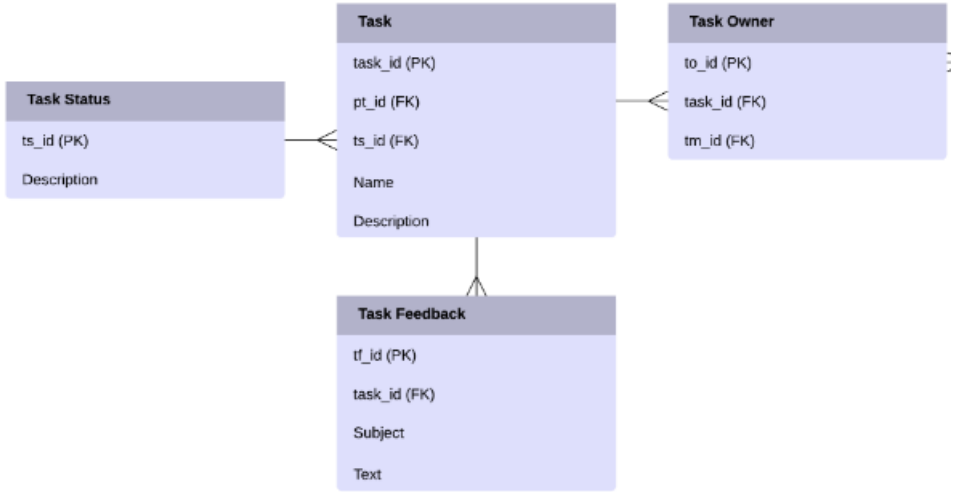

Figure 5 – Project Task Information

Figure 3 is focused on everything that is related to a project’s Task. Tasks are part of a “Project” and a “Project” will have multiple Tasks in many occasions. Each “Task” will have “Users” assigned to them and that is the main use of the “Task Owner” table (which is also used to connect the “Task” table with the “Users” table and helps resolve the many-to-many relationship that existed between them). A set of statuses will be included in the “Task Status” table which will then be used by the “Task” table in order to show the current status

of a “Task”. Finally, the “Task Feedback” table is used to store the feedback that was posted for a “Task”.

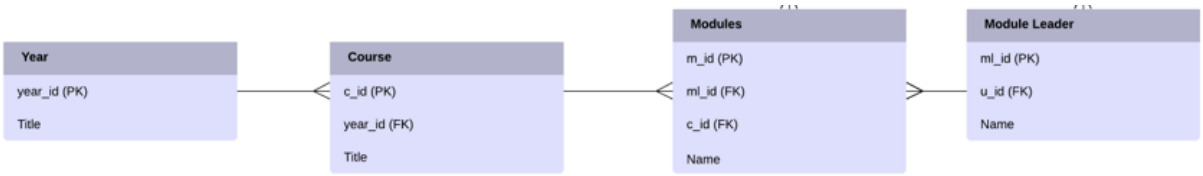

Figure 6 – Module Information

The Module Information section will be used to store any information related to the modules in detail from the “Year” and the “Course” to the “Module Leader”. This is also a pretty important section that will be assigned to the “Teams” and “Users” table to allow for a simple categorisation based on module that they are part of.

4.2.2 Data Dictionary

The Data Dictionary is used primarily to check all the information and the information types required for the database. It contains the field names which represent the titles of each of the table’s columns, will also inform us whether it has any constraints, if it is a primary and foreign key, the datatype and the length of the data. It will also be used to look back and confirm that all the data has been properly set up in every table.

Table Name	Field	PK/FK	Datatype	Null
User	u_id	PK	Varchar(25)	No
	r_id	FK	Int(6)	No
	Name		Varchar(25)	No
Role	u_id	PK	Varchar(25)	No
	Name		Varchar(25)	No
Module Leader	ml_id	PK	Int(6)	No
	u_id	FK	Varchar(25)	No
	Name		Varchar(25)	No
Modules	m_id	PK	Int(6)	No
	ml_id	FK	Int(6)	No
	c_id	FK	Int(6)	No
	Name		Varchar(25)	No
Course	c_id	PK	Int(6)	No
	year_id	FK	Int(6)	No
	Title		Varchar(25)	No

Table 7 – Data Dictionary

4.3 User Interface Design

4.3.1 Introduction

The User Interface Design section aims to show how the front end of the system will look like, the wireframes and the navigation section will demonstrate what a user will have to interact with and the step by step process that they need to perform in order to move through

the system and all the available pages. It is not in any way supposed to look exactly like the final product but will mostly demonstrate just the basic design ideas.

4.3.2 Wireframes

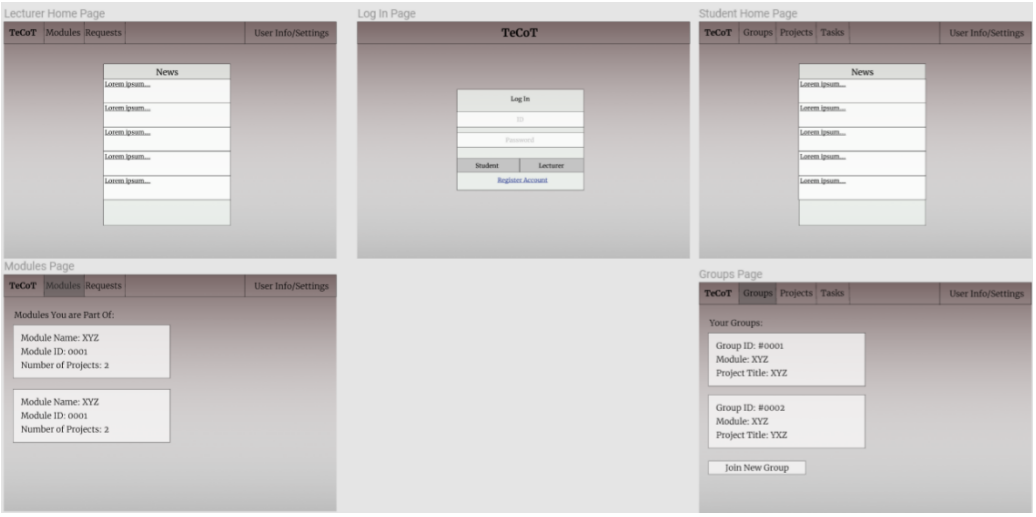

Figure 7 – Wireframes

The wireframes represent how the final pages could possibly look like. Creating clear and interactive wireframes can be extremely helpful when working on the final product. In this case, we will demonstrate just a simple workflow and how it will look with some help from the wireframes that will be included. The workflow will start from the log-in page and it will show the step by step process that the user will be required to follow in order to see the information related to a group that they are part of.

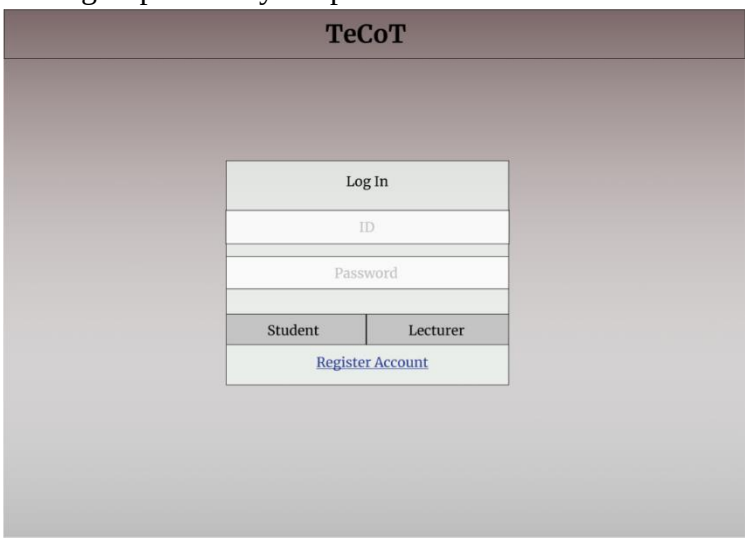

Figure 8 – Login Page

In figure 15, there is a simple wireframe of the current expectations of the log-in. The final product will be expected to retrieve the userId from a database and use that as a username. The userId will have unique identifiers that separate the different user roles and will not ask for the user to select between a “Student” or a “Lecturer” login.

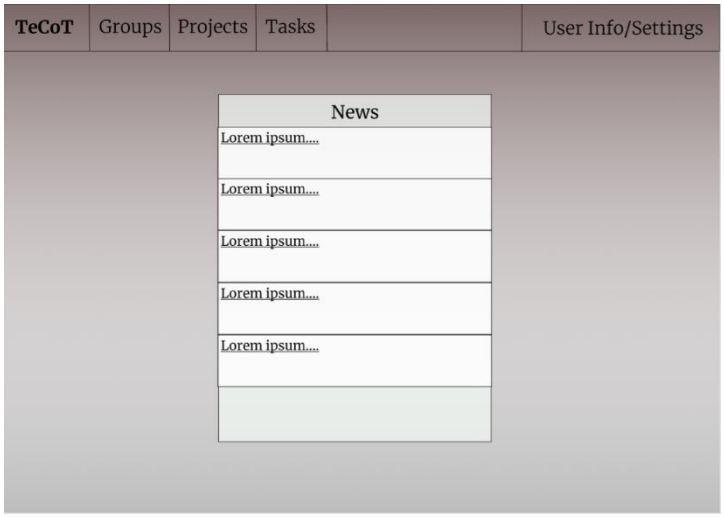

Figure 9 – Student Home Page

The Student Home Page can also be referred to as the Student Landing Page and is the first page that the user will get redirected to after logging in. This page will contain information and news relevant to the user and allow them to navigate to everything else using the bar that is located at the top of the page.

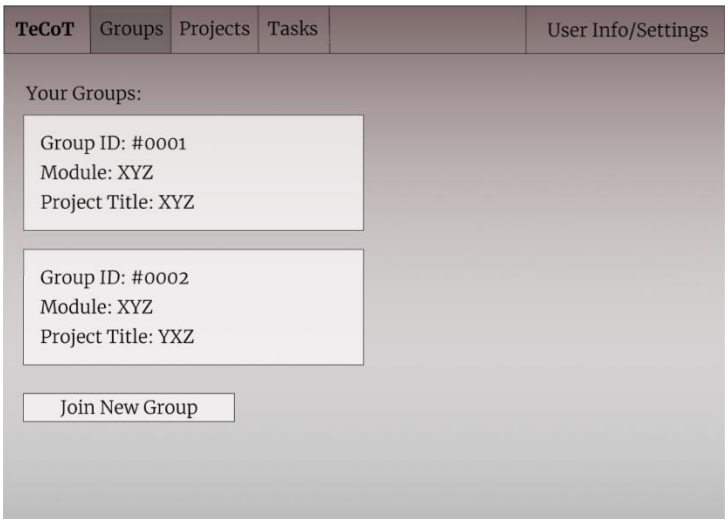

Figure 10 – Groups Page

At this stage, the user will be led to a page that includes all the groups that they are part of with some minor information in order to more easily identify each group to help them decide which group's info they want to investigate.

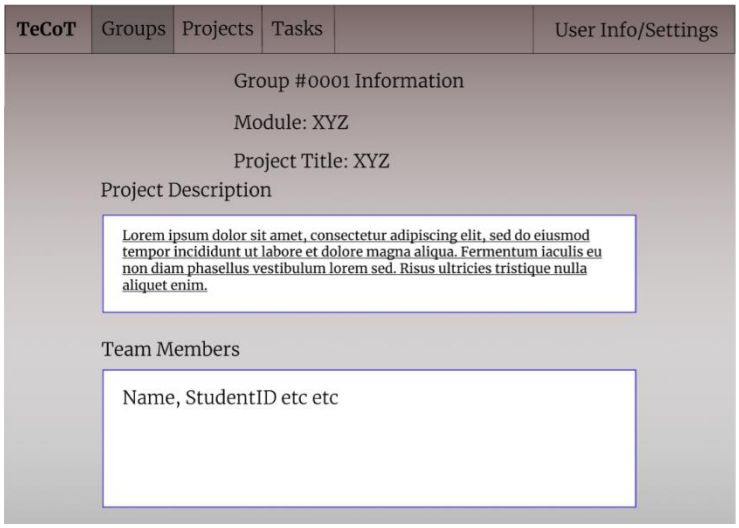

Figure 11 – Group Information Page

The final page for this process, the user reaches a page that includes all the information of a group in detail and will later allow them to interact with certain aspects like selecting a team member they would like to contact in order to get their contact details.

4.3.3 Navigation

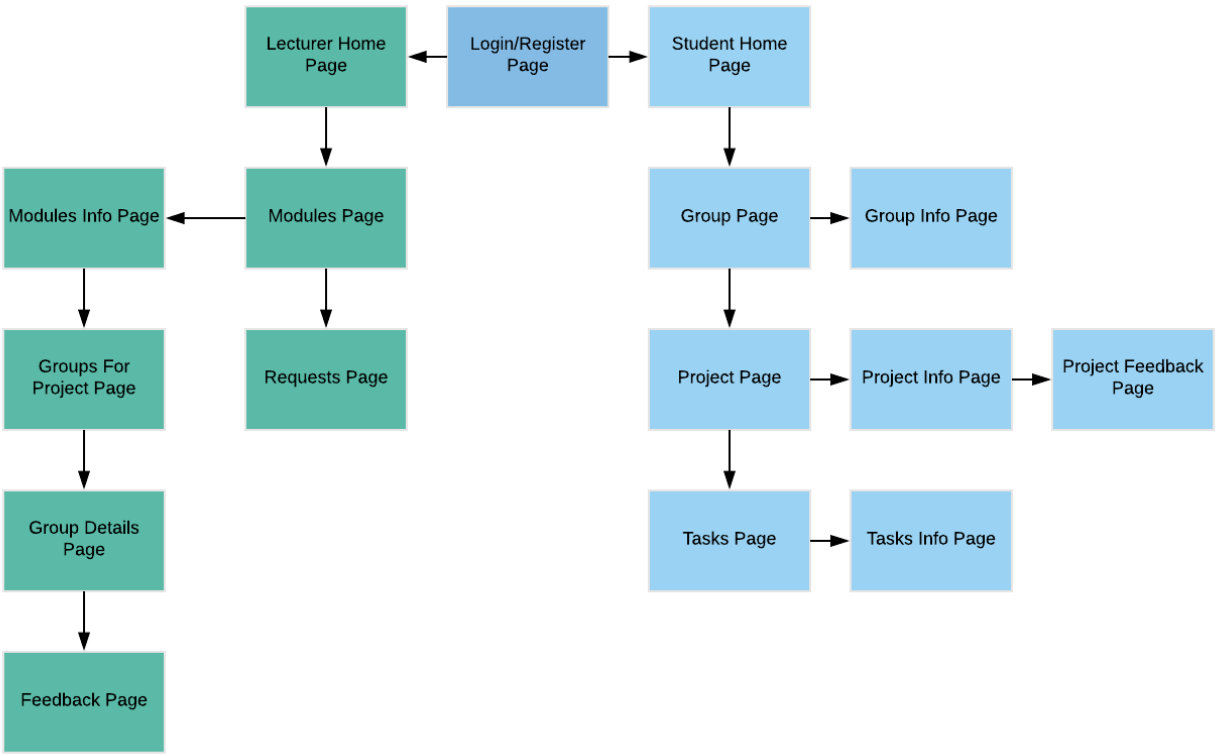

Figure 12 – Navigation Diagram

The navigation diagram shows the process that the user goes through in order to navigate through the system. The green(left) side is the path that only the supervisor can navigate through while the blue(right) side is student's options and side. The diagram is clear and concise to help developers and the client understand the steps and the complexity of the system.

4.3.4 The TeCoT Logo

One of the last steps of the design was to create a logo that will match the intended style of the website. Something minimalistic, clean and overall small that can easily be placed on the top right corner of the page, right inside the navigation bar.
It is a simple vector image that was created with a software called “Inkscape”. The idea behind the design are the main letters of the Team Collaboration Tool, “T” and “C”, along with the acronym right underneath the actual logo for extra clarity.
There are two different versions, one with the black lettering and the other with white lettering. The first will be used in the log-in screen since it will have to stand out on a white background while the white version will be the one placed on the navigation bar which has a darker background.

Figure 13 - Logo

4.4 Conclusion

The design chapter shows the main steps that were taken in order to transfer everything theoretical that was mentioned in the requirements chapter into something more visible and practical, something that a user could interact with. One of the most important steps taken during the design phase was the establishment of a working database with all the data and connections required. The wireframes that will give an idea of how the system will look and the navigation diagram that will help with visualising the processes and steps available for certain tasks.
There are differences between each step of the design process, there are differences even from the requirements chapter and that is mainly because through the implementation, new issues or ideas arise that result in slight changes of the design and this chapter is here to show some of them.

5 Implementation Chapter

5.1 Introduction

The purpose of the Implementation Chapter is to document the step by step process that took place in order to create and implement the product and its functions. The tools that were used during the development process, the back-end and the front-end steps, issues and the possible future plans that must be taken into consideration for all the updates that will result in an improved and complete product.

5.2 Technologies

Text Editor

In order to develop the system, the main tool that will be used throughout the whole process is going to be a text editor. The software that was chosen for this job is called “Visual Studio Code” or “VSC”. The reason it was picked over other text editors is due to the vast amount of addons developed for it that allow for a more personalised and more specialised experience. It is clean and simple to learn and use even though there are more simple text editors out there if simplicity was a concern.

HTML/CSS
HTML/CSS is used for the overall presentation of the website. They are used in order to provide the front-end side of the tool. HTML is the code that dictates what everything on a webpage is and where it should be while the CSS is used for stylisation and help with how everything should look.

PHP
PHP is used for the data handling and Database interaction. There were other options that could essentially replace the PHP (such as NodeJS) but due to XAMPP (a software that will be mentioned later in this section) and also the universities servers both using the PHP language as a default it was decided to not attempt to overcomplicate the project for some smaller tweaks since PHP is still viable with just a few issues that can be patched up.

JavaScript
JavaScript is used for the logic and interactivity that can be seen on the system. Some things cannot be done just by using HTML and PHP. JavaScript handles more complex jobs that help create a more interactive, functional and visually pleasing webpage.

Bootstrap
Bootstrap is a framework and could be considered one of the more important tools that were used in order to improve the overall user experience. It comes with a variety of features and graphics that result in a highly responsive web-page. It allows for several new approaches while designing and helps simplify some of the more tedious jobs.

phpMyAdmin
phpMyAdmin is the tool that will be used for the administration of the MySQL database that will be used for this project. The interaction with the database is done using PHP and MySQL which is a considered one of the better options when designing something that is expected to mainly interact with databases. Another reason that phpMyAdmin was chosen was due to both XAMPP and the KUNET servers using phpMyAdmin as the default administration tool.

XAMPP
XAMPP is a tool that provided a simple local-hosting option while developing offline. It comes with several tools such as “Apache”, “MySQL” and “FileZilla” that can be considered essential for this type of development. Very easy to use and setup and helps by making it even easier to update and change small things instead of having to edit, save and upload new files every time there are any changes.

Web Browsers
Several web browsers were used during the development of the tool. Google Chrome was mainly used due to the easy to use developer tools. Other popular browsers such as Firefox and Edge were also used to ensure that the website functions as expected in all of them to allow the user to access it with their preferred tools.

5.3 Architecture
The main architecture goal is for the whole system to be accessible and functional on most of the devices and browsers that would be expected to be used today by students and lecturers alike. After interviews, discussions and further consideration, it was decided that it should be available on most of the modern browsers while also making sure that the webpage adapts to the device that will be used to access it with a focus on desktops, laptops and phones. This

goal can be achieved with the combination of HTML/CSS, Bootstrap and a bit of JavaScript to cover some of the interactive elements that Bootstrap is unable to cover. Bootstrap is the key component when it comes to the adaptability and the scalability of the system between the different devices due to the features implemented. Furthermore, PHP and MySQL will be used for the database interaction and will be considered and referred to as the back-end of the webpage.

5.4 Database Implementation

Correctly designing and implementing the database is one of the most important steps during the development of TeCoT because, even if it looks and operates flawlessly in the front-end, the focus is data handling and sharing. Therefore, the database table structures, the relationships, the data and the interaction with the front end should be accurate and always return the expected results.

5.4.1 Designing the Database

The database design was firstly mentioned in the design chapter of this document. The first step was to identify all the different tables that would be needed, normalise them and create an initial ERD with the intention of changing it and updating it whenever needed. After identifying all the tables, the primary and foreign keys, the data type for each field had to be considered and included which resulted in the creation of the data dictionary.

Modules data

Field	PK/FK	Datatype	Purpose	Null
m_id	PK	Int(6)	Records unique ID of a module	No
ml_id	FK	Int(6)	Records ID of the module leader	No
c_id	FK	Int(6)	Records ID of the course	No
Name		Varchar(25)	Records name of the user	No

Table 8 - Data Dictionary of the "Modules" table

5.4.2 Creating the Database

During the development, even though "phpMyAdmin" comes with built in tools that allows for the easy creation of tables by filling in forms of data without having to be concerned about prior "SQL" knowledge, the decision was made to create and document the code behind the creation of all the tables using "SQL" statements. Even though it is more complicated at first, restarting the whole process to solve an issue is easier and faster since it is possible to just store the working code, edit the parts that need editing and then re-create it with the updated data.

```
CREATE DATABASE project
COLLATE utf8mb4_unicode_ci;
USE project;

CREATE TABLE `Role_Table`(
    `RoleID` INT(6) UNSIGNED NOT NULL AUTO_INCREMENT,
    `RoleTitle` VARCHAR(30) NOT NULL,
    PRIMARY KEY (`RoleID`)
);

CREATE TABLE `User_Table`(
    `UserID` INT(6) UNSIGNED NOT NULL AUTO_INCREMENT,
    `RoleID` INT(6) UNSIGNED,
```

```
    PRIMARY KEY (`UserID`),
    FOREIGN KEY (`RoleID`) REFERENCES `Role_Table`(`RoleID`)
);
```

Figure 14 - Database and Table Creation SQL

5.4.3 Populating the Database

After all the tables have been created, it is time to populate them with data. The first step is to try and identify the volume of data that should be used to emulate how the final, live version will possibly contain and test if the database will be able to work with it. The second step will be to generate the data that will be inserted in the database.

5.4.4 Generating Data

For starters, the tool that will be used to generate the data will be Excel. Each Excel file will correspond to a table in the database. The reason behind the usage of Excel is simply because of the ability to export as a CSV (comma-separated values) which is recognised by "phpMyAdmin" and allows the database designer to edit and change data without having to go through the clunky "phpMyAdmin" interface. "IDs" were easy to generate using Excel to generate numbers that increase by 1 in each row. A name generator (Random-name-generator.info, 2019) was used to populate the "Names" column with data. For all the different "Courses", "Modules" etc, the table was populated with data from a university's webpage (Le.ac.uk, 2019) which includes a list of the courses and the modules within each course in an easy to copy format. Lastly, the foreign keys in each table were created by using the "=RANDBETWEEN (x, y)" line in Excel, where X is the lowest possible number and Y the highest possible number for that foreign key. This resulted in functional but also interesting combination of data.

5.4.5 Querying Data

The final step will be to make sure that the database displays the correct data when asked to. To achieve that, a complicated query that consists of multiple JOINS will have to be written. If the query successfully displays the expected data without any errors, the database is functioning correctly. For the first test, the query that will be used is a rather important one.

```
SELECT * FROM projectteam
            INNER JOIN project ON projectteam.projectid = project.projectid
            INNER JOIN team ON projectteam.teamid = team.teamid
            INNER JOIN teammember ON team.teamid = teammember.teamid
            INNER JOIN user_table ON teammember.userid = user_table.userid
            INNER JOIN module ON project.moduleid = module.moduleid
            WHERE project.projectid = '1' AND user_table.userid = '1'
```

Figure 15 – SQL Query

This SQL is attempting to display the content in a box that will include all the information related to a project that a user is part of. From "ProjectTeam", we connect to "Project" and "Team. "Project" contains the "Title", the "Description" of the progress etc. The "Team" table contains the "Team" name but will also enable the connection to the "TeamMember" table using the "TeamID" and then that will lead to including the "User_Table" using the "UserID" which will be required in order to find and display the projects that a user I part of. The "Module" table is linked for some extra information regarding the "Project" (mainly the name of the module the project is part of).

5.5 Creating the GUI

The GUI was built by using HTML/CSS and PHP elements combined. For the CSS part, Bootstrap elements were used with a combination of different custom CSS files.

The aim of this design is to create a responsive and minimalistic design without sacrificing any of the functionality or the information that the system can provide.

The remainder of this section will focus on key design elements, the code and the choices behind them.

5.5.1 Task Cards

A very important aspect of the tool is creating tasks for each project and setting a status to ensure that all the team members can see and understand the priority of each task. A user's first encounter of the project tasks is within the "Project" page, right underneath the project's description. On this page, it will only display the four newest tasks to emphasize on the latest changes and keep the users as up to date as possible without flooding the page with information. An alternative would be to show the tasks with the higher priority, but that approach would defeat the point of displaying a summary of information to the user.

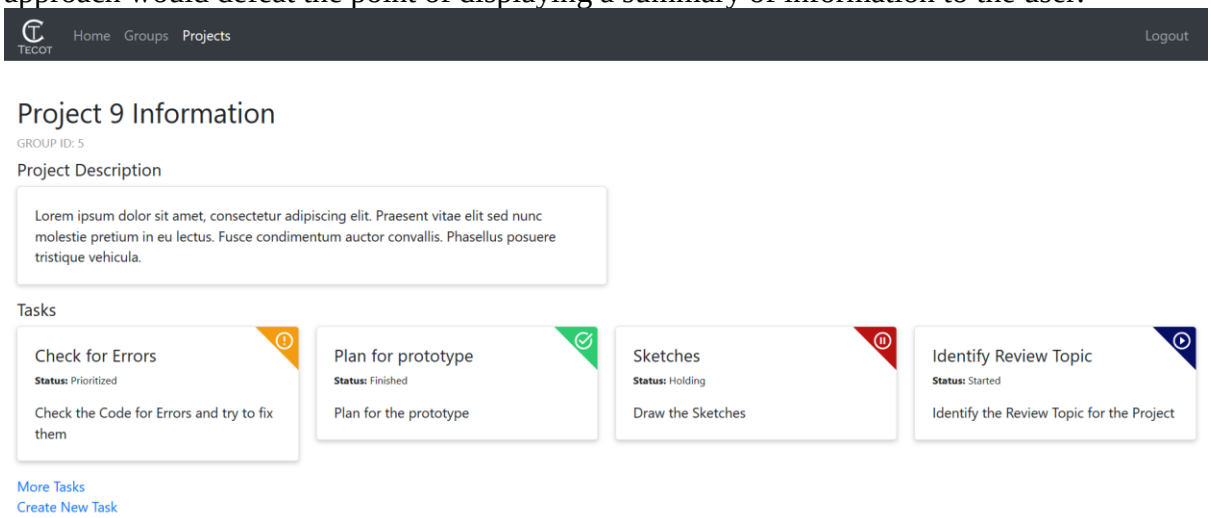

Figure 16 - Tasks in the Project page

The task page will include all the project's tasks. Each task is nested within a card element with all the necessary information. The cards are colour coordinated with each colour representing a different status. The initial idea was to completely change the colour of the whole card to represent the status. However, that would not work well with the attempt towards a minimalistic effect and therefore, the idea changed into just colouring the corner and including an SVG (Scalable Vector Graphics) image that ensures that users that could be colour blind would be able to tell the status with just a glance.

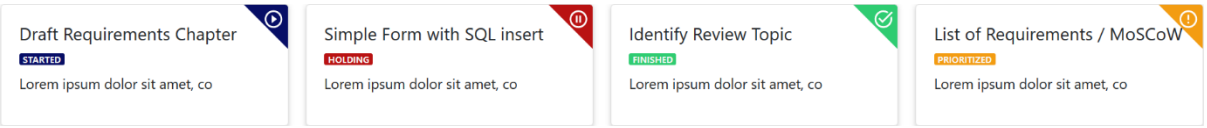

Figure 17 - All the different task statuses

Furthermore, the tasks will be displayed on the page with their prioritised order. The current order is "Prioritised", "Started", "Holding", "Finished".

The SVG images were gathered from Icons8 (Icons8.com, 2019) and were used in an if statement that is checking to see if a certain "Status" is being retrieved from the database.

```
<div class="card card--<?php echo $row['StatusDescription']?>">
<svg class="card__status-icon"xmlns="http://www.w3.org/2000/svg" x="0px"
```

```
y="0px"
width="50" height="50"
viewBox="0 0 192 192"
style=" fill:#000000;"><g fill="none" fill-rule="nonzero" stroke="none"
stroke-width="1" stroke-linecap="butt" stroke-linejoin="miter" stroke-
miterlimit="10" stroke-dasharray="" stroke-dashoffset="0" font-family="none"
font-weight="none" font-size="none" text-anchor="none" style="mix-blend-mode:
normal"><path d="M0,192v-192h192v192z" fill="none"></path><g fill="#ffffff"><g
id="surface1">
    <?php if($status == "Holding"){?><path d="…"></path>
    <?php }else if($status == "Prioritized"){?><path d="…"></path>
    <?php }else if($status == "Finished"){?><path d="…"></path>
    <?php }else if($status == "Started"){?><path d="…"></path><?php }?>
</g></g></g>
</svg>
```

Figure 18 – Code for Status Icon

The code in Figure 26 starts by setting a common "style" for all the SVGs and then proceeds to check for the contents of $status (which in this case stores the "StatusDescription" of each row). Based on the contents, the if statement will assign the path to the correct SVG and then display it (the path was replaced by "…" due to the size of it).
A CSS trick was used to colour the corner and the tutorial was found on the CSS-Tricks website (Coyier, 2019) The way it works is that if there is a transparent box with a border but no height or width, all the borders will connect and create an arrow for each direction. For this implementation, everything besides the top border were set to transparent to create a down-facing arrow. The arrow was placed on the top right corner of the card and then the excess part that was outside of the card was hidden and, in the end, created a colourful corner. The colours are assigned with CSS by using the class that was assigned to the div before the SVG.
The code is:

```
<div class="card card--<?php echo $row['StatusDescription']?>">
```

Figure 19 – Card Status Colour Class

And refers to the custom.css file on the lines:

```
.card--Holding::before {
    border-top-color: #b91212;
}

.card--Started::before {
    border-top-color: #070e66;
}

.card--Prioritized::before {
    border-top-color: #f39c12;
}

.card--Finished::before {
    border-top-color: #2ecc71;
}
```

Figure 20 – Card Status Colour CSS

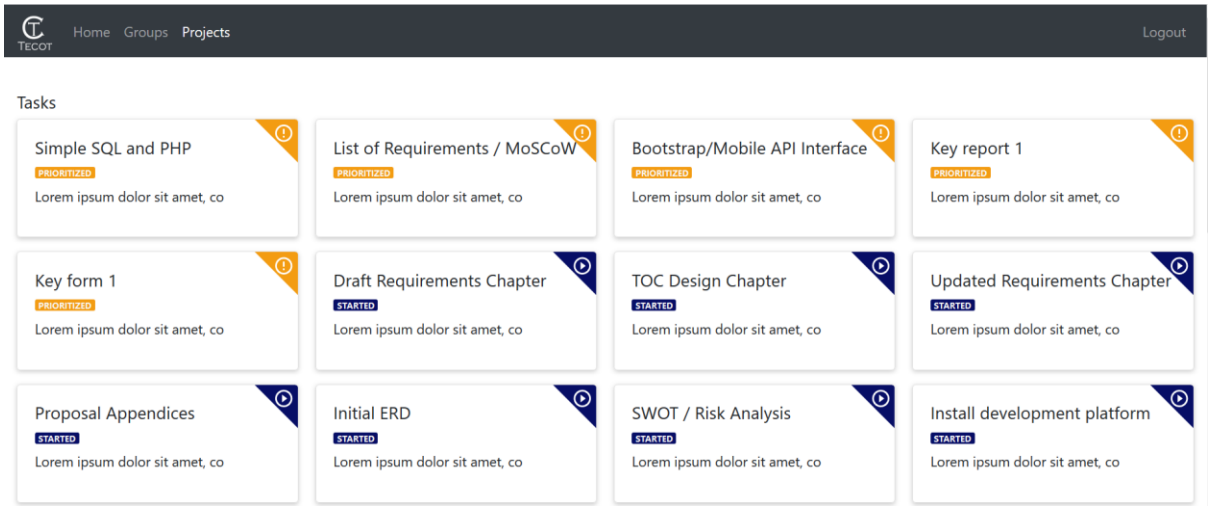

Figure 21 - Task card in prioritised order

Clicking on one of the cards will redirect the user to a page that allows them to edit the information such as the Title, the Description and the Status. After confirming the changes, the user will simply get redirected back the Tasks page and the card will be updated with the new information.

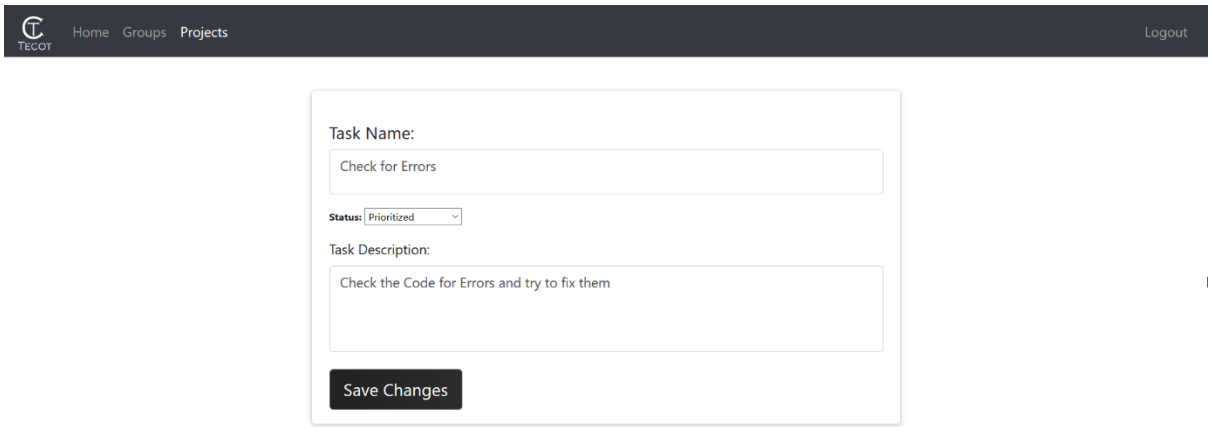

Figure 22 - Edit Task Page

5.5.2 Navbar

The navbar is a functionality that will be used throughout the entire project but will have different functionality for each user type.

Figure 22 – Supervisor's Navbar

Figure 23 – Student's Navbar

Currently, two different versions were implemented. One for the supervisor and one for the student. They might look the same, but they function in different ways since the options

redirect to different pages (e.g. Projects: For the supervisor it leads to the projects that they are in charge of and for the student, it redirects them to the projects they are part of).
On the far-right, the Logout functions as it should be expected, and it will logout and redirect the user back to the login page.
And on the left side, the logo is located right next to the home button. Both of those are expected to function the same way which means that one of them can be removed to save space.
The most challenging part about the implementation of the navbar was the inclusion of a visual function that indicates the tab that the user is currently in. This was done with a script that retrieves the title of the page, stores it in a variable as a class by combining it with the word "#nav-" and then proceeds to run a function that removes the "active" class from all the items that contain the class "navbar li" and then adds the class only to the one that was saved inside the variable. The code can be seen right underneath for reference.

```
    $(function() {
        $('.navbar li').removeClass('active');
        $(navid).addClass("active");
    });
```

Figure 24 – Change Active Element Code

5.5.3 Expandable Cards

One of the more commonly used HTML/CSS tricks during the implementation was the expandable cards. They were used in order to keep the page clean, without a lot of clutter and overall easy to navigate. It is one of the tricks that comes with Bootstrap. All that needs to be done is set an item to "hide" and connect it with a button that will "reveal" it when clicked. Even though it sounds simple, some issues occurred during the implementation and they were the result of how the cards were generated. In order to create them, a loop was running that would create as many cards as items that were retrieved from the SQL query. Setting the same ID would not have worked because by clicking one button, all the sections would expand and not just the one that was clicked. To overcome that issue, each pair of items was assigned the same ID from the data that was retrieved. That ID would be unique to every row of the results and there is no way for any other button to interfere. It was achieved using this code:

```
<a href="#" data-toggle="collapse" data-target="#collapseExample-<?php echo
$row['TeamID']; ?>" aria-expanded="false" aria-controls="collapseExample-<?php
echo $row['ProjectID']; ?>" class="card-link btn">&plus; Show More</a>
            <div class="collapse" id="collapseExample-<?php echo
$row['ModuleId']; ?>">
                <?php echo "Module: ". $row["ModuleName"]. "<br>". "Group ID:
". $row["TeamID"]. "<br>"; ?>
```

Figure 24 – SQL used for unique ID

The biggest challenge was making sure that the button would change from "+Show More" to "-Show Less" when clicked. A script was used for that purpose, one that would add an event listener which checks the classList of the item that was clicked. If the classList contained the class "card-link", it would run the function that changes "aria-expanded" and the innerHTML based on whether the "aria-expanded" of the object was set as true or false.

```
window.addEventListener("click", function (e) {
    var targetClass = e.target.classList;
```

```
    if (targetClass.contains("card-link")) {
        var ariaExpand = e.target.getAttribute("aria-expanded");
        if (ariaExpand === "false") {
            e.target.innerHTML = "&minus; Show Less";
            e.target.setAttribute("aria-expanded", "true");
        } else {
            e.target.innerHTML = "&plus; Show More";
            e.target.setAttribute("aria-expanded", "false");
        }
    }
});
```

Figure 25 – Javascript behind the "Show More" function

5.6 Conclusion

The Implementation chapter focused on the main choices and the issues the came with them, along with how they were overcome. It attempts to talk about the more complicated choices and provide clear instructions that can help the reader further understand them.

6 Validation Chapter

6.1 Introduction

In this chapter, the focus is going to be on the steps that were taken or could be taken to ensure that the main requirements for the project have been implemented and functioning as expected, making sure that there is a minimum number of bugs and overall issues. The testing will be performed by taking the requirements within the Requirements chapter into consideration. Validating and testing the project is a very important step during the development phase and making sure that it is performed properly will result in a better experience for the user when the product is finally working and ready to be released.

6.2 Test Tables

Creating test tables that are associated to the functional and non-functional requirements will help with the validation process. The tables will have boxes that are linked to the different requirements, the expected results and the current status of the implementation at this stage of the development. The full list of scenarios within the test tables can be found in Appendix M and the Functional and Non-Functional requirements can be found in Appendices F and G and can be used for reference.

A few examples of the validation process will be documented below.

ID	FR	SCENARIO	EXPECTED RESULT	RESULT
7	9	Student creates a new task for a project.	System creates the task and displays it on project/task page. Notifies users related to the project via email.	Success
10	12	Student edits a task.	System displays the new version of the task. System notifies relevant students.	Success
12	14	Student changes the task status.	System replaces the old status with the new one and notifies relevant users.	Success

Figure 28 – Part of the student's test tables

Creating new tasks for a project that the user is part of is one of the key functionalities of the tool and therefore it is important to make sure that it functions properly. Each Project will have a list of tasks unique to the team and the project.

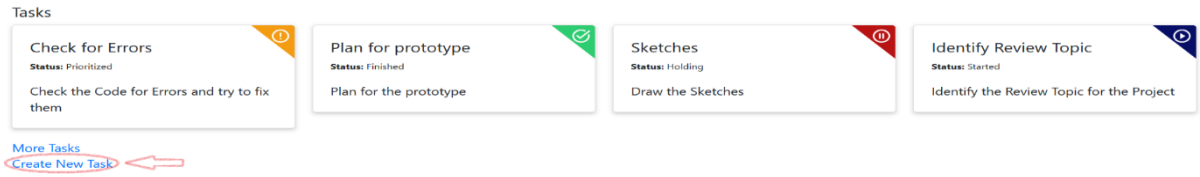

Figure 26 – Create New Task Option

The option to create a new task exists both on the main project page but also in the "More Tasks" page of the project. Clicking on it redirects the user to the task creation page in which they will input the information they desire.

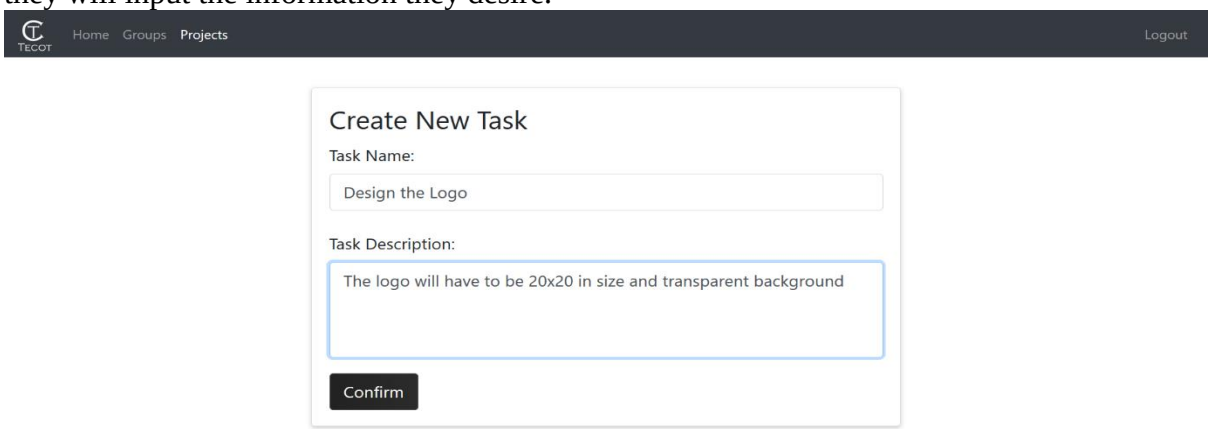

Figure 27 – Create New Task

After confirming, the user should get redirected back to the Project's page in which they can see the 4 newest tasks. The new task has been added and is first on the list of the latest tasks and its status was set by default as "Holding" until a group member reviews it and changes the status.

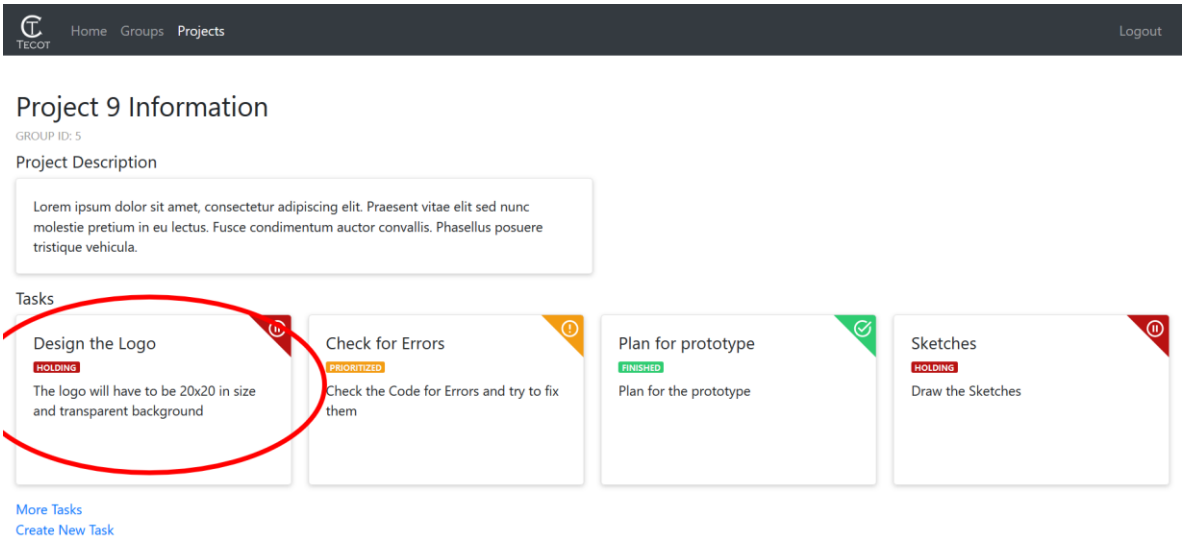

Figure 29 – New Task as "Holding"

Currently the status can be changed in the “Edit Task” page. All the user must do is click on the dropdown menu and select one of the options. Currently there are 4 different statuses, but it is possible to add more in the future.

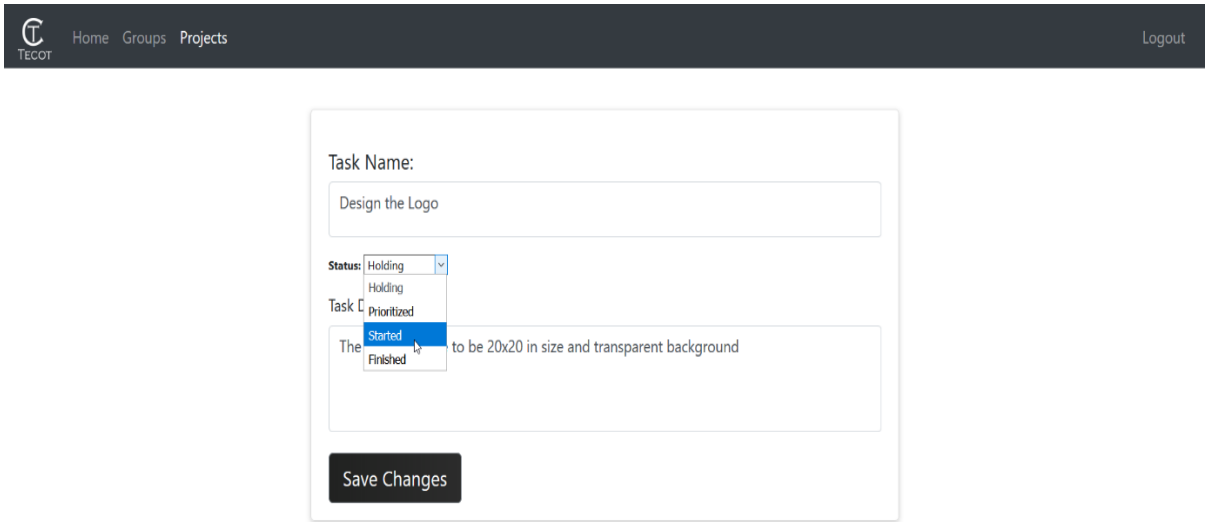

Figure 30 – Edit Task

After saving the changes, the user gets redirected back to the “Project Information” page and the task can now be seen with the update information.

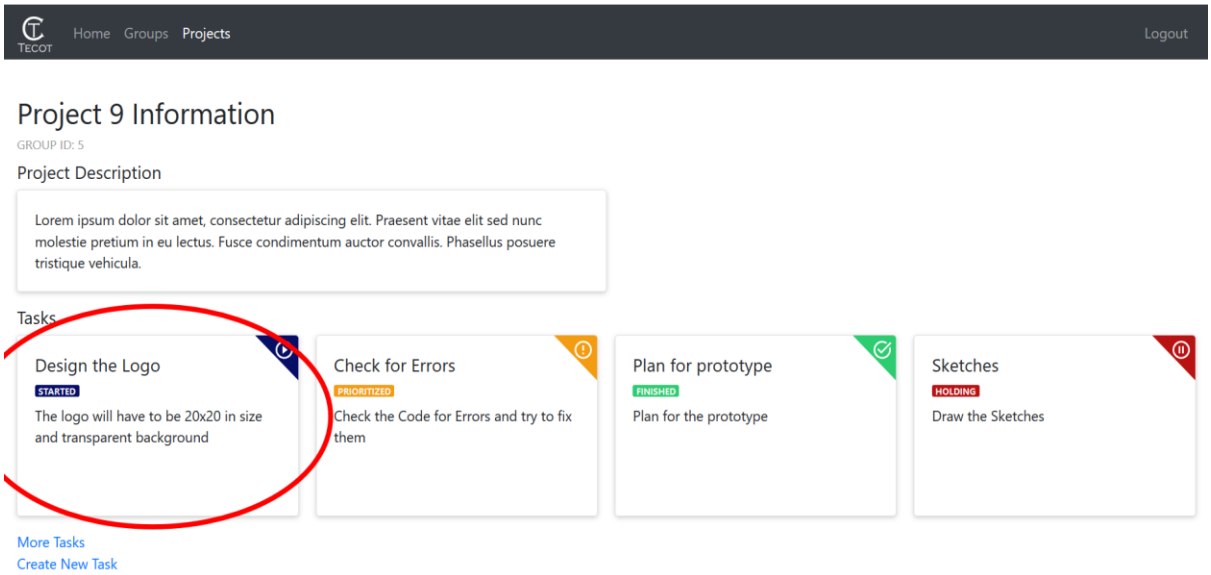

Figure 31 – New Task as “Starting”

ID	FR	SCENARIO	EXPECTED RESULT	RESULT
15	20	Supervisor creates a new activity.	System adds the activity to the projects page	Success
16	21	Supervisor deletes an activity.	System removes the activity from the projects page and system/database.	Not Implemented
17	22	Supervisor sets an activity goal.	System displays the goal in the activity details.	Success

18	23	Supervisor edits an activity goal.	System replaces the old goal with the new one and notifies relevant students.	Not Implemented
19	24	Supervisor adds a deadline to a project.	System displays the deadline on the related Project page.	Success
20	25, 26	Supervisor edits a deadline of a project.	System replaces the old deadline with the updated version and notifies relevant students.	Not Implemented
21	28	Supervisor creates a new group.	System (notifies relevant students)? through email. New group is displayed on groups page. Display "New group/s successfully created" message.	Success
22	30	Supervisor assign student to group	System adds the new selected user to the group. Displays the user in the Group Members section and shows up in the user's groups.	Success

Table 9 – Supervisor scenarios

With the help of the test table of figure 33, another functionality can be validated that focuses on the creation of a new project, setting a deadline, creating a group for it and then assigning a user to that group.
The steps that will be followed are similar to the previous validation. Interacting with the system in a step-by-step approach while taking notes and handling any possible errors.
The first screen for this scenario is going to be the page that lists all the Projects from a supervisor's module.

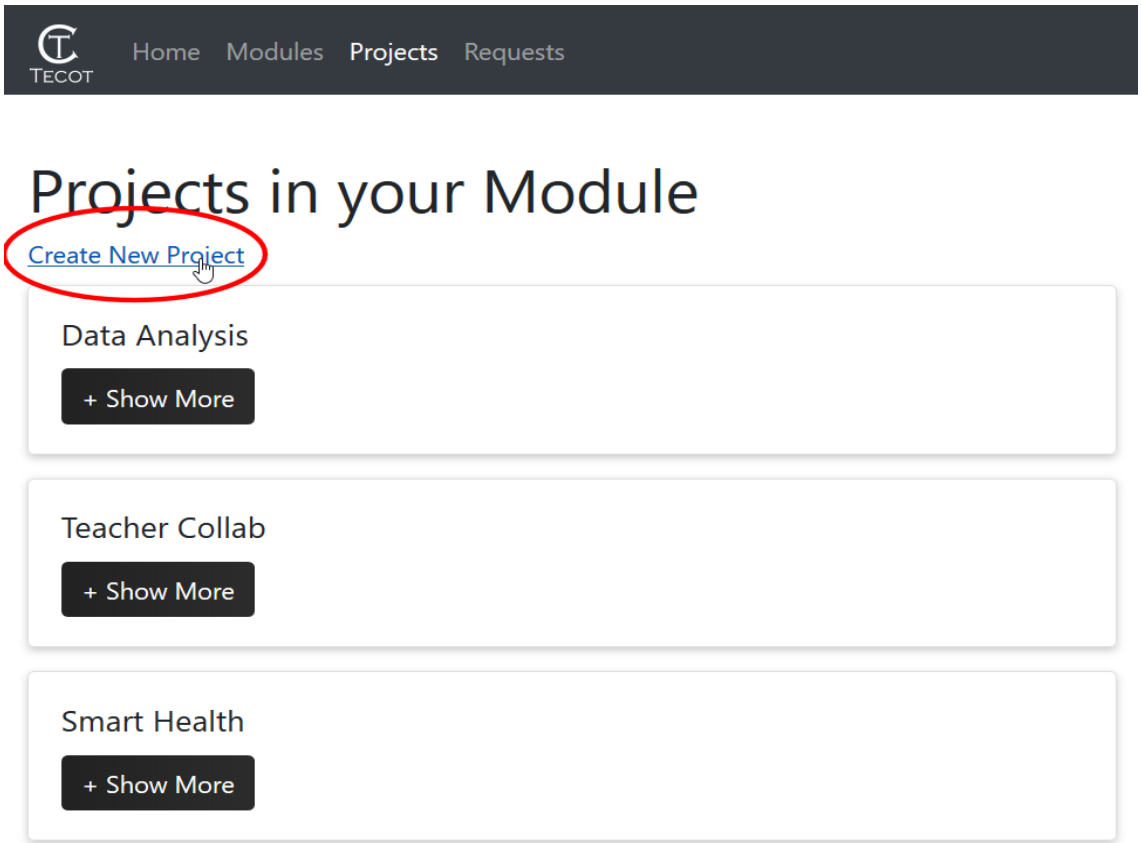

Figure 32 – Projects in Module page

At the top of the page, there is a “Create New Project” button that when clicked redirects the user to a page which consists of 3 forms (1. Project Title, 2. Project Description, 3. Deadline).

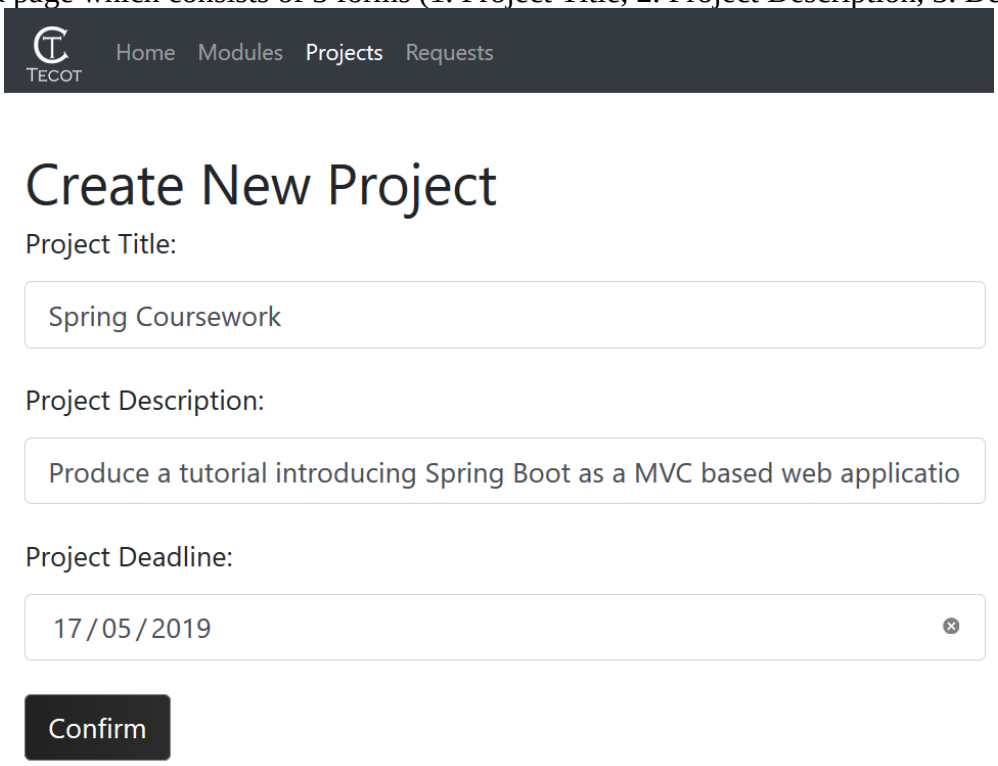

Figure 33 – Create New Project

When the user confirms that this is the correct info, they will be redirected back to the projects page and the newly created project will be placed above the previously created projects for ease of access and it also validates the “Supervisor creates a new activity” scenario. Clicking on the “Show More” button will reveal all the details(which confirms that the scenarios with ID of 17 and 19 were successfully implemented) and the ability to go to the next page that will display all the groups assigned that were created for the project.

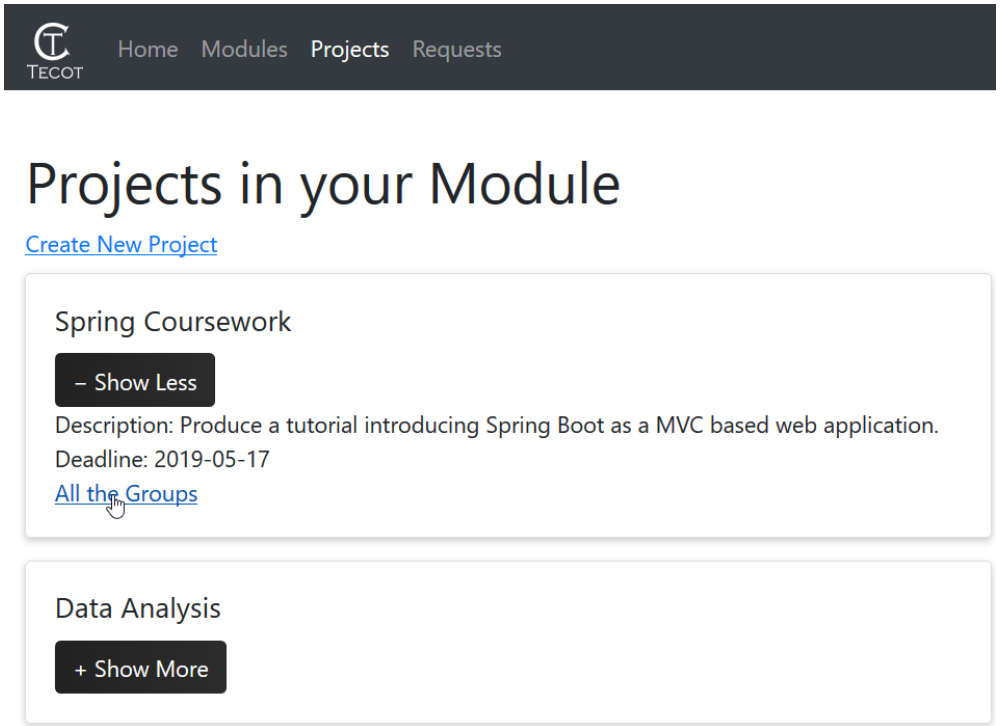

Figure 34– New Project

The groups page is currently empty because the new project was just created and in the current state of the tool, the user has to manually create the groups.

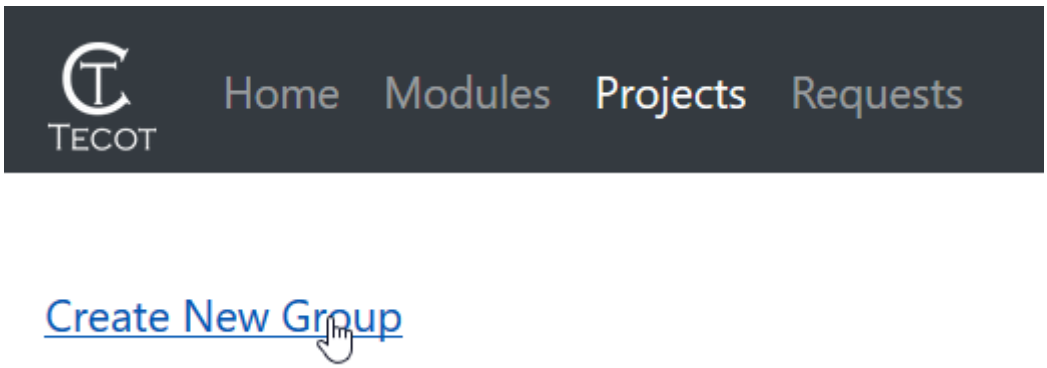

Figure 35 – Create New Group Option

Currently, the group creation only asks for the user to provide a group name. One idea is to allow the user to populate the group with a set number of random users which is a function that has not been implemented yet.

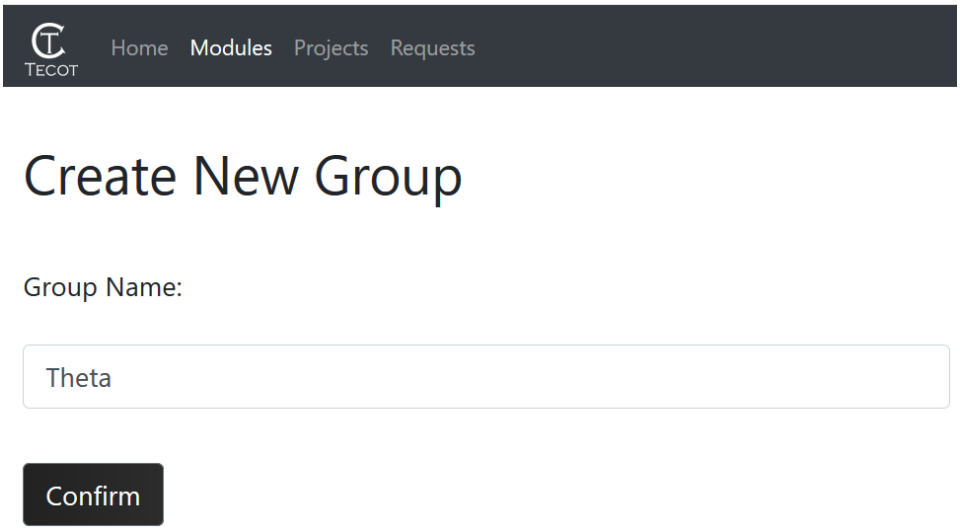

Figure 36 – Create New Group Page

After confirming the creation of the new group, the supervisor is then redirected to the groups page in which they can see the newly created group. This validates the “Supervisor creates a new group” scenario.

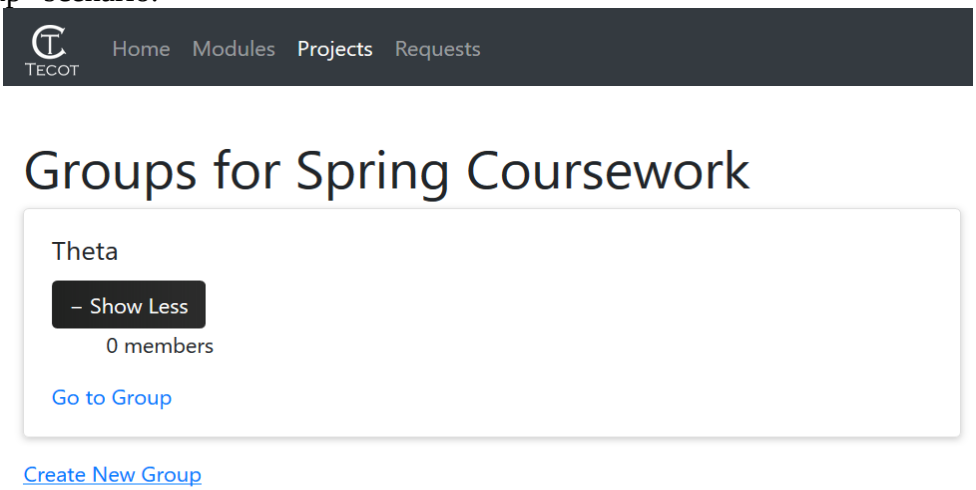

Figure 37 – Validation of Group Creation

The group page currently houses three different cards. The first has a list of the members, the second displays the current progress of the group (Through tasks and submissions) and the third contains the more sensitive options (Add, Remove Member and Delete Group). The supervisor will have to click on the “Show More” option on the third card and then click on “Add Member” if they wish to add a new member to the group.

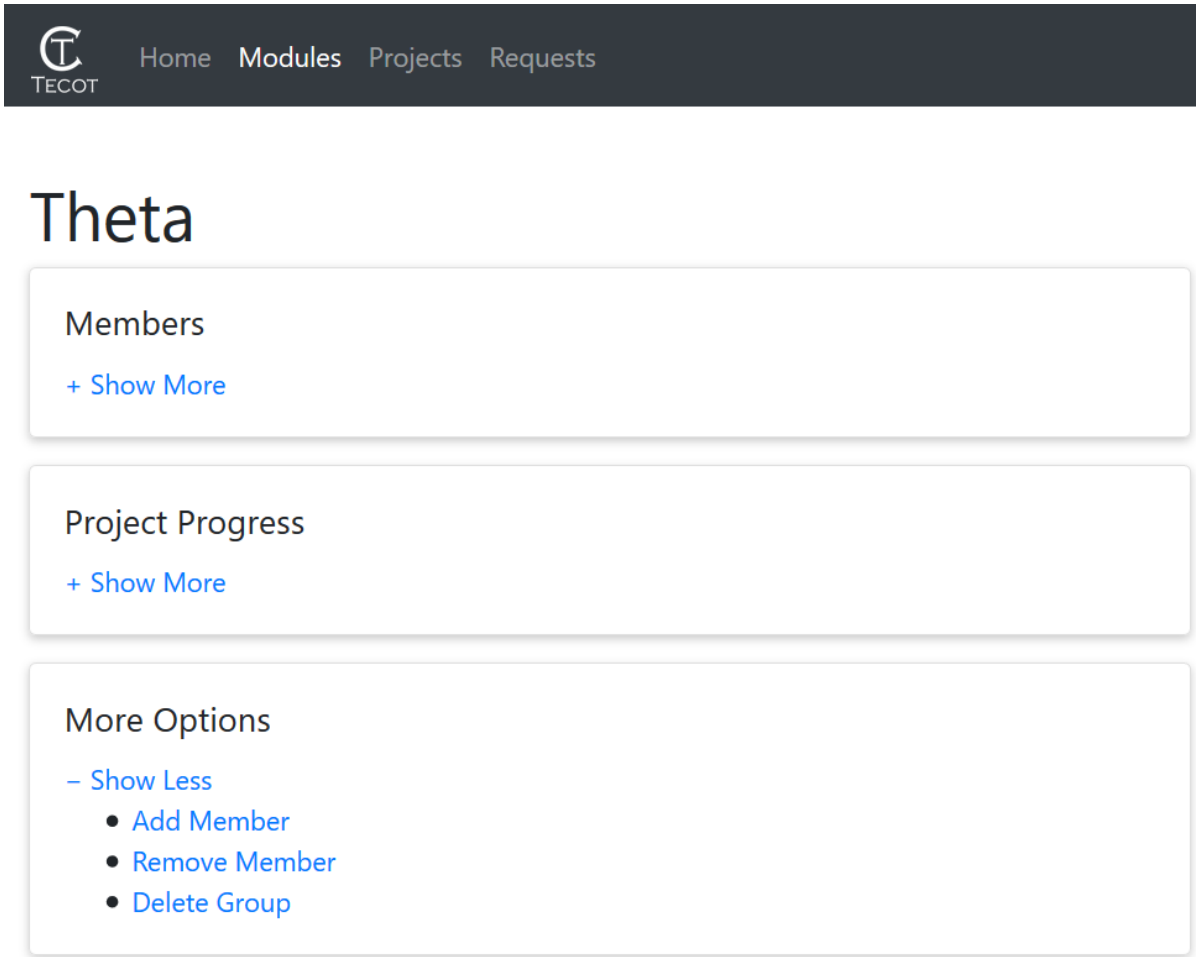

Figure 38 – Group Page

The Add Member page is currently very basic design-wise. One of the future plans is to allow for the user to add multiple members at once but currently they are greeted by a page with a form that when clicked displays all the relevant users and works as a search form to easily traverse through the hundreds if not thousands of users.

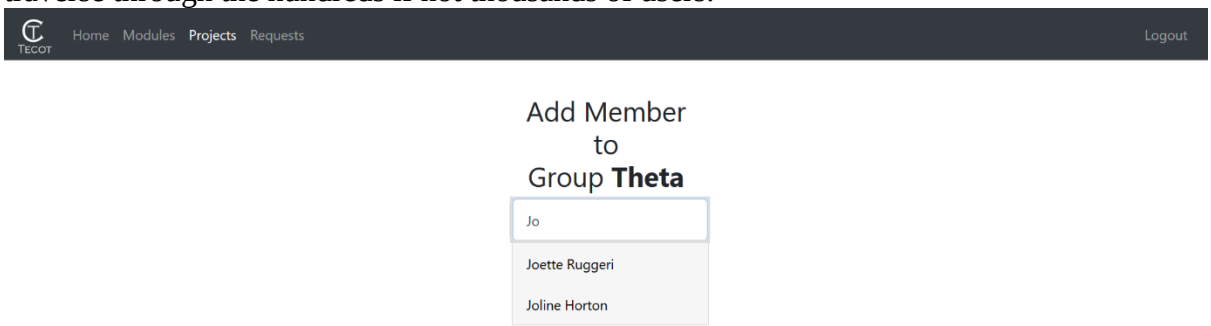

Figure 39 – Add Member Page

By simply clicking on the name they wish to add, they will be redirected back to the group page and inside the members card, they can now see the new member inside the list. This validates the “Supervisor assign student to group” scenario.

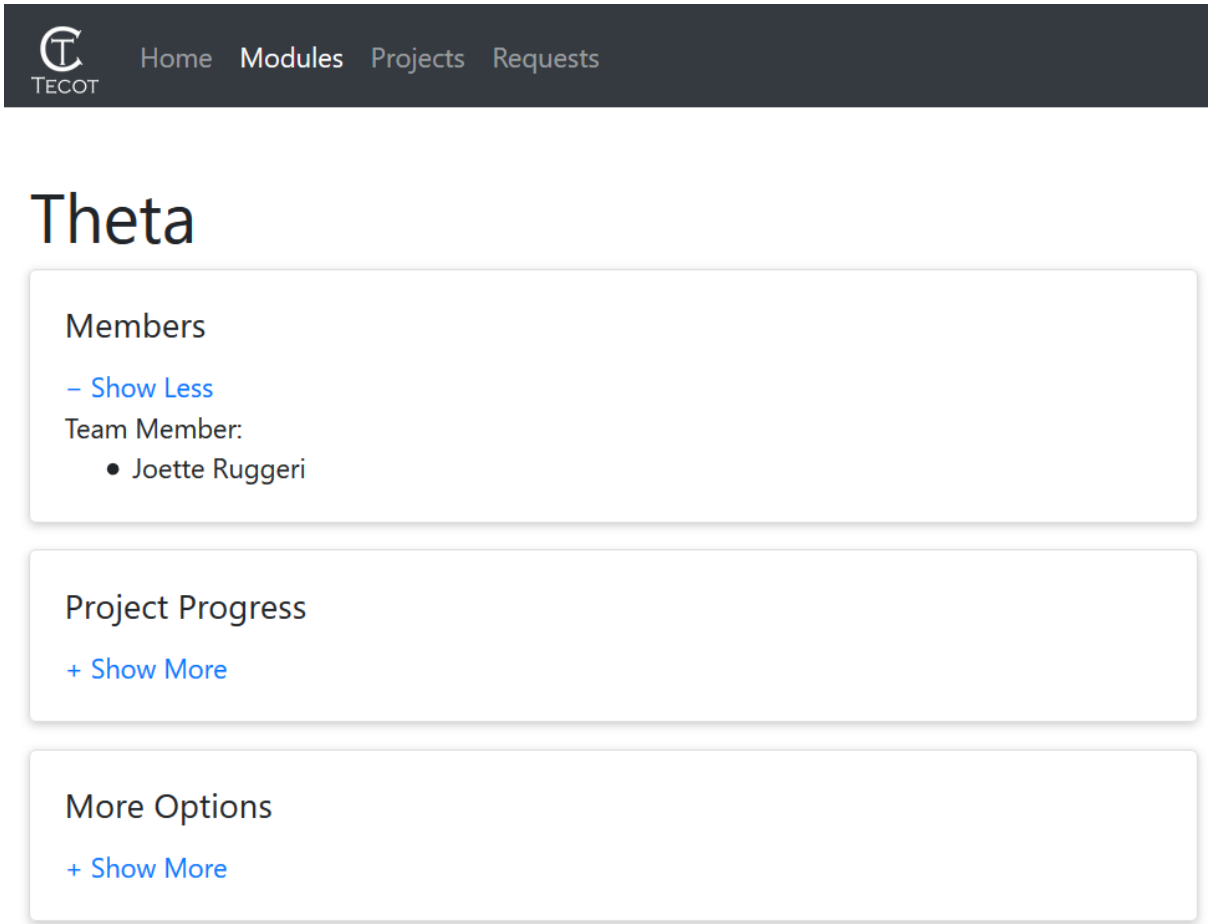

Figure 40 – New Member Validation

6.3 Conclusion

The validation process will get repeated for every new functionality or even minor change that happens during the development process. This is not the final product since everything could change at any given moment but is a very good indication that certain parts are working as expected and will allow for the development to move to the next implementation.

7 Critical Review

7.1 Introduction

The final part of the report aims to summarise on the overall achievements, the main problems that were encountered during the development and all the possible future plans and goals for the project to improve.

7.2 Achievements

This project started development slowly and with plenty of issues that will be thoroughly mentioned in the "Problems Encountered and Limitations" section. However, near the end, it all came together into a tool that will be able to set the foundation for a new Team Collaboration tool. Most of the functionalities were implemented, and the product looked clean and worked as expected.

It started as an idea the developed into an actual plan after several discussions with the client. The first step after the topic was decided was the basic design of what the tool would look like using sketches and wireframes. This part resulted in the idea that the tool should be accessible to every type of user with a minimalistic and clean design.

Identifying all the users and the requirements, something crucial to the development cycle.

Starting the development process, picking up tools that were accessible at the time (which also resulted in limitations) but also making sure that they fit the purpose. Creating a database that stores a plethora of data that was used to test the capabilities of the system and the interaction with the user.
The implementation chapter was the most challenging but also the most exciting chapter with new things to learn and work with. This was also the part in which, even though, things were going smoothly, it did seem that there were changes in order and certain things should be approached in different ways.
Finally, validating the product resulted in a sense of accomplishment. After the error handling, patching up issues and testing steps multiple times, there was nothing like changing the current state of a scenario from "Not Implemented" to "Success".
There are more things that should be considered for this project and there will be a discussion in the "Future Plans" section.

7.3 Problems Encountered and Limitations

This project was a new experience of this scale. The starting tools that were used during development were picked mainly due to the ease of access. Those decisions limited the possibilities and forced the project down a more linear path. At first, the project was developed using PHP/HTML (since the local hosting program "XAMPP" came with everything required built in) but after some research, the intention was to switch to something more modern which was a bit challenging since the tools were limiting the choices. Thankfully, that did not hinder it greatly, instead, it might have caused some issues that can be resolved at a later stage.
Another limitation was the thought process behind the log-in and overall user system. The tool was visualised as a Team Collaboration Tool for students to use and interact with their Supervisors/Lecturers. In other words, it was intended to share a user database with the university and apply the same "UserID" filters when logging in to decide whether the user is a student or someone else.
The biggest limitation of all was the time limit. Functionalities that could have been included were not in order to conserve some time and focus on the more crucial parts of the tool and making sure those are polished and functioning as intended. This limit also resulted in some more minor pages not being as efficient as they should be and not as visually pleasing as well (e.g. Adding New Members to a group).
Finally, the limitation of knowledge. The project was a new experience in which new tools had to be used and new approaches had to be known. This resulted in a big problem near the end of the development when putting things together. Certain pages and functionalities were developed earlier with less experience and overall knowledge and when combined with newer pages, the result was as it should be. This led to revisiting and reworking the code until satisfied, but it also means that some parts might still be written in an older, less efficient way.

7.4 Future Plans

This section aims to talk about things that did not get implemented mainly due to the limitations mentioned above.

1. The inclusion of user's homepages that return a list of news or updates.
2. The addition of an administrator role that will be given access to certain functionalities that the current roles should not have. The current idea was that the supervisor/lecturer should have access to them, but it is also unrealistic to expect a lecturer to act as an administrator to the entire system on the side.

3. Simplify certain actions, minimise the number of steps or pages involved in an action. (e.g. A user should be able to create multiple new groups, populate them with random members or add multiple members at once without having to go back and forth multiple times).
4. Populate pages. The intention was the pages to be minimalistic and clear. However, this resulted in an excess of white space that should be filled with other information but also remain responsive.
5. Editing data. Outside of a user's group page, most of the data (such as Project Titles, Descriptions etc) were not given an edit functionality.
6. Fully implement a log in system that uses the user's ID to determine their role and redirect them accordingly.
7. Add an email notification system that will alert users when changes are being made.

Most of those plans were not left out due to the difficulty of the task but because of the time constraint. They were some of the ideas that were considered during the development but were not fully implemented.

7.5 Legal, Ethical, Societal and Security Issues

Security is the biggest possible issue with this project. A person that could somehow access all the data would be able to alter or even delete certain things from the database. That could include, the whole groupwork, certain tasks and even groups. A lot of resources that talk about the best practises regarding the security of your database. The most important step is to properly encrypt the data and only give full access to very few and reliable staff. The website should also follow the standard account security procedures which includes steps like enforcing users to set strong passwords, then those passwords should be stored encrypted and salted. Another step would be to lock the account after a certain amount of failed log in attempts. A firewall protection should be set up on top of all the above steps to improve the overall security. Lastly, if all else fails, it is recommended that any sort of confidential information that is not required should not be stored within the database to reduce the value and therefore the risks. SQL injections is a common web hacking technique which allows the hacker to access or use data from within the database by placing malicious code in SQL statements via web page input. To reduce that risk, we must use parametrized queries to have more control over the user inputs.
One of the ethical and legal issues that should also be considered is the General Data Protection Regulation (GDPR) adds an extra layer of reassurance for a user that will entrust his personal data to an organisation by making sure that there are people responsible for handling the data and also that all the procedures to ensure that everything is secure are followed. Users should know what sort of personal information is going to be used and why.

7.6 Conclusion

This project was the biggest challenge this year. However, challenges also come with new knowledge and due to the scale of this project, the knowledge piled up and a lot of new tools and techniques were discovered. If there was a chance of completely redoing the project with the knowledge that was accumulated, it would all be done in a very different and hopefully better way. However, knowledge does and will carry on to the next project and then the project after that.

References

Lipinski, M. 2011, "Database security", *SC Magazine*, vol. 22, no. 11, pp. 50.

Bamrara, A. "Evaluating Database Security and Cyber Attacks: A Relational Approach." Journal of Internet Banking and Commerce 20.2 (2015): 1-17. Web.

Moon C.S., Chung S., Endicott-Popovsky B. (2014) A Cloud and In-Memory Based Two-Tier Architecture of a Database Protection System from Insider Attacks. In: Kim Y., Lee H., Perrig A. (eds) Information Security Applications. WISA 2013. Lecture Notes in Computer Science, vol 8267. Springer, Cham

Highland, H., 1996. Database security. Computers & Security, 15(4), pp.274–275.

Greengard, S., 2018. Weighing the Impact of GDPR. Association for Computing Machinery. Communications of the ACM, 61(11), p.16.

European Commission - European Commission. (2019). *Data protection in the EU*. [online] Available at: https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en [Accessed 20 Apr. 2019].

Fisher, J. (2019). *How to Adopt the GDPR Risk Based Approach*. [online] Linkedin.com. Available at: https://www.linkedin.com/pulse/how-adopt-gdpr-risk-based-approach-james-fisher [Accessed 20 Apr. 2019].

Random-name-generator.info. (2019). *Random name generator*. [online] Available at: http://random-name-generator.info/ [Accessed 18 Feb. 2019].

Le.ac.uk. (2019). *Courses — University of Leicester*. [online] Available at: https://le.ac.uk/courses [Accessed 17 Feb. 2019].

Coyier, C. (2019). *CSS Triangle | CSS-Tricks*. [online] CSS-Tricks. Available at: https://css-tricks.com/snippets/css/css-triangle/ [Accessed 16 Apr. 2019].

Appendix A -Schedule

Day	*Date*	*Deliverable*	
Fri	28/09/2018	Documentation	Proposal Appendices
Fri	05/10/2018	Methodology	Initial ERD
Fri	05/10/2018	Research	Portfolio of Past Dissertations
Fri	05/10/2018	Documentation	Complete draft proposal
Fri	05/10/2018	Methodology	SWOT / Risk Analysis
Fri	**12/10/2018**	**Documentation**	**Project Proposal**
Fri	12/10/2018	Implementation	Install development platform
Fri	19/10/2018	Methodology	Stakeholder analysis
Fri	19/10/2018	Documentation	TOC Requirements Chapter
Fri	19/10/2018	Implementation	Establish PHP connection to database
Fri	26/10/2018	Research	Volume of testing data required
Fri	26/10/2018	Methodology	Use Cases and User Stories
Fri	26/10/2018	Implementation	Simple SQL and PHP report
Fri	02/11/2018	Methodology	List of Requirements / MoSCoW
Fri	02/11/2018	Research	Identify Review Topic
Fri	02/11/2018	Implementation	Simple Form with SQL insert
Fri	09/11/2018	Documentation	Draft Requirements Chapter
Fri	09/11/2018	Research	Bootstrap/Mobile API Interface
Fri	09/11/2018	Implementation	Inner Join report
Fri	16/11/2018	Methodology	Sketches
Fri	16/11/2018	Documentation	TOC Design Chapter
Fri	16/11/2018	Research	Identify SQL queries from requirements
Fri	23/11/2018	Implementation	Implement mockups of key webpages
Fri	23/11/2018	Methodology	Sequence diagrams
Fri	23/11/2018	Research	Plan for prototype demonstration
Fri	30/11/2018	Documentation	Updated Requirements Chapter
Fri	30/11/2018	Implementation	Key report 1
Fri	30/11/2018	Methodology	Revised ERD and Data Dictionary
Fri	07/12/2018	Implementation	Key form 1
Fri	07/12/2018	Research	Test Strategy
Fri	14/12/2018	Documentation	Draft Design Chapter
Fri	**14/12/2018**	**Presentation**	**Early Prototype demonstration**
Fri	11/01/2019	Methodology	Wireframes/Navigation
Fri	11/01/2019	Research	Key References and Mind Map
Fri	11/01/2019	Implementation	Key report 2
Fri	18/01/2019	Documentation	Updated Design Chapter
Fri	18/01/2019	Research	Data generation plan
Fri	18/01/2019	Implementation	Key form 2
Fri	25/01/2019	Research	Identify most complex SQL
Fri	25/01/2019	Implementation	Key form 3

Fri	25/01/2019	Implementation	Key report 3
Fri	25/01/2019	Research	Identify viva story
Fri	01/02/2019	Documentation	Draft Review Chapter
Fri	01/02/2019	Implementation	Key report 4
Fri	01/02/2019	Implementation	Key form 4
Fri	01/02/2019	Evidence	Populate database with required data
Fri	08/02/2019	Documentation	TOC Implementation Chapter
Fri	08/02/2019	Documentation	Technologies and Architecture Section
Fri	08/02/2019	Implementation	Key report 5
Fri	08/02/2019	Implementation	Key form 5
Fri	08/02/2019	Evidence	Test Tables
Fri	15/02/2019	Documentation	Second Major Section
Fri	15/02/2019	Research	Identify viva-oriented functionality
Fri	15/02/2019	Implementation	Key report 6
Fri	15/02/2019	Implementation	Key form 6
Fri	22/02/2019	Documentation	Updated Review Chapter
Fri	22/02/2019	Research	Viva plan
Fri	01/03/2019	Evidence	Simple reports utilising big data
Fri	08/03/2019	Documentation	Draft Implementation Chapter
Fri	08/03/2019	Implementation	Dashboard 1
Fri	15/03/2019	Evidence	Inner joins utilising big data
Fri	22/03/2019	Documentation	Draft Validation Chapter
Fri	29/03/2019	Implementation	Dashboard 2
Fri	05/04/2019	Documentation	Draft Introduction and Conclusion
Fri	12/04/2019	Documentation	Appendices and references
Fri	12/04/2019	Implementation	Dashboard 3
Fri	19/04/2019	Documentation	Complete draft report (submitted)
Mon	22/04/2019	Presentation	Practice Viva
Fri	26/04/2019	Implementation	Last functionality implemented
Mon	**29/04/2019**	**Documentation**	**Report Submission**

Table 10 – Deliverable List

Appendix B – Initial ERD

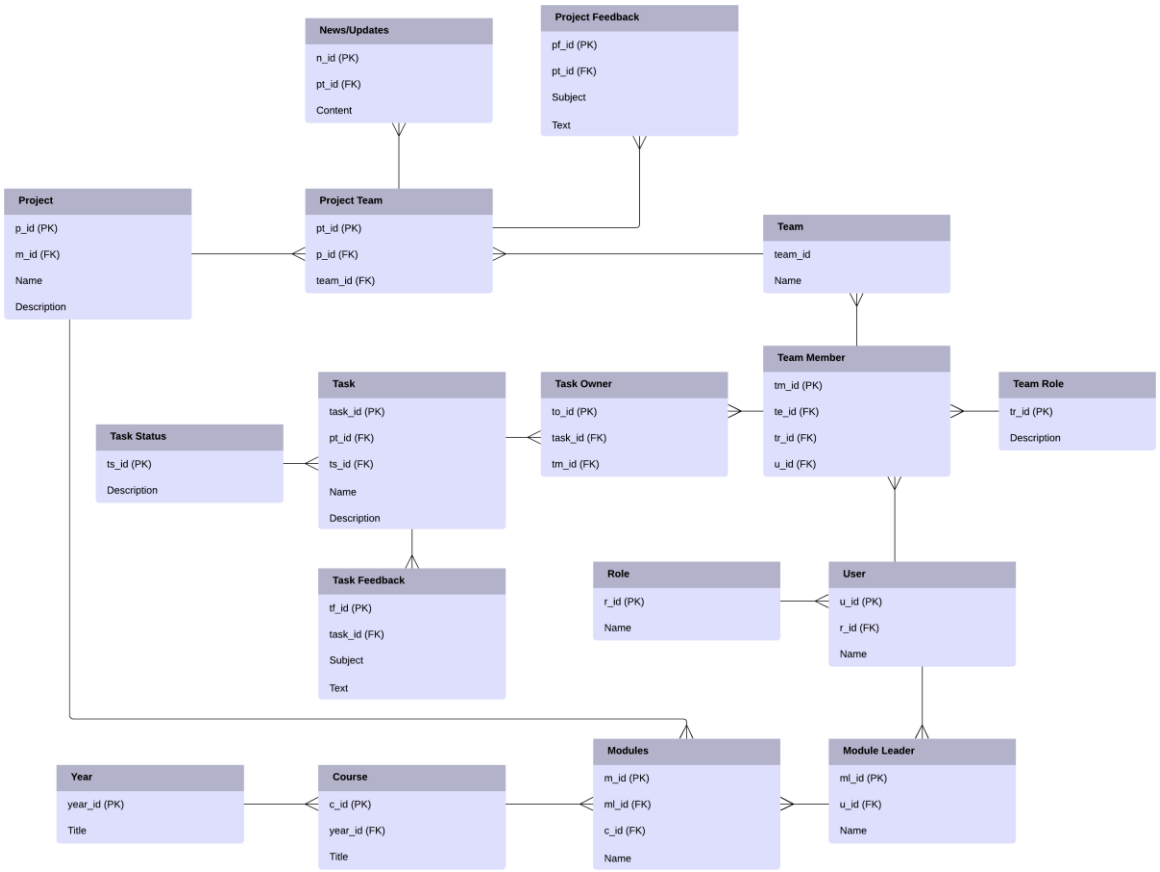

Figure 41 – ERD

Appendix C – Volume of Testing Data Expected

Kingston university consists of approximately 19,000 students and 1,000 members of staff which means that the total of "Users" is expected to be around 20,000. A team will always consist of at least 2 different students and therefore there would be 10,000 different "Teams". However, if a student might have to join a team for 3 out of his 4 modules, that increases then number of unique "Teams" up to 30,000. Each team will be working on 2 different Projects per module on average which leads to a total of 60,000 unique "ProjectTeam". A "TeamMember" is going to display the unique members in every team. Since every team will consist of an average of 4 members, the amount of values that "TeamMember" will need to go through in the database is will be around 240,000. Also, each "ProjectTeam" will consist of its own unique tasks. Depending on the size of the project, that can range from 1-10 "Tasks" (600,000 "Tasks" in total) per "ProjectTeam". If every user has at least one task assigned to them that will result to 600,000 "TaskOwners". There is also the consideration of the "ProjectFeedback" and "TaskFeedback". A project is not expected to receive feedback more than 10 times since the focus during the building phase will be on the "TaskFeedback". If each project gets feedback 10 times and each individual task 3, the expected size of "ProjectFeedback" is 600,000 and "TaskFeedback" 900,000. Each user will be receiving "News/Updates" about the projects that he is related to. If a project produces 30 "News/Updates", the total expected amount will be 1,800,000.

After a brief google search (Students in Kingston University) we can see that the university consists of around 20.000 "Users" (staff and students).

We should also consider that a team will consist of at least 2 people and therefore there would be 10,000 different "Teams".

However, since a student could be part of many teams (up to 3 different teams most likely) that is 30,000 "Teams" to consider.

Each team will be working on multiple projects but let's assume that throughout the year they have time to work on 2 different projects at best. That is a total of 60,000 "ProjectTeam".

If each "ProjectTeam" has 4 "TeamMembers" in it, then the volume of data that we will have to test on "TeamMembers" is going to be around 240,000

Each "ProjectTeam" will have its own set task and it is safe to assume that each project will have around 5 tasks which will give us a total of 300,000 "Tasks" and that also translates to 300,000 "TaskOwners" if every owner had at least one task and each task had one owner.

A "ProjectTeam" will receive feedback from a lecturer (possibly no more than 10 times). That means we should test with 600,000 "ProjectFeedback"

If every task received up to 3 different feedbacks until the submission, that would mean that we need space for 900,000 "TaskFeedback".

All the major changes and feedback in a "ProjectTeam" will result in "News/Updates". If each "ProjectTeam" produces 30 of them, this results to 1,800,000 "News/Updates" to work with.

A task will have no more than 5 different "TaskStatus" choices.

We will also assume that there are 100 "Courses" and 500 "Modules" in total. And therefore 500 "ModuleLeader".

For "Role" we will not need more than 3-4 different roles.

Appendix E – Use Cases

Use case number	UC1
Name	Create Groups
Description	Lecturer creates new groups for the new project that they assigned in the lecture
Actor	Lecturer
Pre-Conditions	Lecturer is logged in the System.
Basic Flow	3. Click on “Create new groups” 4. Set Group information 2.1 Select Year 2.2 Select Module 3. Select number of Groups 4. Click “Done” 5. System creates the group 6. System sends an invite notification to the selected students
Extensions	
Post Conditions	Groups created Members assigned

Table 11 – UC 1

User case number	UC 2
Name	Leave Feedback
Description	Lecturer wants to leave feedback on one of the projects
Actor	Lecturer
Pre-Conditions	Lecturer is logged into the system. Activity and a group have been created
Basic Flow	1. Click on “Projects” 2. Select Year 3. Search for the project 4. Select the project 5. Click on “Feedback” 6. Click on “Write New” 7. Write feedback 8. Post feedback
Extensions	System notifies the group members about the feedback via the email notification system
Post Conditions	Feedback created

Table 12 – UC 2

User case number	UC 3
Name	Read Task Feedback
Description	Student wants to read the feedback that was left by the lecturer on his task.
Actor	Student
Pre-Conditions	Student is logged into the system. Feedback has been created by the lecturer.
Basic Flow	1. Click on “Projects” 2. Select the Project 3. Click on “Tasks assigned to you”

	4. Click on “Feedback” 5. Read the feedback
Extensions	
Post Conditions	Feedback read

Table 13 – UC 3

User case number	UC4
Name	Populate Groups
Description	Lecturer populates groups with students.
Actor	Lecturer/Supervisor
Pre-Conditions	Lecturer is logged in. Groups have been created.
Basic Flow	1. Selects “Year” 2. Selects “Module” 3. Selects “List of Students” 4. Checks if any Students have applied to join a group. 5. Assigns students to groups with the help of a drop-down menu.
Extensions	Notifies students about their new assigned groups through the email notification system.
Post Conditions	Students have been assigned to groups.

Table 14 – UC 4

User case number	UC5
Name	Create Task
Description	Student creates a task for the project
Actor	Student
Pre-Conditions	Student is logged in. Student is part of a group. Group has a project.
Basic Flow	7. Click on “Projects” 8. Select the Project 9. Select “Create New Task” 10. Add “Title” and “Description” 11. Add a “Goal” 12. Click “Done”
Extensions	System notifies the rest of the group members about the new task.
Post Conditions	A task has been created and students have to assign themselves to it.

Table 15 – UC 5

Appendix F – Functional Requirements

ID	Functional Requirements	Users
1	Register Account	All Users
2	Log In	All Users
3	Log Out	All Users
4	Read Feedback	All Users
5	Receive Email Notification	All Users
6	View Activity	All Users
7	View Task	All Users
8	View Task Progress	All Users
9	Create Task	Student
10	Assign Task	Student
11	Un-Assign Task	Student
12	Edit Task	Student
13	Delete Task	Student
14	Update Task Status	Student
15	Add Comment	Student
16	Delete Comment	Student
17	Request to Join Group/s	Student
18	Send Email to Group Members	Student
19	Request Feedback	Student
20	Create Activity	Supervisor
21	Delete Activity	Supervisor
22	Set Activity Goal	Supervisor
23	Edit Activity Goal	Supervisor
24	Add Deadline	Supervisor
25	Edit Deadline	Supervisor
26	Give an Extension to Goal Deadline	Supervisor
27	Set Group Limit	Supervisor
28	Create Group/s	Supervisor
29	Delete Group/s	Supervisor
30	Assign Students to group	Supervisor
31	Submit Feedback	Supervisor
32	Create Announcement	Supervisor

Table 16 – Functional Requirements

Appendix G – Non-Functional Requirements

ID	Non-Functional Requirements
1	Interface should also be mobile/tablet friendly
2	Sensitive Data should be encrypted in the database
3	System should remain functional under heavy traffic
4	Pages should not take longer than 2 seconds to load
5	System should log the user out after a period of inactivity
6	System need to be compatible with all the modern browsers
7	The UI must be clear and user-friendly
8	The system will always have to be available and accessible
9	When logging-in, user-roles will be determined based on their userID
10	The System should email the group members within a minute after an update

Table 17 – Non-Functional Requirements

Appendix H – MoSCoW

Functional Requirement	Description	Priority
Register Account	A user will have to register/activate their accounts when using the system for the first time.	Must Have
Log In/Log Out	A user will have to be able to log-in and out of the system	Must Have
Submit/Read Feedback	The focus of the system is the interaction between the students and their supervisors. Reading and submitting feedback is an important part of this interaction.	Must Have
Email notification system	Users involved in an activity/project will be able to send and receive an email notification regarding it.	Must Have
View Activity/Project	It is important that a user can access and view the activity/project that they are involved in.	Must Have
View Task	Tasks are the equivalent of a to-do list within the activity/project and users will have to follow those steps in order to reach their goal.	Must Have
Create/Delete Activity	A supervisor must be able to create and delete an activity/project	Must Have
Create/Delete Task	The user must be able to create and delete their own tasks on the activity/project they are involved in.	Must Have
Edit Task	In case there is a minor change in a task, the user must be able to edit it and its contents.	Must Have
Set/Edit Activity Goal	A supervisor must be able to set and edit the final activity goal to give the students a sense of direction and what they will have to work towards.	Must Have
Create/Delete Group/s	A Supervisor must be able to create several groups for every activity/project they oversee.	Must Have
Assign Student to Group	A Supervisor must be able to assign the Students to the groups to be able to manage the groups properly and keep everything under control.	Must Have
Submit Feedback	It is important that the Supervisor can write and submit their Feedback for the person assigned to an activity or task to see.	Must Have

View Task Progress	A task should have a progress status in which a user responsible for it can set the different progress states.	Should Have
Request to join a group	A Student should be able to show interest and request to join a certain group. It is up to the Supervisor to decide whether the Student can or cannot join a group.	Should Have
Add/Edit Deadline	An Activity/Project should have a deadline and it should be able to get edited in case there is a change.	Should Have
Give an Extension	Furthermore, if a group requires an extension for certain reasons, the Supervisor should be able to give one.	Should Have
Group Limit	A Supervisor should be able to set a group limit based on the project's requirements.	Should Have
Assign/Unassign Task	A Student should be able to assign or unassign themselves from a task.	Should Have
Update Task Status	The Task status should be able to be changed and updated based on the current progress	Should Have
Request Feedback	A Group or Student should be able to request Feedback on tasks or the Project in general from their Supervisor whenever needed.	Could Have
Create Announcement	A Supervisor could be able to create an announcement for all the groups involved in a Project	Could Have
Add/Delete Comment on Task	A comment system between group members on Tasks could be implemented to create a better communication system that can also be documented.	Could Have
Download/Print Projects and its contents	Downloading and printing anything regarding a project will not be required since everything will be editable and stored on the web page.	Won't Have
Students create and join the groups they want	Students will not be able to join the group of their choice and that is a restriction to keep the whole process under the control of the supervisor.	Won't Have
View profiles	A User's profile will only contain the data provided by the university database and there is no need for a "Profile" system.	Won't Have
Direct messages	Direct messages on the system itself are not needed since the email notification system will be the main form of communication.	Won't Have
Edit Personal Data	Anything regarding personal data and user profiles will have to be edited through the university and not through this system.	Won't Have

Table 18 - MoSCoW

Appendix I -Sketches

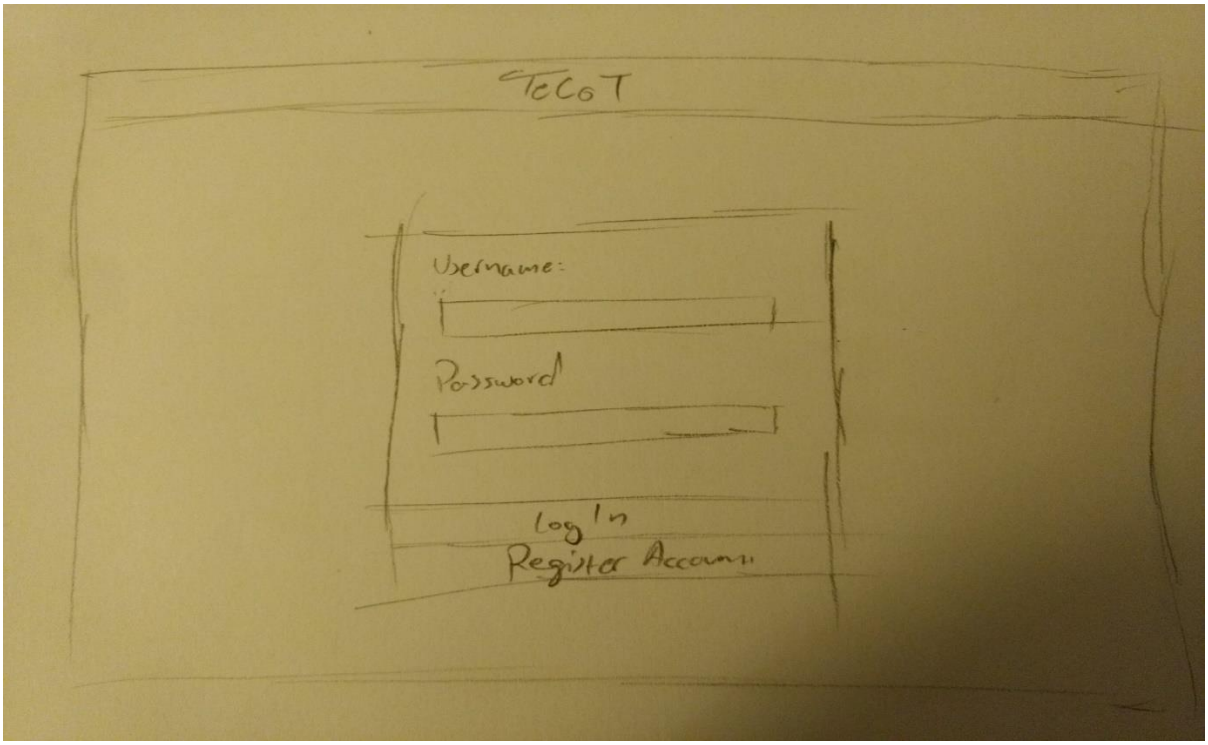

Figure 42 - Log-In Page

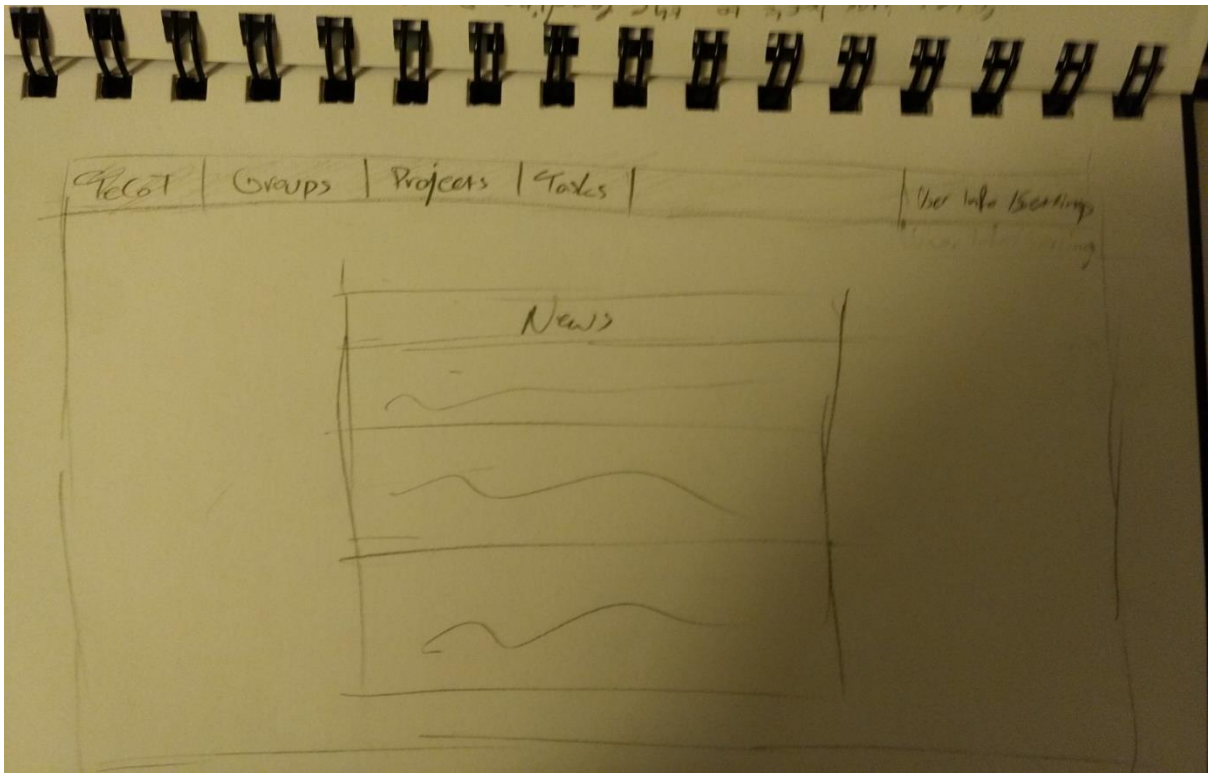

Figure 43 - Home Page

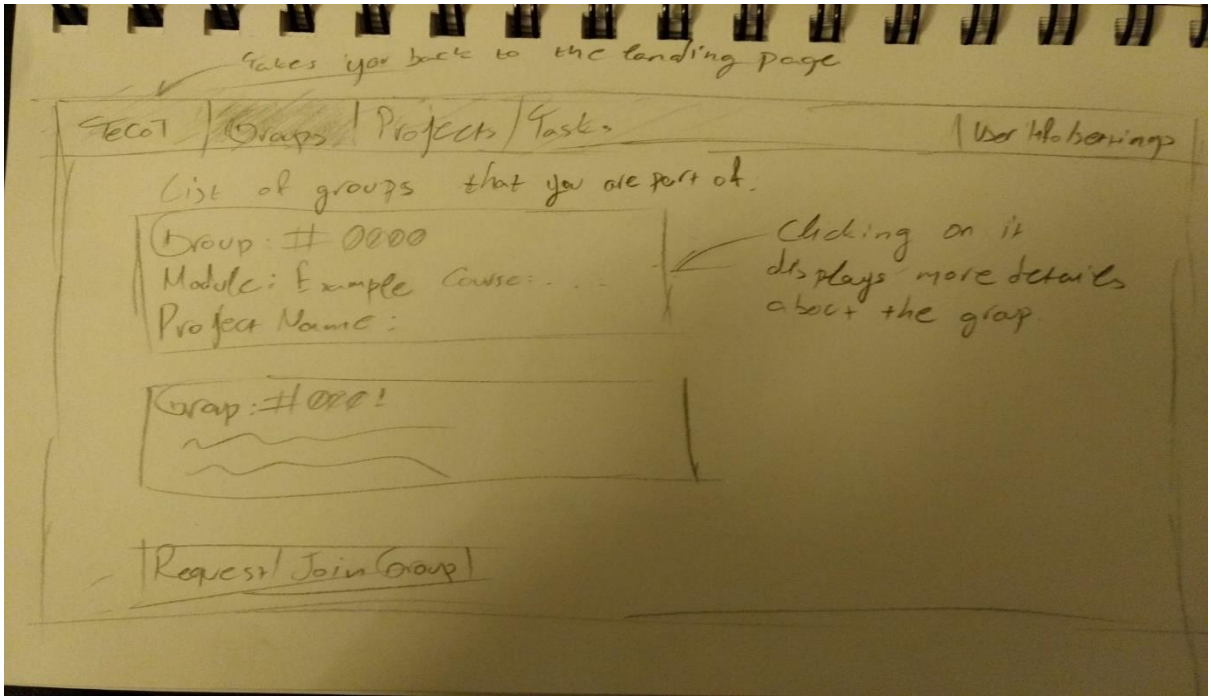

Figure 44 - Students' Group Page

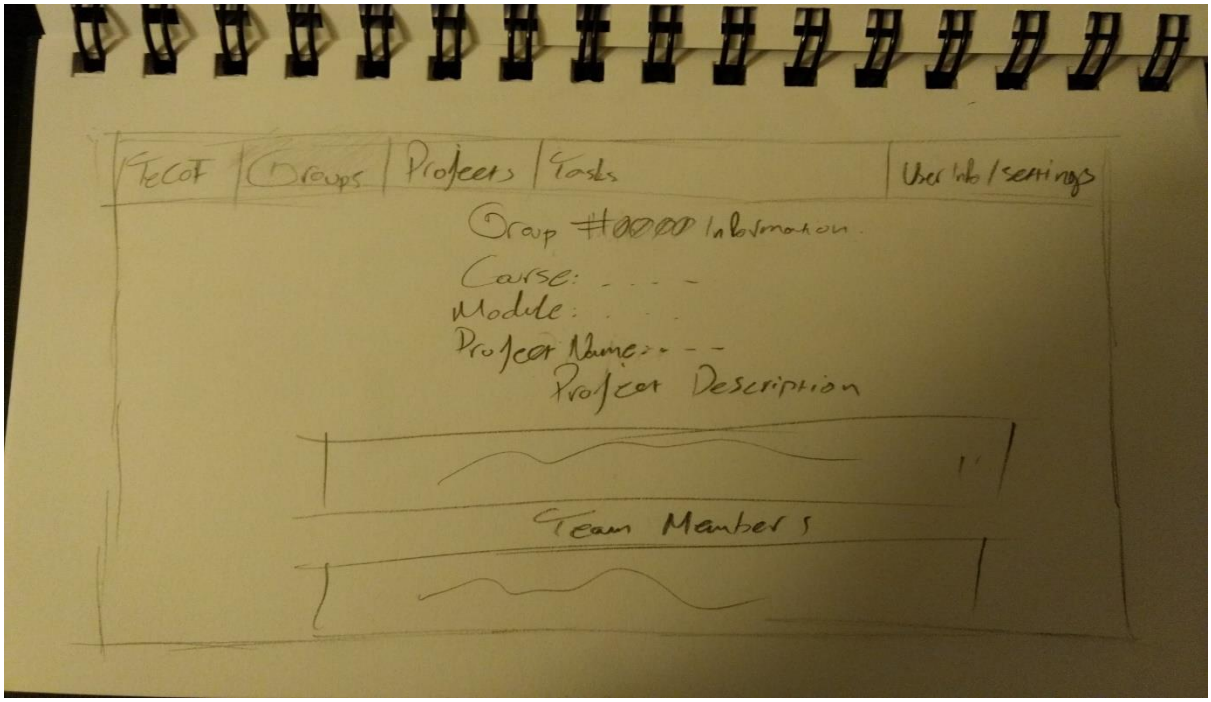

Figure 45 Group Information Page

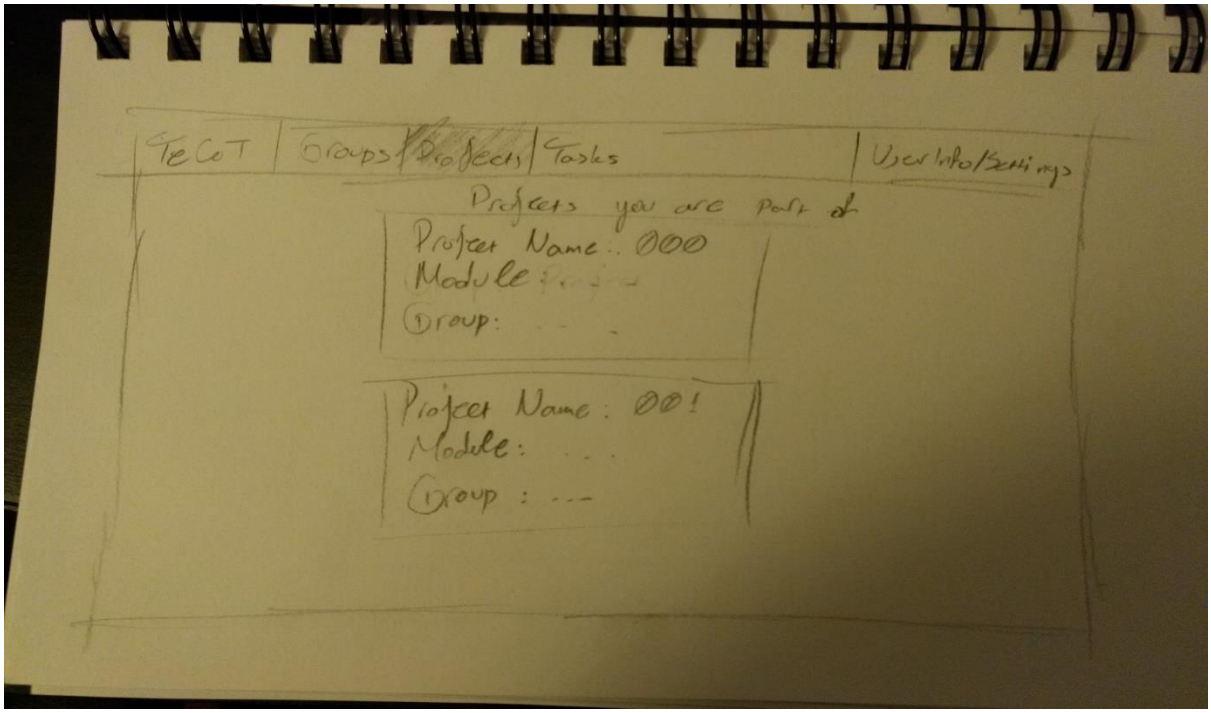

Figure 46 - Students' Projects Page

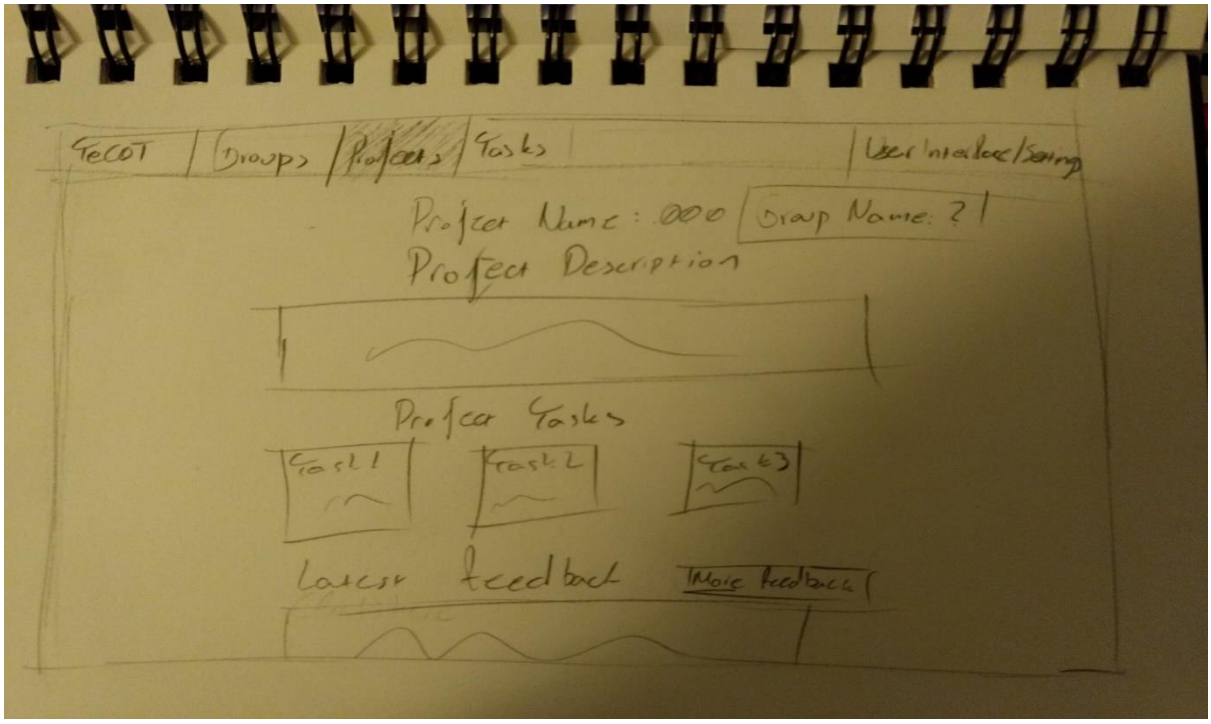

Figure 47 - Project Information Page

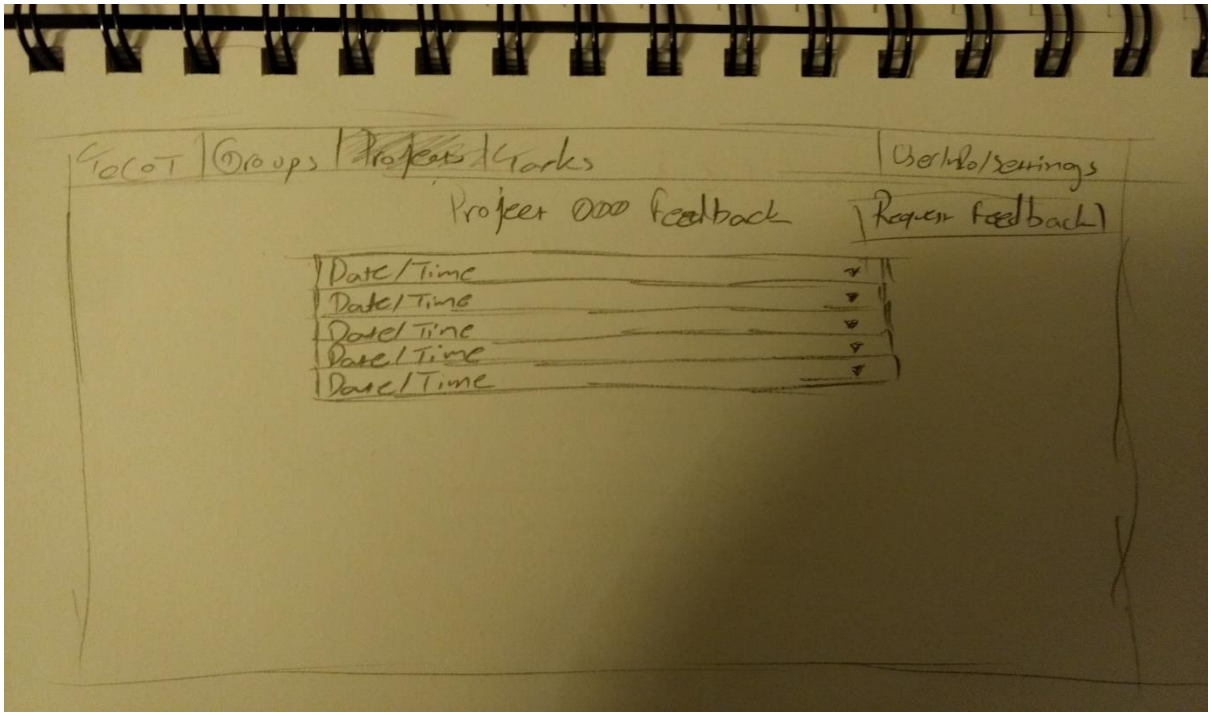

Figure 48 - Project Feedback Page (Initial View)

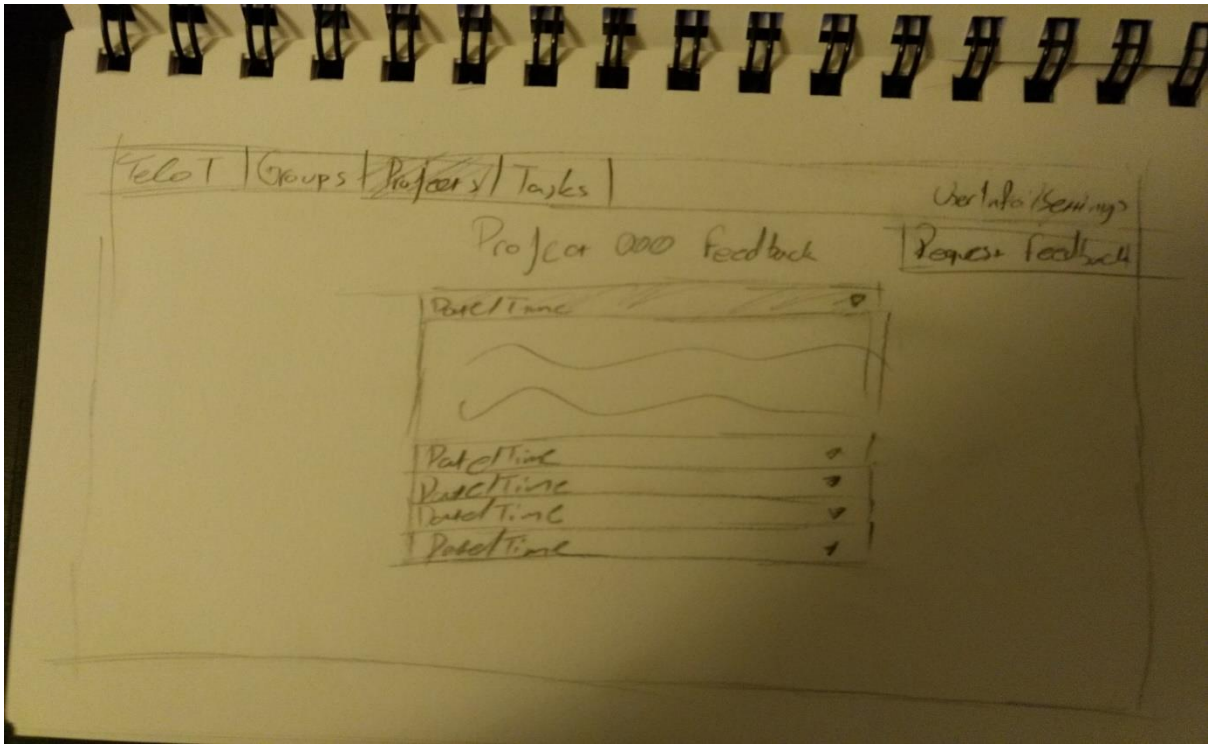

Figure 49 - Project Feedback Page (Date/Time Drop Down)

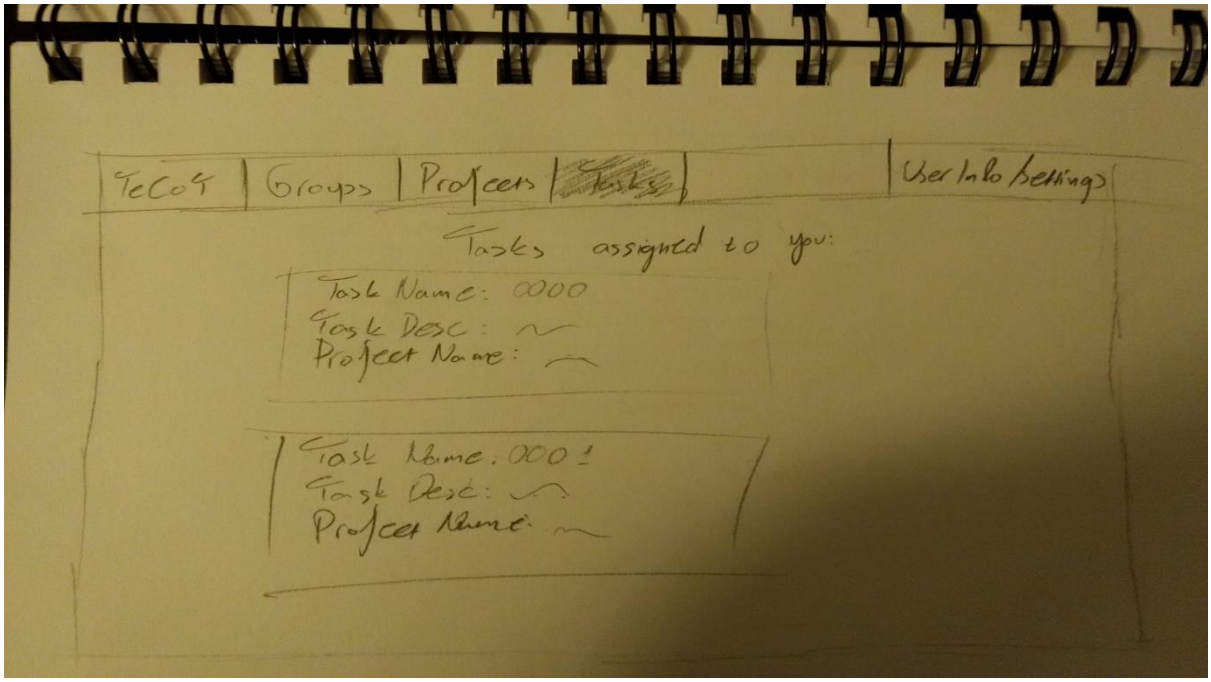

Figure 50 - Students' Tasks Page

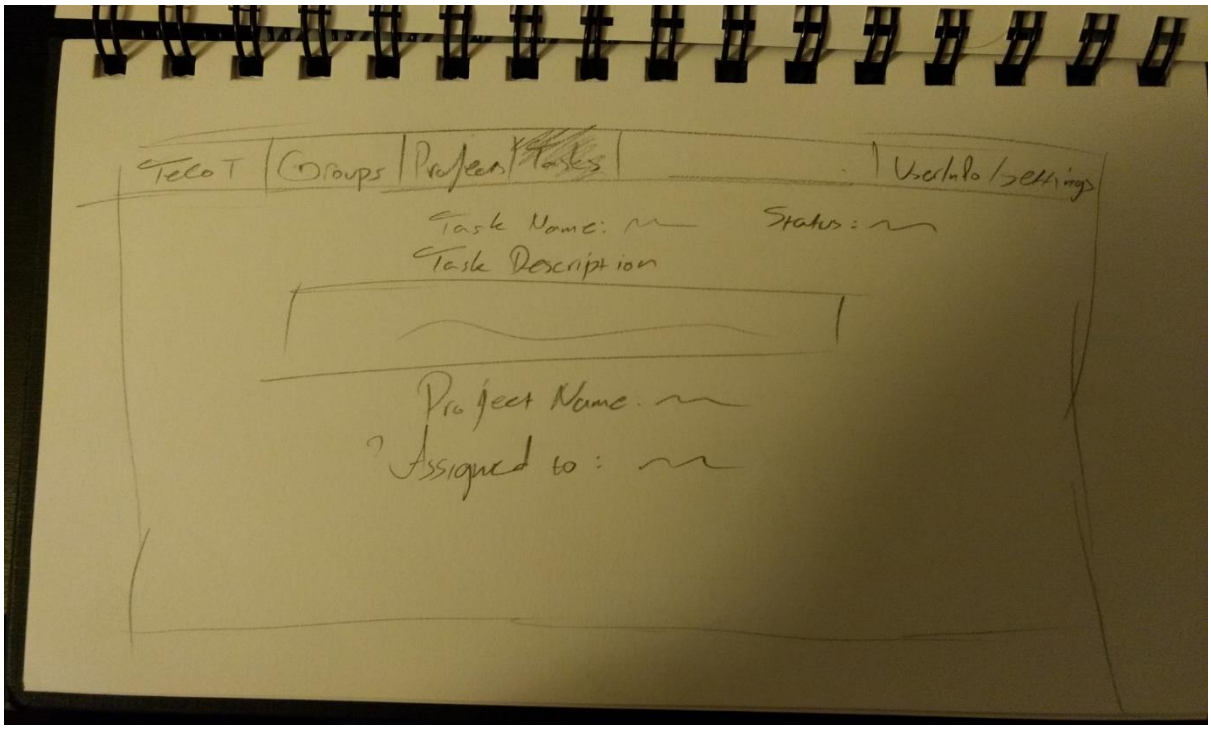

Figure 51 - Task Information Page

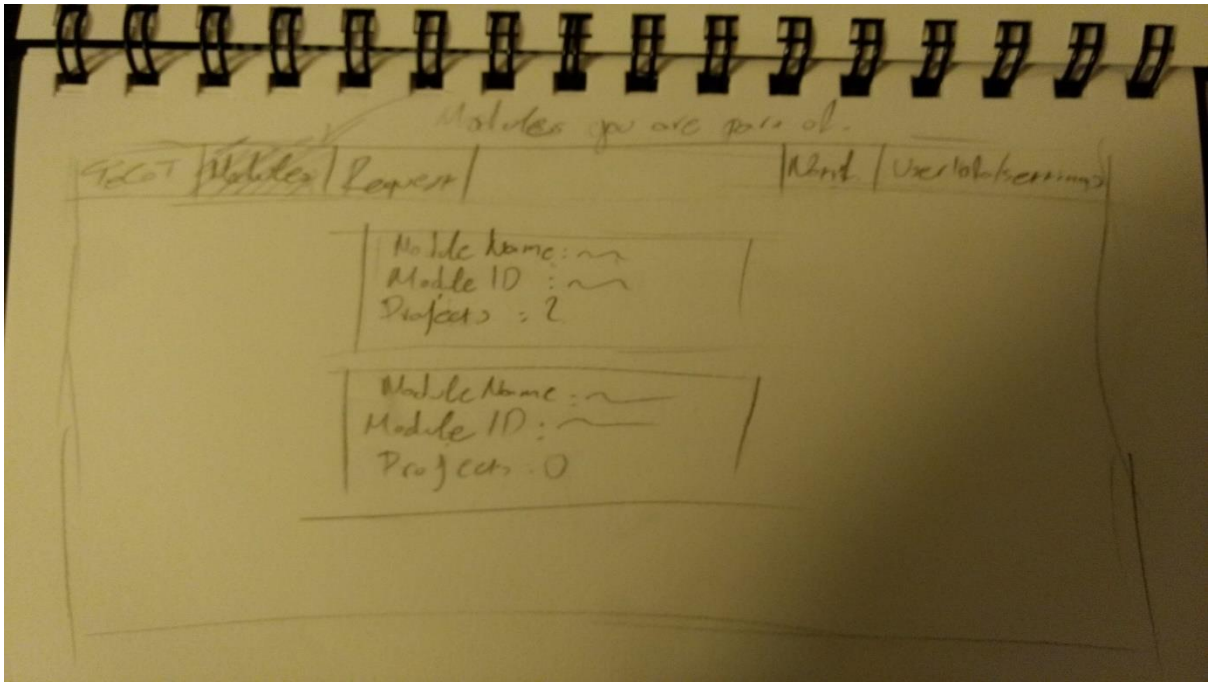

Figure 52 - Lecturers' Modules Page

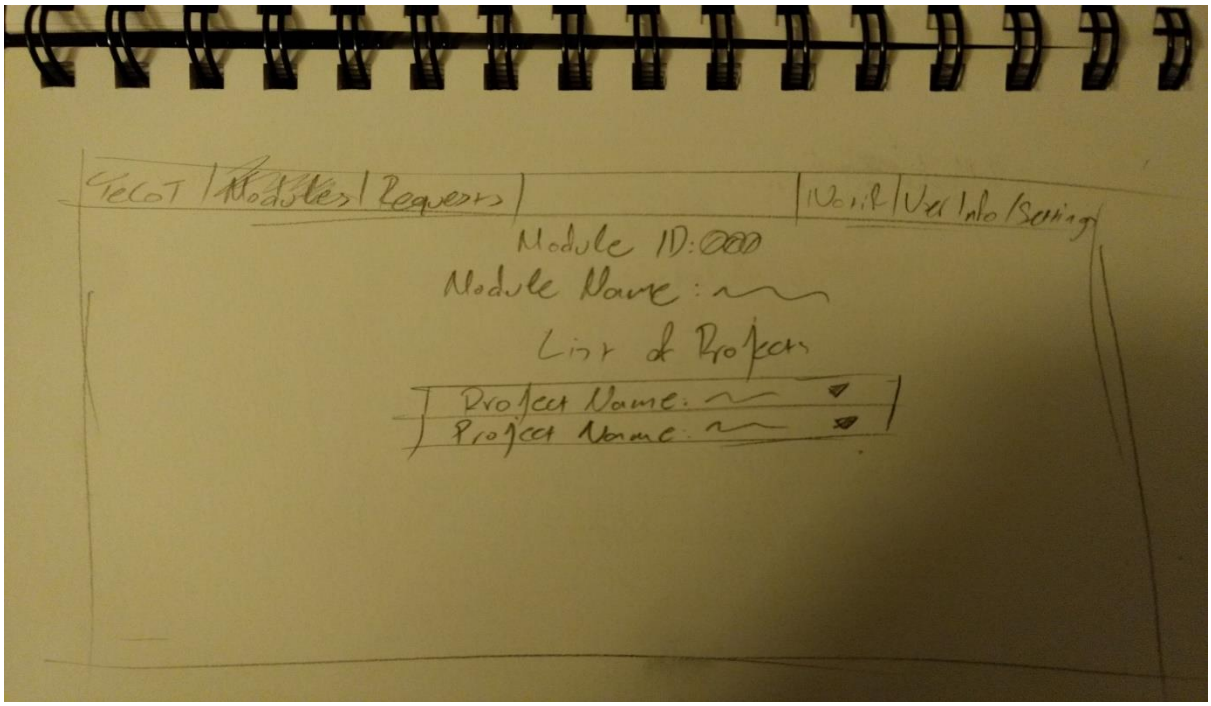

Figure 53 - Module Information Page (Initial View)

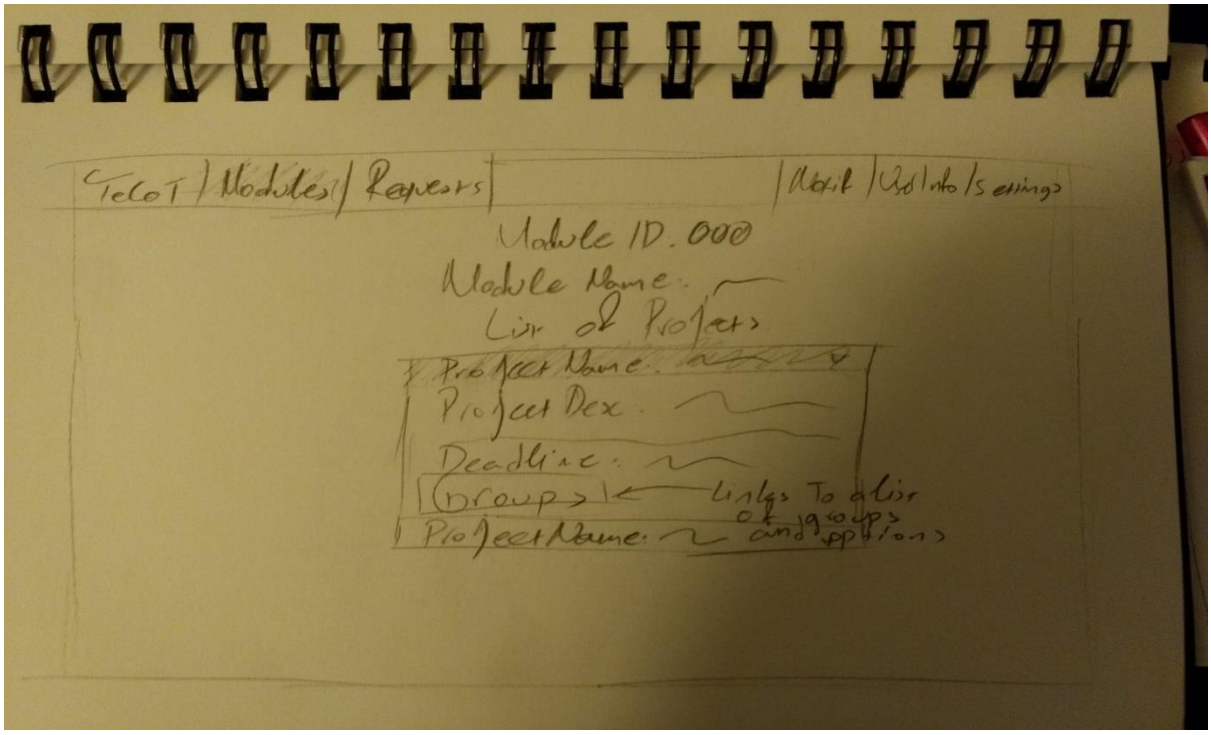

Figure 54 Module Information Page (Project Drop Down)

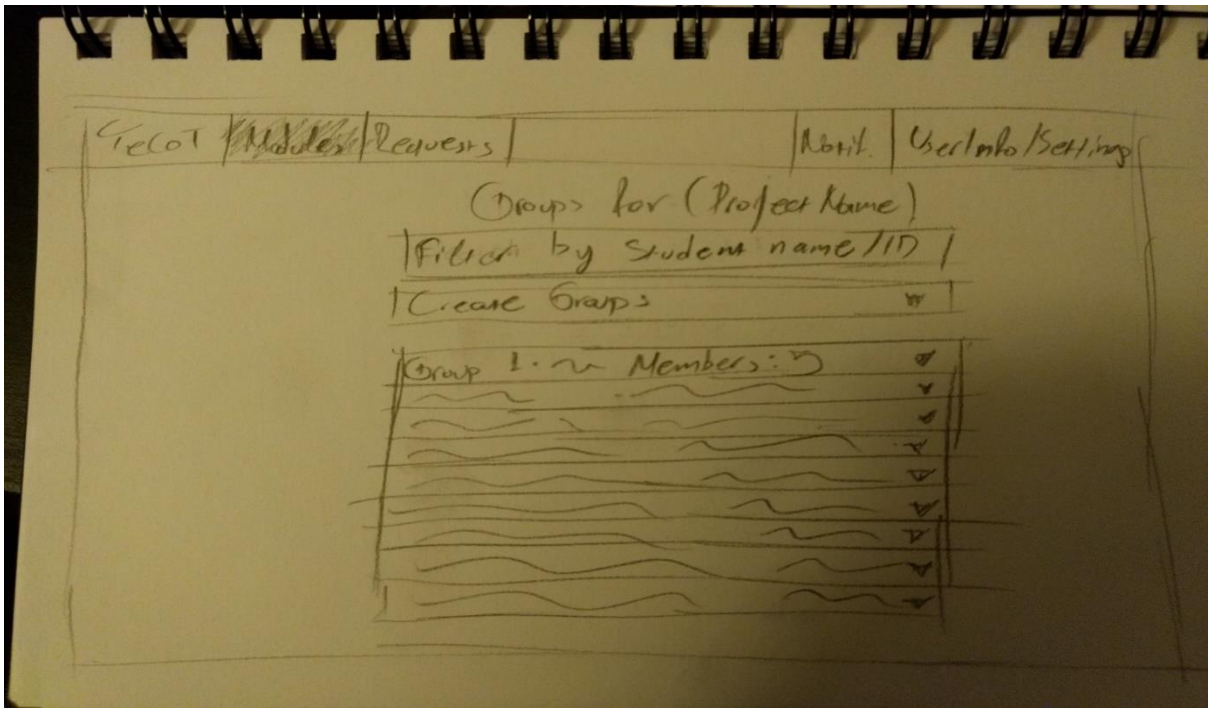

Figure 55 - Groups for a Project Page (Initial View)

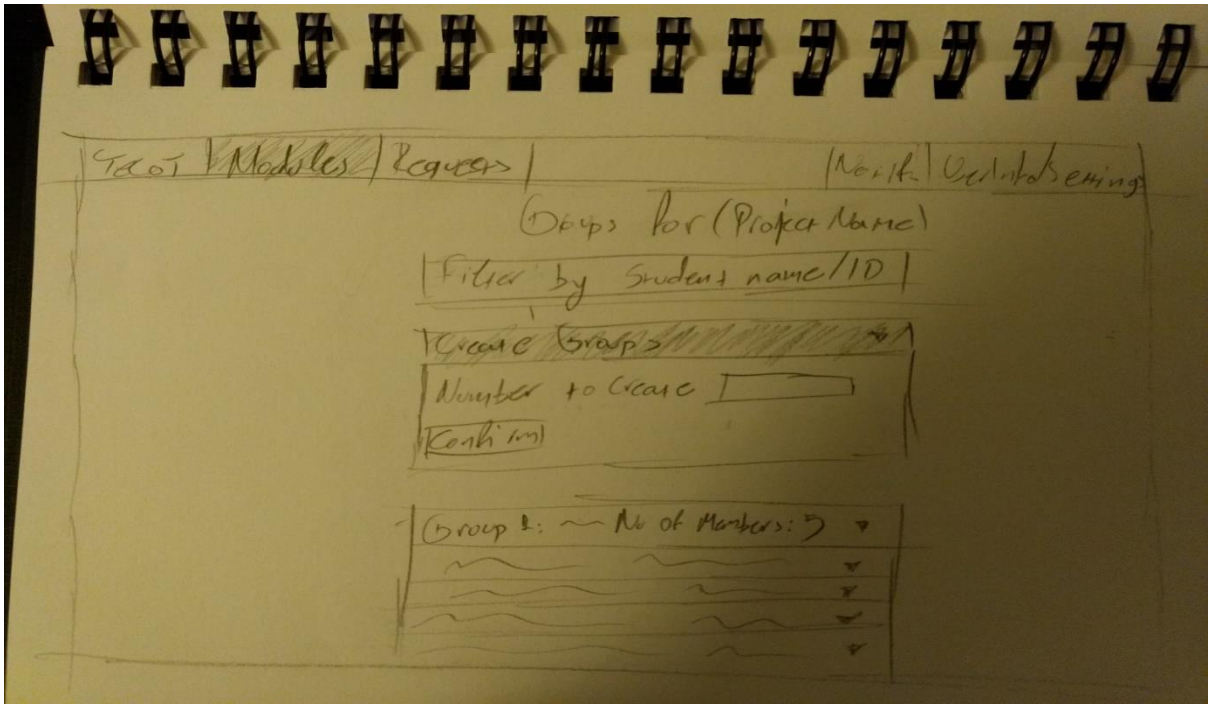

Figure 56 - Group for a Project Page (Group Creation Drop Down)

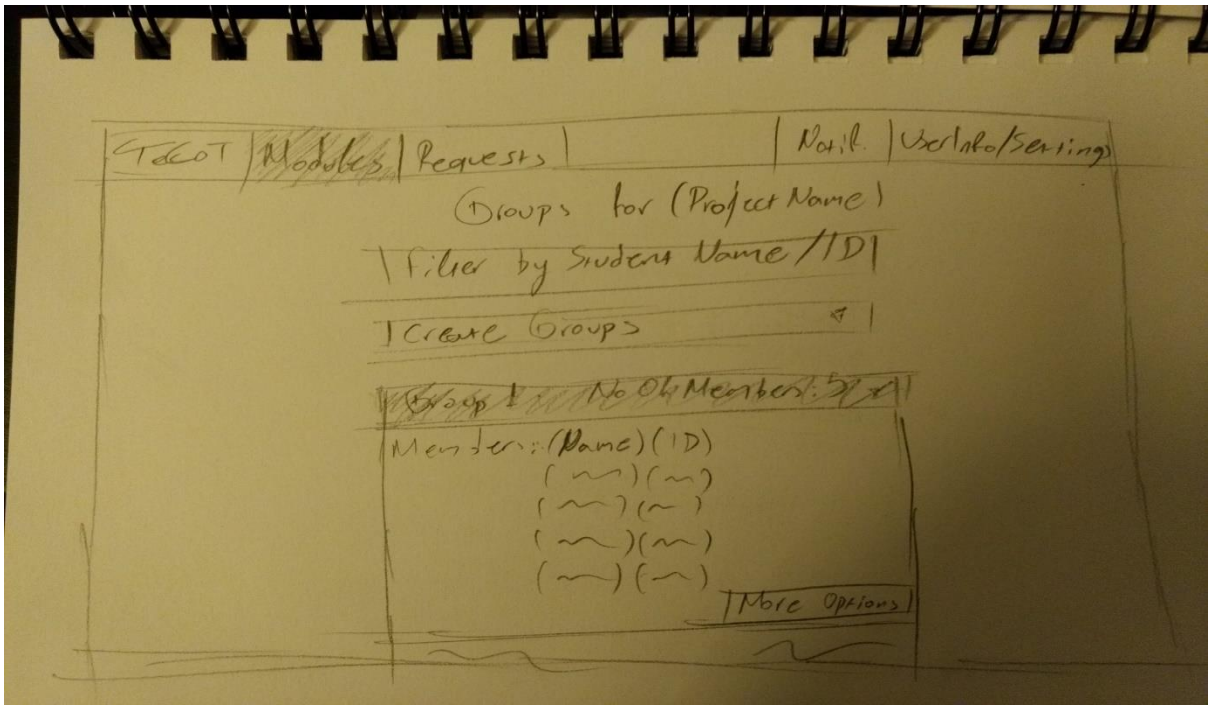

Figure 57 - Group for a Project Page (Group Details Drop Down)

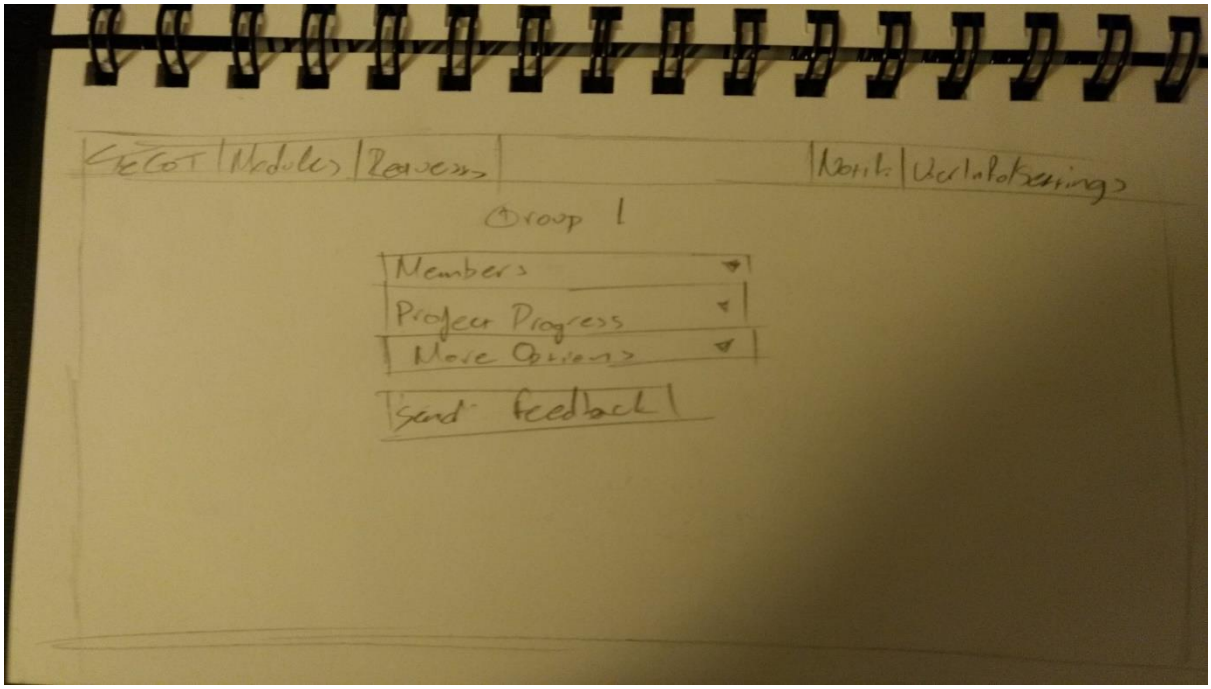

Figure 58 - Group Options (Initial View)

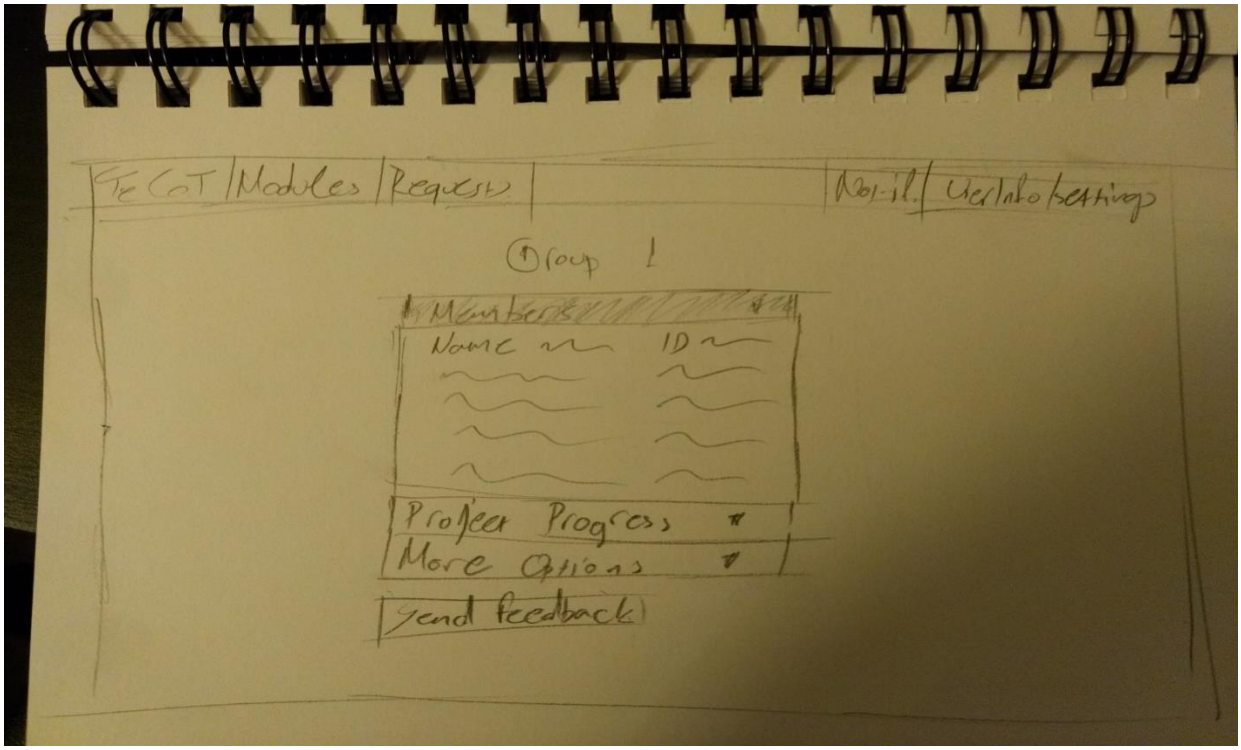

Figure 59 - Group Options (Members Details Drop Down)

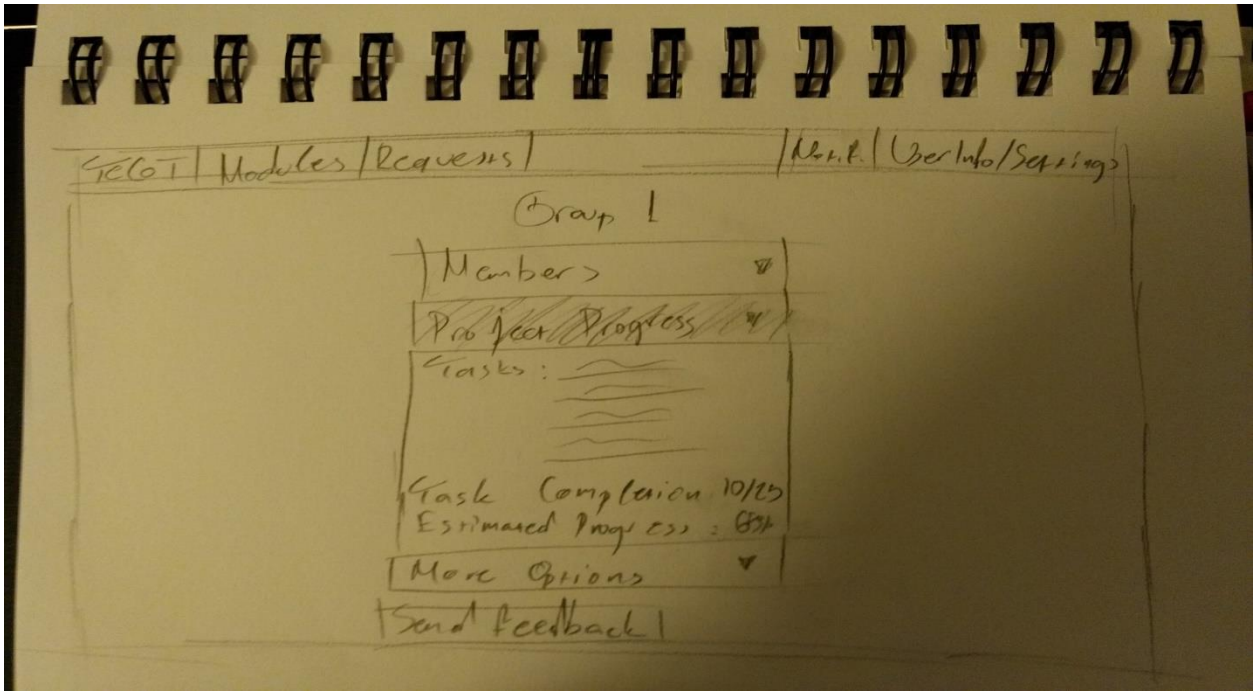

Figure 60 - Group Options (Progress Details Drop Down)

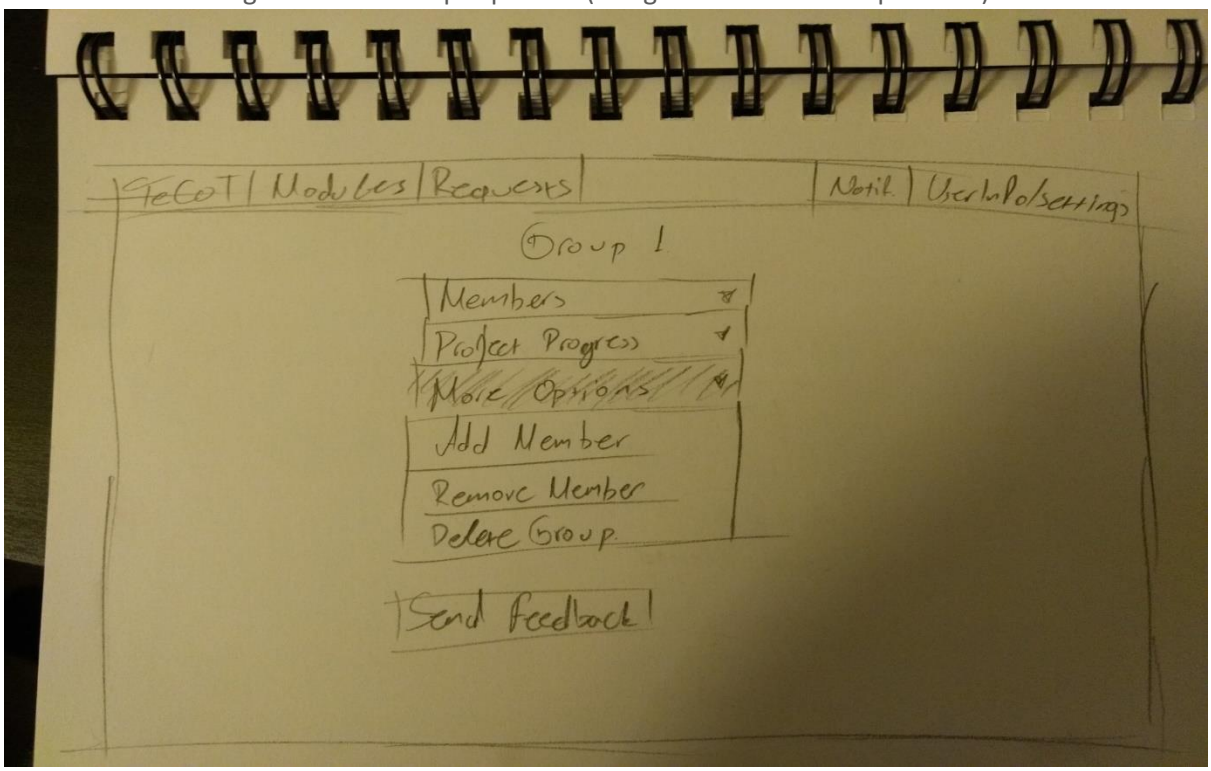

Figure 61 Group Options (More Options Drop Down)

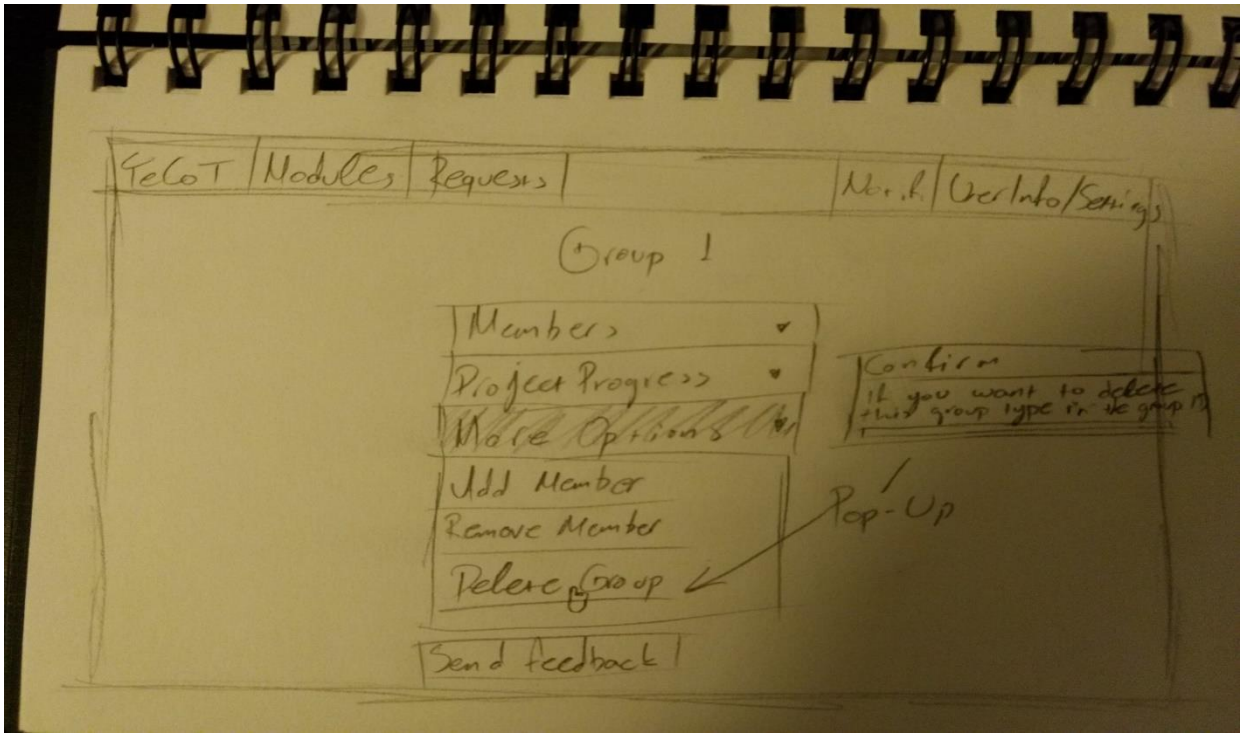

Figure 62 - Group Options (Confirmation Before Deletion Pop Up Window)

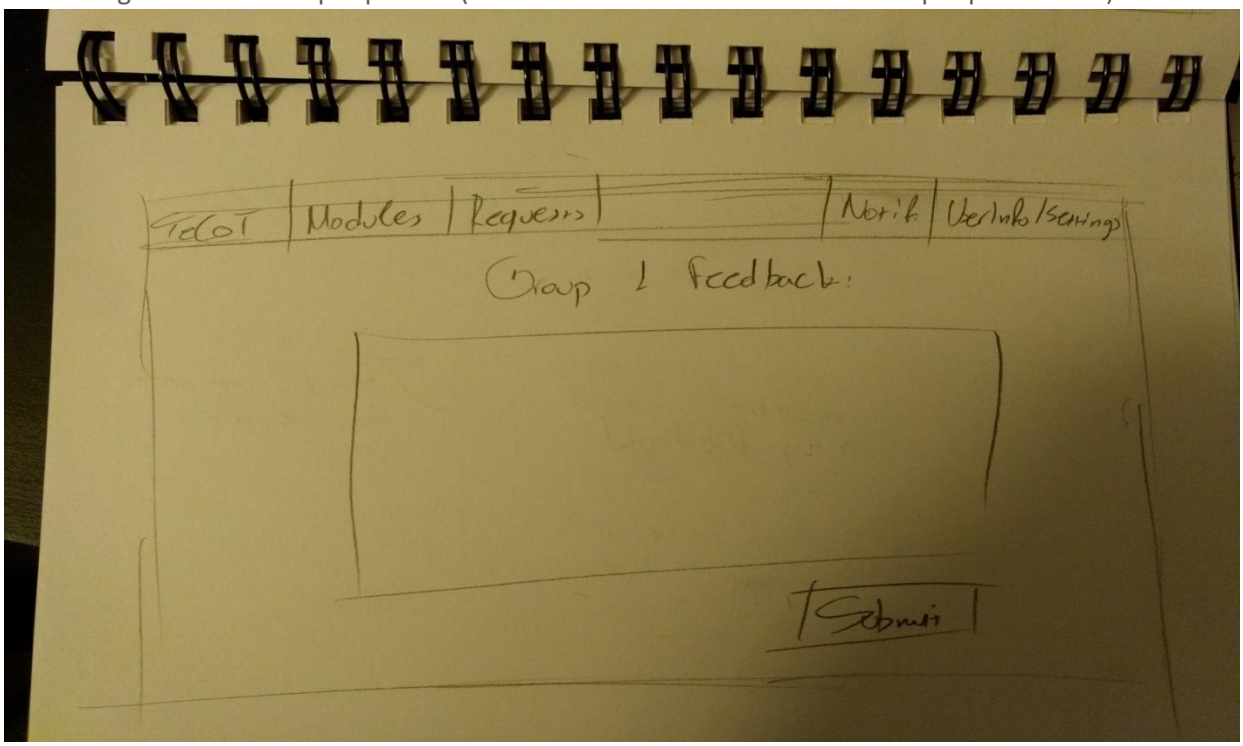

Figure 63 Group Feedback Form

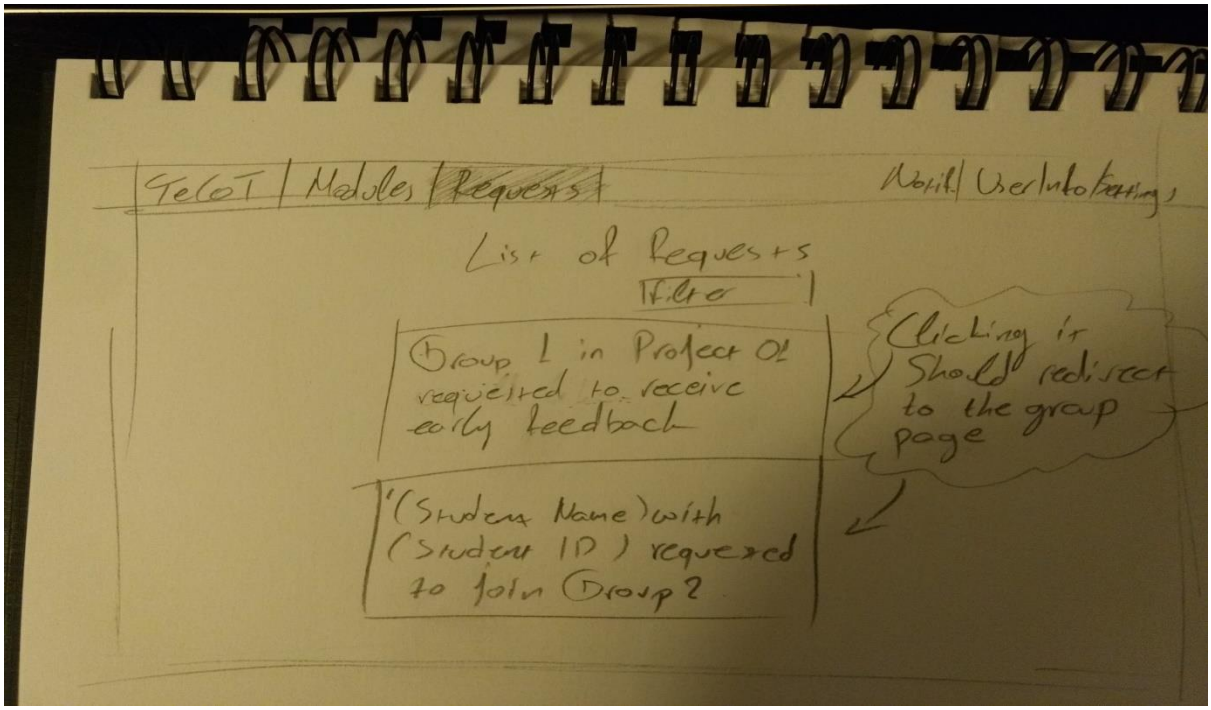

Figure 64 List of Requests Page

Appendix J – Wireframes

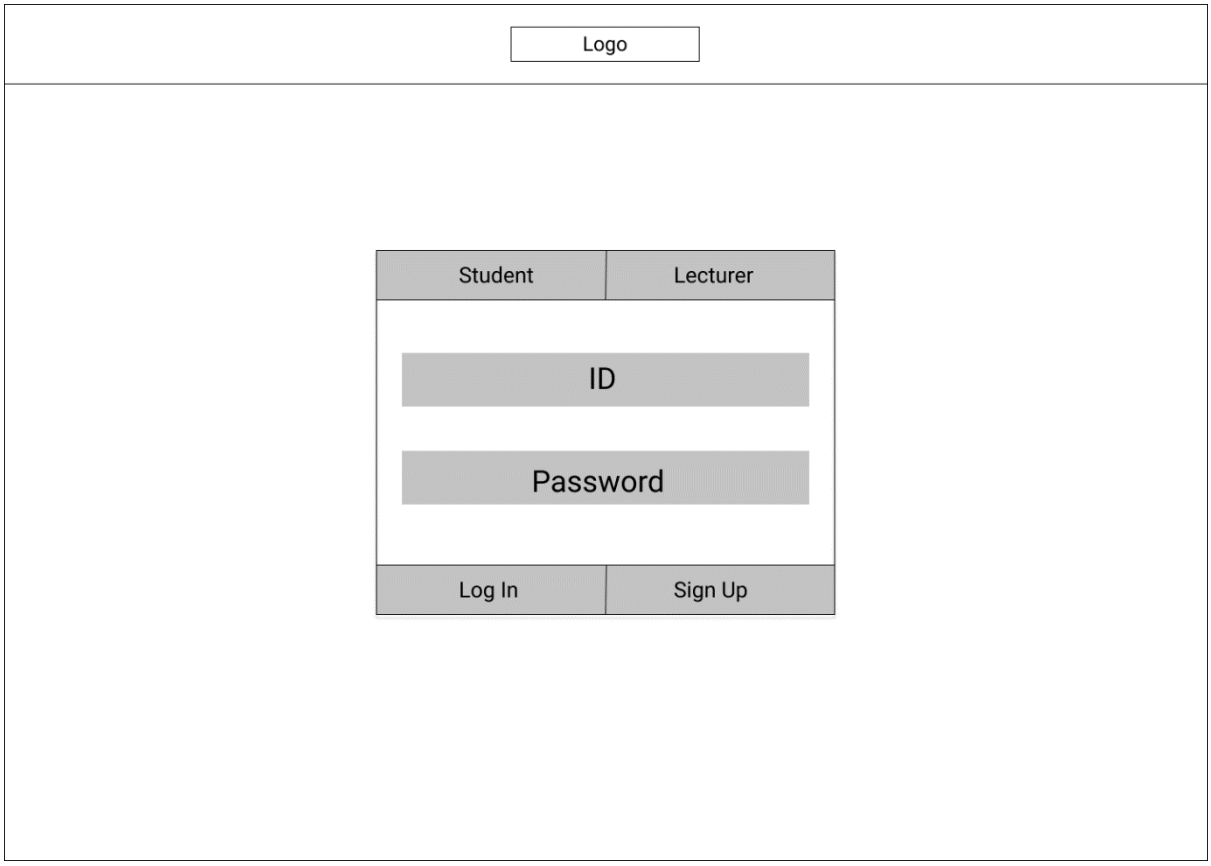

Figure 65 – Early Login screen

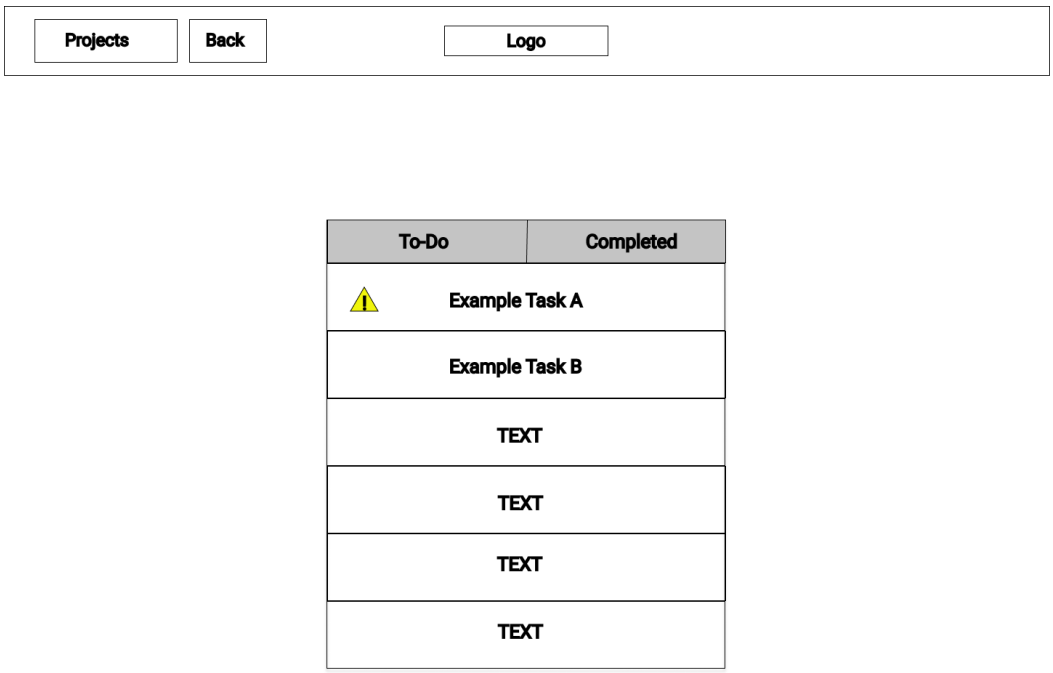

Figure 66 – Early Task List

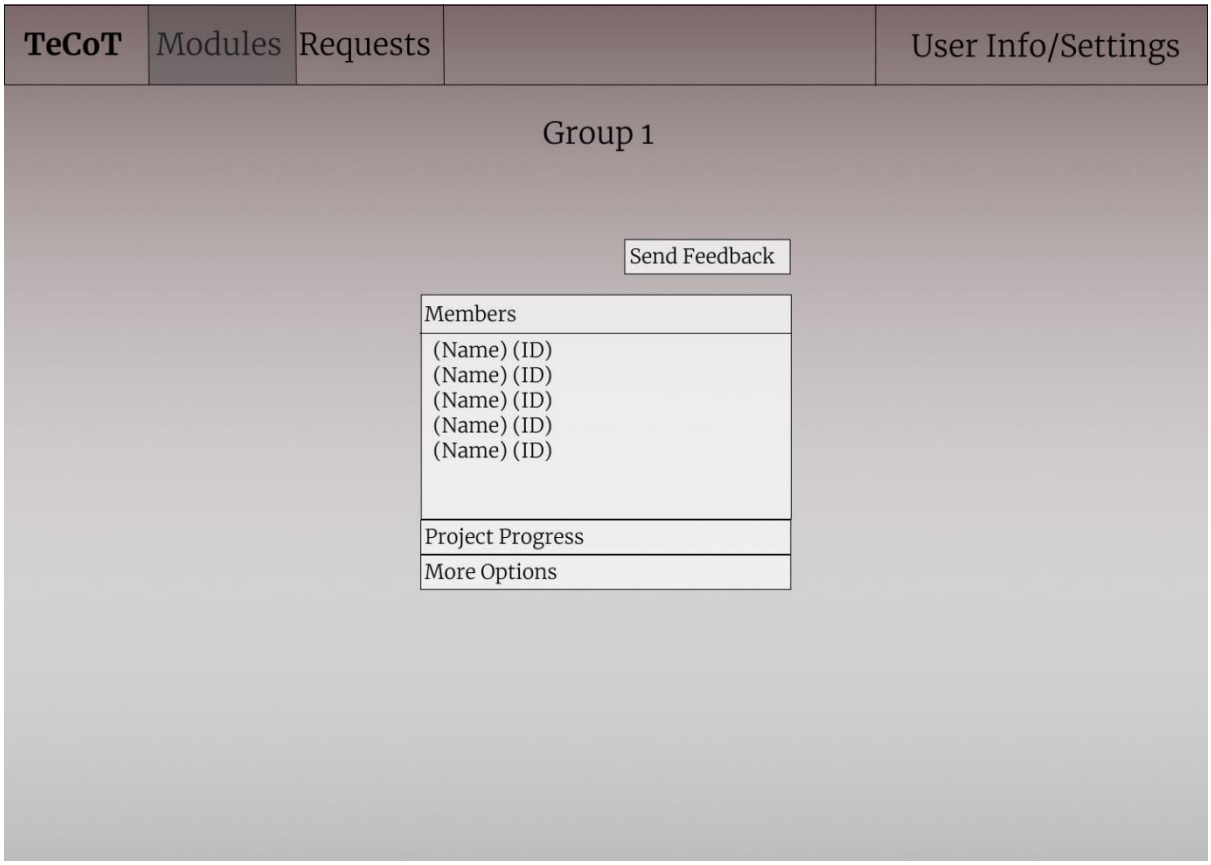

Figure 67 – Group Details

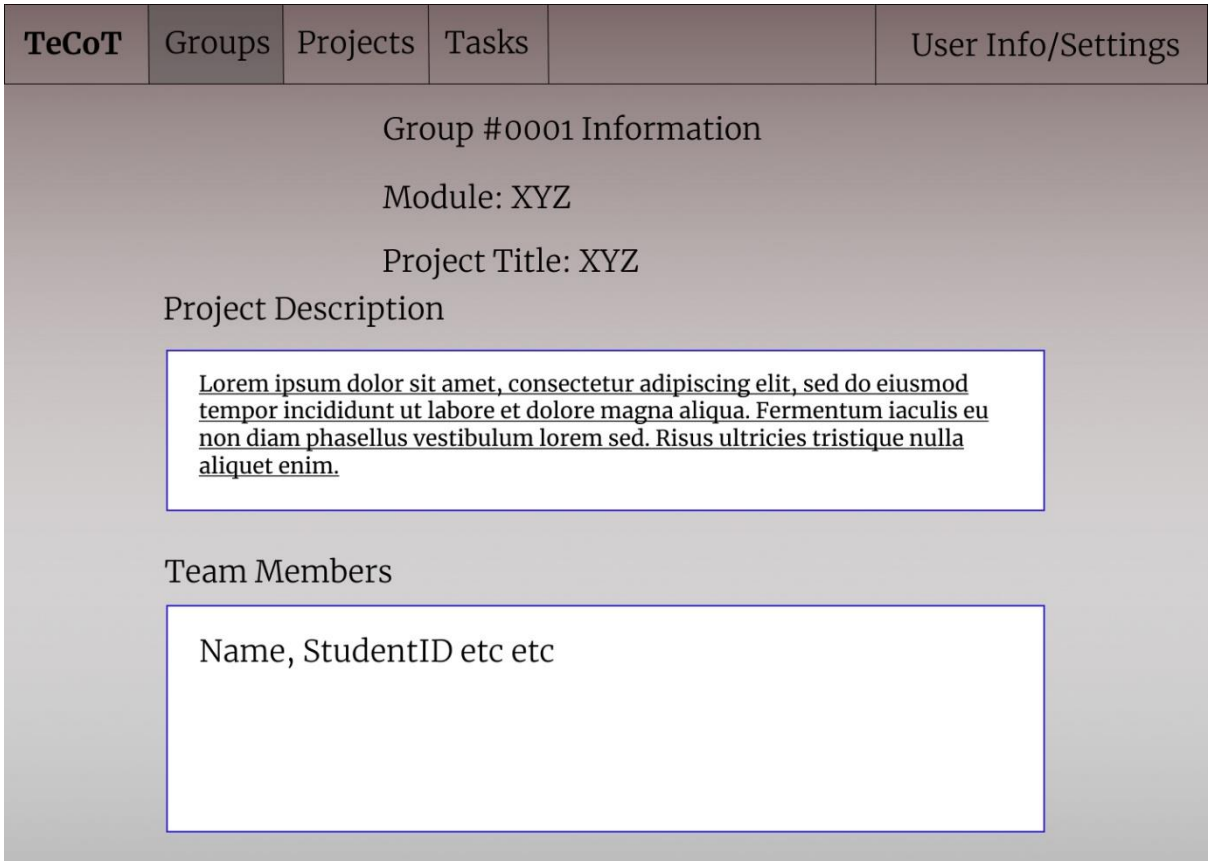

Figure 68 – Project – Team page

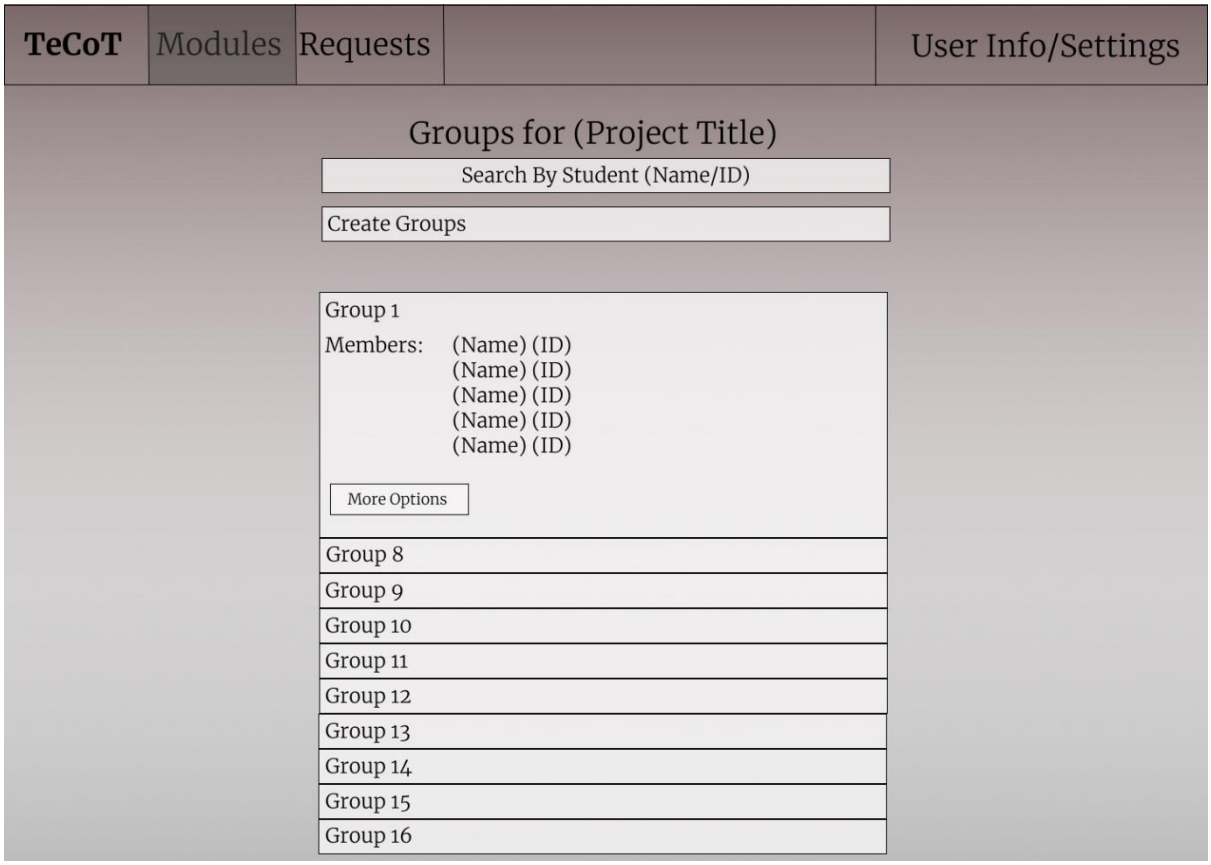

Figure 69 – Group For Project Page

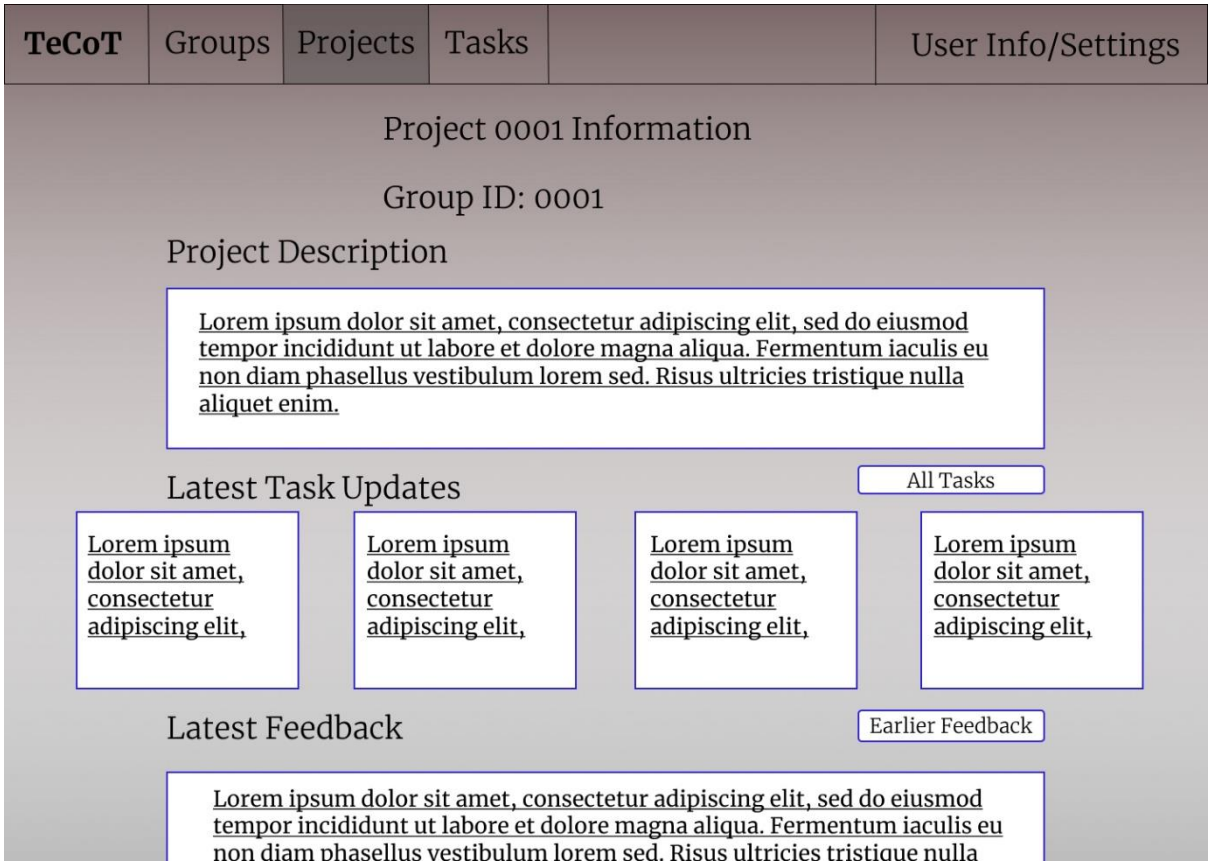

Figure 70 – Project details

Appendix K – Part of the Data CSV

1	**1**	**Kacy**	**Munsterman**
2	3	Candis	Mcwaters
3	1	Arlen	Monfort
4	3	Cherryl	Gearheart
5	3	Tera	Griffis
6	3	Lorita	Calixte
7	1	Claudette	Knights
8	1	Cammy	Cummings
9	1	Eusebio	Wigley
10	1	Joette	Ruggeri
11	1	Dannette	Brubaker
12	3	Trina	Running
13	3	Oralee	Bastien
14	2	Sandee	Ingwersen
15	2	Liz	Waggener
16	1	Kai	Laporta
17	2	Bari	Dipiazza
18	2	Mary	Andersen
19	2	Curt	Pitter
20	1	Ehtel	Ruffo
…	…	…	…
280	3	Cleta	Prow
281	2	Bernardina	Bohm
282	3	Ryan	Parra
283	3	Hana	Dawe
284	2	Blair	Durocher
285	3	Leta	Paras
286	1	Celestine	Ciancio
287	3	Ahmad	Goetsch
288	2	Lashaunda	Cappelli
289	2	Jerald	Hruska
290	2	Arleen	Klippel
291	2	Stephaine	Arcand
292	2	Allison	Cowie
293	2	Ramona	Shuff
294	3	Noelle	Miga
295	1	Katelyn	Crippen
296	3	Bambi	Burgard
297	1	Willian	Courtois
298	2	Cathey	Bonier
299	3	Mckinley	Harshaw
300	1	Kennith	Troiano

Table 19 – Part of the Data CSV

Appendix L – Data Dictionary

Table Name	Field	PK/FK	Datatype	Null
User	u_id	PK	Varchar(25)	No
	r_id	FK	Int(6)	No
	Name		Varchar(25)	No
Role	u_id	PK	Varchar(25)	No
	Name		Varchar(25)	No
Module Leader	ml_id	PK	Int(6)	No
	u_id	FK	Varchar(25)	No
	Name		Varchar(25)	No
Modules	m_id	PK	Int(6)	No
	ml_id	FK	Int(6)	No
	c_id	FK	Int(6)	No
	Name		Varchar(25)	No
Course	c_id	PK	Int(6)	No
	year_id	FK	Int(6)	No
	Title		Varchar(25)	No
Year	year_id	PK	Int(6)	No
	Title		Varchar(25)	No
Team Member	tm_id	PK	Int(6)	No
	te_id	FK	Int(6)	No
	tr_id	FK	Int(3)	No
	u_id	FK	Varchar(25)	No
Team Role	tr_id	PK	Int(6)	No
	Description		Varchar(25)	No
Team	te_id	PK	Int(6)	No
	Name		Varchar(60)	No
Project Team	pt_id	PK	Int(6)	No
	p_id	PK	Int(6)	No
	te_id	FK	Int(6)	No
Project Feedback	pf_id	PK	Int(6)	No
	pt_id	PK	Int(6)	No
	Subject		Varchar(256)	No
	Text		Varchar(256)	No
News/Updates	n_id	PK	Int(6)	No

	pt_id	PK	Int(6)	No
	Content		Varchar(256)	No
Project	p_id	PK	Int(6)	No
	m_id	FK	Int(6)	No
	Name		Varchar(60)	No
	Description		Varchar(256)	No
Task Owner	to_id	PK	Int(6)	No
	task_id	FK	Int(6)	No
	tm_id	FK	Int(6)	No
Task	task_id	PK	Int(6)	No
	pt_id	FK	Int(6)	No
	ts_id	FK	Int(6)	No
	Name		Varchar(256)	No
	Description		Varchar(256)	Yes
Task Status	ts_id	PK	Int(6)	No
	Description		Varchar(256)	No
Task Feedback	tf_id	PK	Int(6)	No
	task_id	FK	Int(6)	No
	Subject		Varchar(256)	No
	Description		Varchar(256)	No

Table 20 – Data Dictionary

Appendix M – Test Tables

ID	FR	SCENARIO	EXPECTED RESULT	RESULT
1	2	User attempts to log in.	Successful log-in: System displays user's home page Failure to log-in: Alert the user that the credentials were incorrect.	Not implement
2	3	User logs out of the system.	System takes the user back to login page.	Implemented the button
3	4	User clicks on the "Project Feedback" button.	System opens the feedback page related to the Project.	Success
4	6	User clicks on the project to view its details.	System displays that project page with all its details.	Success
5	7,8	User clicks on the task to check the task details.	System displays that task page with all its details.	Success
6	8	User views the task progress.	System displays the task status.	Success
7	9	Student creates a new task for a project.	System creates the task and displays it on project/task page. Notifies users related to the project via email.	Success
8	10	Student assigns a task to another student.	System notifies the student via email. System adds the task under that student's tasks. System displays who the task has been assigned to.	Not implement
9	11	Student un-assigns a task.	System removes the task from that student's tasks.	Not implement
10	12	Student edits a task.	System displays the new version of the task. System notifies relevant students.	Success
11	13	Student deletes a task.	System removes the task from the task page and system/database.	Not implement
12	14	Student changes the task status.	System replaces the old status with the new one and notifies relevant users.	Success
13	17	Student requests to join a group.	System sends the request to the lecturer notifying via email or through a notification in the webpage (in order to avoid email clutter). System displays a success message "Request sent".	Not implement
14	19	Student requests feedback on a project.	System sends the request to the relevant supervisor and notifies via email.	Not implement
15	20	Supervisor creates a new activity.	System adds the activity to the projects page and notifies relevant students through email.	Success
16	21	Supervisor deletes an activity.	System removes the activity from the projects page and system/database.	Not implement
17	22	Supervisor sets an activity goal.	System displays the goal in the activity details.	Success
18	23	Supervisor edits an activity goal.	System replaces the old goal with the new one and notifies relevant students.	Not implement

19	24	Supervisor adds a deadline to a project.	System displays the deadline on the related Project page.	Success
20	25, 26	Supervisor edits a deadline of a project.	System replaces the old deadline with the updated version and notifies relevant students.	Not implement
21	28	Supervisor creates a new group.	System (notifies relevant students)? through email. New group is displayed on groups page. Display "New group/s successfully created" message.	Success
22	30	Supervisor assign student to group	System adds the new selected user to the group. Displays the user in the Group Members section and shows up in the user's groups.	Success
23	29	Supervisor deletes a group.	System removes the group from related Groups page and system/database. System notifies relevant students.	Not implement
24	31	Supervisor writes and submits new feedback.	System creates the feedback and it is displayed on the feedback page of the related Project.	Success

ID	NFR	SCENARIO	EXPECTED RESULT	RESULT
1	1	Load the webpage on a desktop browser	The webpage loads and displays things correctly in a desktop browser	Success
2	1	Load the webpage on a mobile browser	The webpage loads and displays things correctly in a mobile browser	Success
3	1	Load the webpage on a tablet browser	The webpage loads and displays things correctly in a tablet browser	Success
4	2	Check if all the sensitive data is encrypted in the database	The data should be hashed/encrypted within the database	Not implement
5	4	Stress test and try accessing various pages	The pages should load in less than 2-3 seconds	Success
6	5	User is inactive for over 15 minutes on the webpage.	System will log the user out after 15 minutes of inactivity	Not implement
7	6	Access the webpage from several different browsers	Everything should function properly when accessing things from widely used browsers (Chrome, Firefox, Edge etc)	Success
8	9	Student logs in with the student ID	Students is redirected to the "Student" homepage	Not implement
9	10	Lecturer logs in with the Lecturer ID	Lecturer is redirected to the "Lecturer" homepage	Not implement
10	11	After an update, the users related to the update should receive an email within a few minutes	User is alerted through a new email within a few minutes	Not implement

Table 21 – Test Tables

Appendix N – Link to the BOX folder

https://kingston.box.com/s/ieazrq0ijga74evv7o1osd95pwi2hslg