ONE COMMUNITY

RAHAND MAHDI AHMED

UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : RAHAND MAHDI AHMED

Date of Birth : May 3, 2000

Title : ONE COMMUNITY

Academic Session : 2022-2023/7

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
✓	**OPEN ACCESS**	I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

SIGNATURE OF STUDENT	**SIGNATURE OF SUPERVISOR**
QU182SCSJ057	MR. KANAR TARIQ
MATRIX NUMBER	**NAME OF SUPERVISOR**
Date: 25 JUNE 2012	Date: 25 JUNE 2022

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

May 2023

QIU Library

Sir,

CLASSIFICATION OF THESIS AS OPEN
ONE COMMUNITY
RAHAND MAHDI AHMED

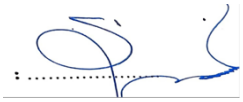
Please be informed that the above-mentioned thesis entitled “ONE COMMUNITY” be classified as OPEN ACCESS.

Thank you.

Sincerely yours.

MR. KANAR TARIQ, As Sulaymaniyah Iraq, +964 770 143 6938

"I hereby declare that we have read this thesis and in my opinion this thesis is suffcient in term of scope and quality for the award of the degree of BSc of Computer Science (Software Engineer)"

Signature	:	
Name of Supervisor	:	MR. KANAR TARIQ
Date	:	25 JULY 2022

ONE COMMUNITY

RAHAND MAHDI AHMED

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software Engineer)

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2022

DECLARATION

I declare that this thesis entitled *"*ONE COMMUNITY*"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	.....
Name	:	RAHAND MAHDI AHMED
Date	:	25 JULY 2022

DEDICATION

This thesis is dedicated to my mother, who taught me that self-learning is the key of success. It is also dedicated to my father, who taught me that the barriers are the opportunity to learn new stuff.

ACKNOWLEDGEMENT

I spoke with a lot of scholars, academicians, and practitioners as I was putting this thesis together. They have helped me to comprehend and think. I want to convey my profound gratitude to Mr. Kanar, who served as my primary thesis supervisor, for his support, direction, criticism, and friendship. This thesis would not have been the same without their ongoing assistance and attention.

It is also appropriate to thank my fellow student for their assistance. All of my co-workers and the individuals who have helped on various times have my genuine gratitude. Their opinions and advice are really helpful. Unfortunately, there isn't enough room to mention them all. I am appreciative of every member of my family.

ABSTRACT

The One Community project is a website aimed at fostering student engagement through a forum-style platform. It provides a lasting forum where students from diverse backgrounds can interact, ask questions, and share knowledge. The system architecture follows the Model-View-Controller (MVC) approach, ensuring a clear separation between the interface, user interactions, and data management. The project incorporates features such as post creation, commenting, categorization, and user management. The data design includes normalized tables for user data, posts, comments, categories, likes, and comment replies. The project's significance lies in promoting global student collaboration, bridging the gap between educational institutions, and facilitating lifelong learning. Suggestions for development include careful planning, functional implementation, and thorough testing to ensure optimal functionality and user satisfaction. Overall, the One Community project offers a platform that encourages student engagement and knowledge sharing beyond traditional educational boundaries.

ABSTRAK

Projek Satu Komuniti ialah laman web yang bertujuan untuk memupuk penglibatan pelajar melalui platform ala forum. Ia menyediakan forum yang berkekalan di mana pelajar dari pelbagai latar belakang boleh berinteraksi, bertanya soalan dan berkongsi pengetahuan. Seni bina sistem mengikut pendekatan Model-View-Controller (MVC), memastikan pemisahan yang jelas antara antara muka, interaksi pengguna dan pengurusan data. Projek ini menggabungkan ciri seperti penciptaan pos, mengulas, pengkategorian dan pengurusan pengguna. Reka bentuk data termasuk jadual ternormal untuk data pengguna, siaran, ulasan, kategori, suka dan balasan ulasan. Kepentingan projek ini terletak pada menggalakkan kerjasama pelajar global, merapatkan jurang antara institusi pendidikan, dan memudahkan pembelajaran sepanjang hayat. Cadangan untuk pembangunan termasuk perancangan teliti, pelaksanaan fungsi dan ujian menyeluruh untuk memastikan kefungsian optimum dan kepuasan pengguna. Secara keseluruhannya, projek Satu Komuniti menawarkan platform yang menggalakkan penglibatan pelajar dan perkongsian pengetahuan melangkaui sempadan pendidikan tradisional.

Table of Contents

TITLE:		PAGE
DECLARATION		**II**
DEDICATION		**III**
ACKNOWLEDGEMENT		**IV**
ABSTRACT		**V**
ABSTRAK		**VI**
LIST OF TABLES		**IX**
LIST OF FIGURES		**X**
CHAPTER 1 INTRODUCTION		**1**
1.1.	INTRODUCTION	1
1.2.	PROBLEM BACKGROUND AND PROPOSED SOLUTION	1
1.3.	PROJECT AIM	2
1.4.	OBJECTIVES	3
1.5.	SCOPES	3
1.6.	PROJECT IMPORTANCE	3
1.7.	GANTT CHART	4
1.8.	PROJECT ORGANIZATION	4
CHAPTER 2 LITERATURE REVIEW		**6**
2.1.	INTRODUCTION	6
2.2.	LMS FEATURES FROM LITERATURE	6
2.3.	FEATURES OF THE LITERATURE	9
2.4.	CHAPTER SUMMARY	10
CHAPTER 3 METHODOLOGY		**11**
3.1.	INTRODUCTION	11
3.2.	METHODOLOGY	11
3.3.	PHASES	12
3.3.1.	IDENTIFY THE PROBLEM	12
3.3.2.	COLLECT AND ANALYZE DATA	13
3.3.3.	SOLUTION	15
3.3.4.	GANTT CHART	15

3.4. TECHNOLOGY TO DEVELOP THE SYSTEM 15
3.5. SYSTEM REQUIREMENT ANALYSIS 16
3.5.1.SOFTWARE REQUIREMENTS 16
3.5.2.HARDWARE REQUIREMENTS 17
3.6. CHAPTER SUMMARY 17
CHAPTER 4 REQUIREMENTS ANALYSIS AND DESIGN 18
4.1. INTRODUCTION 18
4.2. REQUIREMENTS ANALYSIS 18
4.3. DESIGN 20
4.4. DATABASE DESIGN 21
4.4.1.TABLES 21
4.4.2.TABLE RELATIONSHIPS 26
4.5. INTERFACE DESIGN 29
4.5.1.GENERAL INTERFACE 29
4.5.2.STANDARD PAGES 32
4.5.3.ADMIN INTERFACES 35
4.6. SUMMARY 37
CHAPTER 5 IMPLEMENTATION, AND TESTING 38
5.1. INTRODUCTION 38
5.2. ARCHITECTURE STYLE AND RATIONALE 38
5.3. ARCHITECTURE MODEL 38
5.4. DETAILED DESCRIPTION OF COMPONENTS 39
5.4.1.COMPONENT DIAGRAM 39
5.4.2.DETAILED DESCRIPTION 40
5.5. DATA DESIGN 42
5.5.1.DATA DICTIONARY 42
CHAPTER 6 CONCLUSION 44
6.1. INTRODUCTION 44
6.2. ACHIEVEMENT 44
6.3. SUGGESTIONS PLANS 45
REFERENCES 46

LIST OF TABLES

TABLE NO. **TITLE** **PAGE**

TABLE 2-1 FEATURES OF THE LITERATURE....9
TABLE 4-1 TABLE FOR USER 21
TABLE 4-2 TABLE FOR USER 23
TABLE 4-3 TABLE FOR USER 24
TABLE 4-4 TABLE FOR USER 25
TABLE 5-1 DATABASE NORMALIZATION TABLE FOR USER DATA.... 42
TABLE 5-2 DATABASE NORMALIZATION TABLE FOR POST 42
TABLE 5-3 DATABASE NORMALIZATION TABLE FOR COMMENT 43
TABLE 5-4 DATABASE NORMALIZATION TABLE FOR CATEGORY 43
TABLE 5-5 DATABASE NORMALIZATION TABLE FOR LIKE 43
TABLE 5-6 DATABASE NORMALIZATION TABLE FOR COMMENT_REPLAYS 43

LIST OF FIGURES

FIGURE NO. TITLE PAGE

FIGURE 1-1 A GANTT CHART THAT EXPLAINS THE TIMELINE AND MILESTONES OF EACH SUBMISSION 4
FIGURE 3-1 A SURVEY FOR QUESTION 1 13
FIGURE 3-2 A SURVEY FOR QUESTION 2 13
FIGURE 3-3 A SURVEY FOR QUESTION 3 14
FIGURE 3-4 A SURVEY FOR QUESTION 4 14
FIGURE 3-5 A GANTT CHART EXPLAINS THE TIMELINE AND MILESTONES OF EACH SUBMISSION..... 15
FIGURE 4-1 THE USE CASE OF THE SYSTEM THAT EXPLAIN. THE FUNCTIONALITY OF EACH TYPE OF USERS 19
FIGURE 4-2 THE FLOW CHART OF THE ONE COMMUNITY WEBSITE THAT EASES THE PROCESS FOR THE BACKEND DEVELOPER 20
FIGURE 4-3 THE ONE-TO-MANY RELATIONSHIP BETWEEN USER TABLE AND POST TABLE..... 26
FIGURE 4-4 THE ONE-TO-MANY RELATIONSHIP BETWEEN COMMENT TABLE AND POST TABLE 27
FIGURE 4-5 THE ONE-TO-MANY RELATIONSHIP BETWEEN USER TABLE AND COMMENT TABLE 28
FIGURE 4-6 THE ONE-TO-MANY RELATIONSHIP BETWEEN POST TABLE AND CATEGORY TABLE..... 28
FIGURE 4-7 THE POSSIBLE FRONT-END INTERFACE FOR SIGN IN PAGE..... 30
FIGURE 4-8 THE POSSIBLE FRONT-END INTERFACE FOR SIGN UP PAGE..... 31
FIGURE 4-9 THE POSSIBLE FRONT-END INTERFACE FOR HOME PAGE FOR GENERAL USER TYPE 32
FIGURE 4-10 THE POSSIBLE FRONT-END INTERFACE FOR PROFILE PAGE FOR GENERAL USER TYPE .. 33
FIGURE 4-11 THE POSSIBLE FRONT-END INTERFACE FOR MY POST PAGE FOR GENERAL USER TYPE . 34
FIGURE 4-12 THE POSSIBLE FRONT-END INTERFACE FOR PUBLISH A POST PAGE FOR GENERAL USER TYPE..... 35
FIGURE 4-13 THE POSSIBLE FRONT-END INTERFACE FOR USERS PAGE FOR ADMIN USER TYPE..... 35
FIGURE 4-14 THE POSSIBLE FRONT-END INTERFACE FOR POSTS PAGE FOR ADMIN USER TYPE..... 36
FIGURE 4-15 THE POSSIBLE FRONT-END INTERFACE FOR CATEGORIES PAGE FOR ADMIN USER TYPE37
FIGURE 5-1 COMPONENT MODEL OF ONE COMMUNITY PROJECT..... 39
FIGURE 5-2 COMPONENT DIAGRAM OF ONE COMMUNITY PROJECT..... 40
FIGURE 5-3 CLASS DIAGRAM OF ONE COMMUNITY PROJECT..... 40
FIGURE 5-4 SEQUENCE DIAGRAM FOR ADMIN..... 41
FIGURE 5-5 SEQUENCE DIAGRAM FOR NON-ADMIN..... 41

LIST OF APPENDENCIES

APPENDIX A 47
APPENDIX B 70
APPENDIX C 85

Chapter 1

Introduction

1.1. Introduction

Nowadays, online forums are commonly utilized to bring people together who share common interests in order for them to share and exchange information and ideas. Forum has been one of the first to go options for discussions about a specific topic. It has been a way of finding solutions for the upcoming issues and the ones that are expected to happen. By the time of coming internet and new technologies, the forum has been updated and digitalized. Students, generally, appreciate the participation in e-learning and studying in this less traditional way

In this way, more people from different locations can engage in the forums. Also, e-learnings holds a big part of this. Almost all of the learning management systems integrate forum features into the systems because this is a way to enhance student's engagement. However, this is not complete and does not satisfy student's needs. Taking Moodle as an example, it is the most widely used in Europe and the Middle East. However, Moodle's all facilities do not impose the use of the virtual environment in the learning process, although all university students have accounts and are enrolled in semester courses [2].

1.2. Problem background and Proposed Solution

It is true that teachers and students in these years prefer distance learning. This is a big solution for many problems. A vast number of teachers believes that distance education improves proves the quality of education [3]. However, this raises new

issues. The main issue with the existing e-learning forums is that they can only be opened by the instructor and per-subject or classes. Meaning, it is limited and controlled by instructors. Students' role starts after their action, not by the students themselves. Nevertheless, it does not stay forever so that the students can refer back to it. If a student wants to find a solution for the same issue discussed last year on the same class and subject, it is not possible because after finishing a class the course will be deleted and a new one will be created and all forums will be gone as well. Even if the class does not get deleted, the forum will not be accessible by others. Another issue is that only the students of the same university in the same course can check the forums. This is only limited to a few groups of students in the world, which is another main problem that does not allow other students in the world to seek around for a solution for the exact same issue.

In this paper, I am introducing a solution for these repeated issues in the existing e-learnings. This can be solved by having a forum website for all the students in the world regardless of their locations, universities, classes, or any categories. On this website, students will have their own profiles that include information about the student. Here, students engage by asking questions about any subjects related to their classes. Any student from anywhere can respond and engage. This way the forums will stay forever.

1.3. Project aim

The aim of this project is that it allows the forums to be accessible even after classes get deleted and the students will be able to go back to it even if the course is finished.

Another aim is that it allows other students, from any part of the world, to seek the solution to the same issues that has faced other students as well with suitable solutions.

1.4. Objectives

The objectives of the project are:

i. To provide an educational environment for students and improve communication between students.
ii. To be accessible at any time and not be deleted and other students with the related issue can take advantage of it whenever they want.
iii. To allow students to check the user profiles to evaluate the answers based on the background of the user.
iv. To make it easier to find a solution to a problem at the university level.

1.5. Scopes

This project focuses on providing certain forum services in which the students will be able to exchange their ideas and ask for help to find solutions for a certain problem, regardless of what the subject is whether it is IT, HR or accounting. It could be a solution for an exams question. One of the limitations this project will have been the fact that it does not have newsfeed features. The student only gets to see the newest posts rather than on their own desires. The second limitation is the absence of chatting. This is due to the fact that this feature needs some programming skills in which we have not studied so far, therefore I have only used the resources that I have studied at the university.

1.6. Project Importance

I decided to create this system because it allows students to submit questions and receive replies without having to sift through an inbox full of emails. Instead of searching through an inbox full of emails, you can just pose a question in the forum

and view everyone's responses in one place.in this system students can post and share their own ideas, also students can post questions or thoughts, and the communication between students. The idea of a forum made many terms much more understandable and easier to follow and improve. It is another way you learn the material through teaching it to someone else [4].

1.7. Gantt Chart

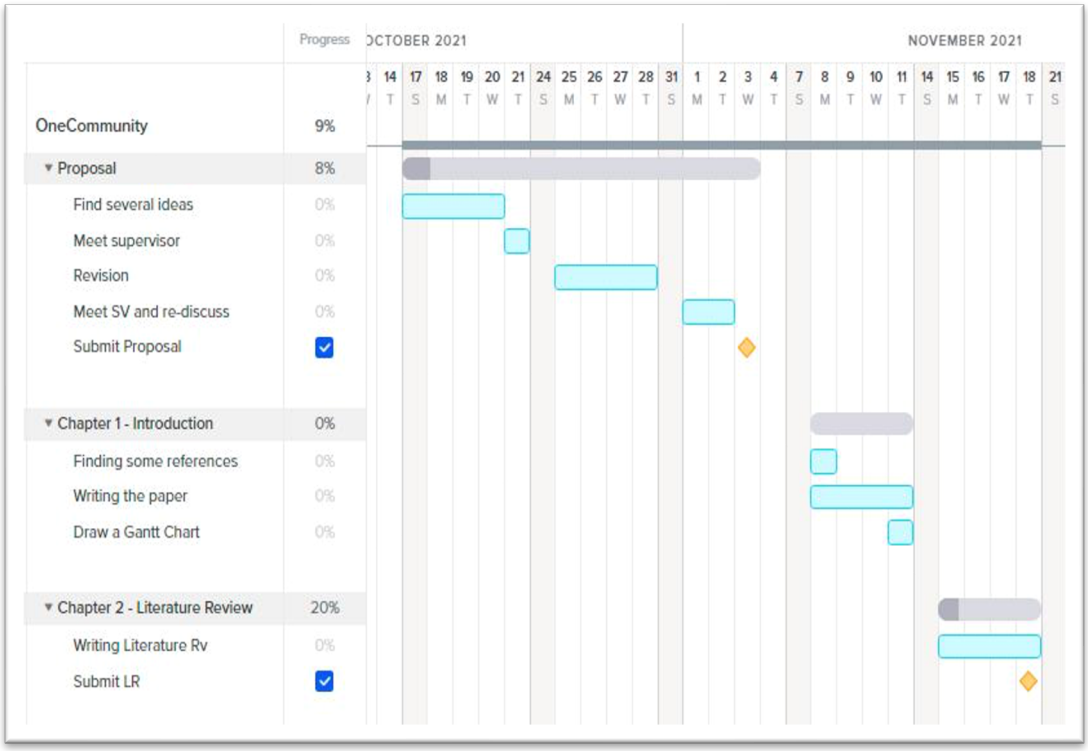


Figure 1-1 A Gantt chart that explains the timeline and milestones of each submission

1.8. Project Organization

In this chapter, we have 6 main part which is the introduction problem background, proposed solution, project aim, objective, scope, and project importance. And the next chapter which is chapter two we have a literature review. Also, this project conation 5 chapters, chapter one which is an introduction, chapter two literature

review, chapter three research methodology, then chapter four analysis and chapter five Design.

Chapter 2

Literature Review

2.1. Introduction

In order to get a detailed background about Learning Management Systems (LMS), a review of several publications and projects on the topic is necessary to consider, especially to address the shortcomings, strength, or methods that have been proven effective to build a valuable tool in education. Therefore, as part of the literature, a mixture of six reports from highly experienced scholars are to be discussed. More importantly, some of their methods and findings, should prove necessary, will be evaluated in comparison to the objectives of this project. Prior to address these relevant papers in this context, it is necessary to note that the majority of the publications use a different term for LMS, Massive Open Online Course or MOOC in short. This paper will use both terms and abbreviations interchangeably.

2.2. LMS Features from Literature

[5] Experimented on the user design interface of the forums to measure the effects of the system in comparison with the learning outcomes of the students. More specifically, these scholars studied the speed of participation and engagement of more than 1000 students on a dedicated platform. Their focus is mainly to understand if forums are a contributing factor to students' engagement in general, and if a mechanism of reputation, like thumbs up, effectiveness and the accuracy of answers provided – similar to Stack Overflow and Yahoo! Answers – can promote larger engagement in the learning community. Through their experiment, they found that a reputation system similar to the above-mentioned do contribute to the engagement of students asking questions and posting feedbacks on a topic.

Therefore, since the main focus of this project is to help and engage students better virtually, through a forum, outside of the classrooms – a feature as such (even in its simplest form where students can thumbs-up a posting on the form as a positive reinforcement mechanism) should be implemented.

Another feature that has been suggested by [6] is adding a chat feature in forums. They study the difference between asynchronous learning network, where online tools such as MOOC are utilized for questions, comments, and answers, and synchronous computer communications where real online interaction, in the form of chats, are scheduled. The latter allows students to collectively be active at a specific time with their peers and teachers to engage. Their finding does not show a definitive solution whether asynchronous or synchronous method works best. The collected data showed a natural distribution where both methods seemed normal. The authors expressed their concern that further study is required to get a clearer idea as to which system might prove more effective.

In comparison to this project, adding a chat feature deems unnecessary for several reasons. First, participants should be online in order to engage in a one-to-one or many-to-many discussions. Seldom is the case where all bodies can be active at the same time. Second, having a chat feature defeats the purpose of a forum. Chats are happening in real-time while posting on forums is more relaxed and can be viewed by many. Lastly, chats contradict the effective feature discussed in [5] where users are rewarded based on their contribution.

[7] Focuses primarily on the information management side of LMS. They argue that categorizing forums based on their relevance increases the quality of the discussions and messages. Their mission is to set a range of pre-defined categories that help participants locate information easily and also write to the right audience. Anything not related to the right category are to be ignored, enforcing the inquisitor to ask in the right forum.

As for this project, categorization feature is of crucial importance. Forums should not be linked with topics that do not relate. Although the categorization feature in this project may not necessarily be a duplication of [7] where a set of pre-defined categories are to be presented to the participant, but in a form of freedom where participants can categorize their topics in the form with the dropdown list.

Similar to [7], [8] also acknowledges the importance of content categorization for larger engagement and usefulness. Their main concern is that the generic MOOCs now adays are trying to identify contents based on several techniques. One of those techniques would be “tagging” questions and/or comments in order to group them under one or more categories. In their view, this method is not useful and creates chaos in the sense that users may have a hard time locating contents. Also, there is no guarantee that users tag their contents correctly, resulting in a mixture of topics that do not relate to one another. Another method is that creating sub-forums for various topics. This, too, creates an overload and users may be discouraged to use the platform just by spending too much time until they find what they are looking for. Their solution to this issue is to use a model that relies on the linguistic identification of the contents and categorizing them in the order they reflect.

As of this project, implementing this type of model would yield a better platform to make. However, more data analysis is needed to see the use cases on how users interact with the platform. All in all, this project will look into integrating an artificial intelligence model based on the linguistic identification in the future.

Focuses more on the engagement of the LMS platforms rather than features. They try to understand how LMS platform can increases engagement between peers and professors especially in an online-led learning environment. They discovered that students usually do not initiate discussion on the forums. Professors, in this context, can play a huge role in creating an interaction between students by posting questioners, assignments, and the like. In addition, professors can call out students for their remarkable work which, in return, can increase engagement.

Although the method suggested by [9] is not a feature but only practice, contribution and initiation from lecturers and professors play an important role in the success of the platform and creating engagement between students. This LMS platform could integrate a different type of users specifically for professors and lecturers with certain additional privileges as an encouragement to have them initiate conversations.

[10] Discusses the importance of E-learning to which it can improve the quality of education, especially in higher education. The author of [6] surveyed around 120 master's students to measure the usefulness of LMS. In his

finding, more than half of the students uses LMS in order to gain academic information while the rest browse for general knowledge, on a daily basis. This, in his words, indicate that a large portion of online forums are visited for the purpose of education, if not at a higher level such as in academia.

In this project, the intention in creating a dynamic platform is purely for students and professors in the education sectors. As supported by [10], the major use-cases of online forums are for education. Therefore, having an LMS in a forum format tailored for education can be a good starting point. Later, according to future data collected from the application, a decision can be made to open the platform for the public or not.

2.3. Features of the literature

Table 2-1 Features of the literature

Source	Title	Feature/Finding
[5]	Should Your MOOC Forum Use a Reputation System?	Using a reputation system to encourage users ask meaning
[6]	A Field Study of Use of Synchronous Chat in Online Courses	full questions and give useful feedback.

[7]	Use of Categorization and Structuring of Messages in order to Organize the Discussion and Reduce Information Overload in Asynchronous Textual Communication Tools	A chatting system in a synchronous style of learning. Categorizing forums based on the relevancy of the content.
[8]	Bringing Order to Chaos in MOOC Discussion Forums with Content-Related Thread Identification	Using an advanced linguistic method to identify and organize forums based on the context.
[9]	Strategies for enhancing student interaction and immediacy in online courses	A study that identifies the success of LMS platforms with lecturers and professors becoming the core of interaction.
[10]	An Empirical Study of Online Discussion Forums by Library and Information Science Postgraduate Students using Technology Acceptance Model 3	A study that shows the use cases of forums – majority are for education.

2.4. Chapter Summary

For this project, several studies from experts in the field have been examined to understand the challenges of generic LMS forums of today and how to overcome those challenges. In order to develop a robust and useful system, the majority of features suggested by the aforementioned will be integrated in the platform.

Chapter 3

Methodology

3.1. Introduction

The purpose of this chapter is to select a suitable methodology that best describes the steps for developing the system. The methodology should have different phases that each phase gives a detail of the project. The phases will be discussed and illustrated with samples, case studies, and facts. To monitor the progress, a Gantt chart will be shown. This chart gives the view of how much time will be needed for developing such system and what are the outcomes and millstones of each group of tasks. Also, the subject of technology will be discussed. The paper will point out the minimal required technologies for the development phase as well as for the technologies that are needed from the end-users to run and work on the final product.

3.2. Methodology

In this section, we are choosing a methodology that best works for the development phase. There are different types of methodologies with different phases that each has pros and cons. The selection depends on the available set of data and the amount of knowledge about the method. Here, a descriptive methodology will be chosen as it best describes the steps to develop the system with a practical result.

3.3.Phases

3.3.1. Identify the problem

In this phase, we are focusing on the main issues. There is an issue with the existing e-learning system which has forum features. This feature has led to more couple of issues that will be addressed here. The first issue is that this forum feature on most of the existing e-learning, taking Moodle as an example, can only be opened by the instructor and per-subject or classes. Meaning, it is limited and controlled by instructors. Students' role starts after their action, not by the students themselves. The second problem is that after opening these forums about a subject will not last forever. It stays there until the end of the semester, and it is going to be deleted. Meaning, the next generation of students could not take advantage of the discussion or solutions. Another point is that forum is only accessible by the student of the same section in the same university, not all the world. Taking that, students in another university with the same issues will not be able to see the solution.

To sum up, the issues are:

1. Only teachers can open a forum, not students by themselves

2. The forum discussion will not last forever so that the next coming students takes advantage of it. Once the course deleted, the forum will be gone.

3. Not accessible for all the students, only for a class of students in a university.

3.3.2. Collect and analyze data

I designed a survey and distributed it among my classmates; they were 20 students. Inside the survey, I asked some basic questions to see what is the need for such a website to have. The result was this way:

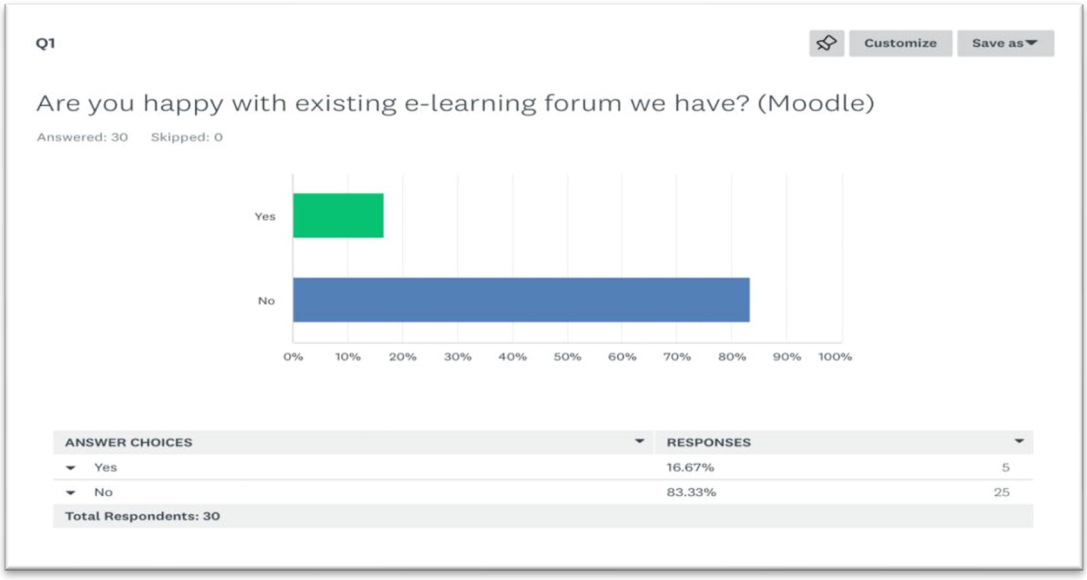


Figure 3-1 A Survey for question 1

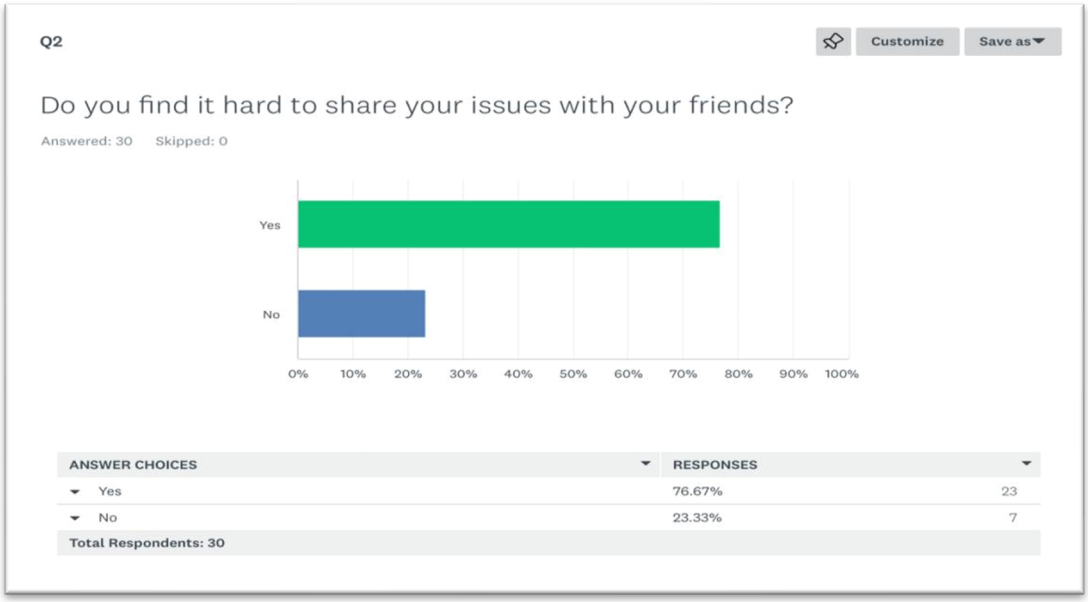


Figure 3-2 A Survey for question 2

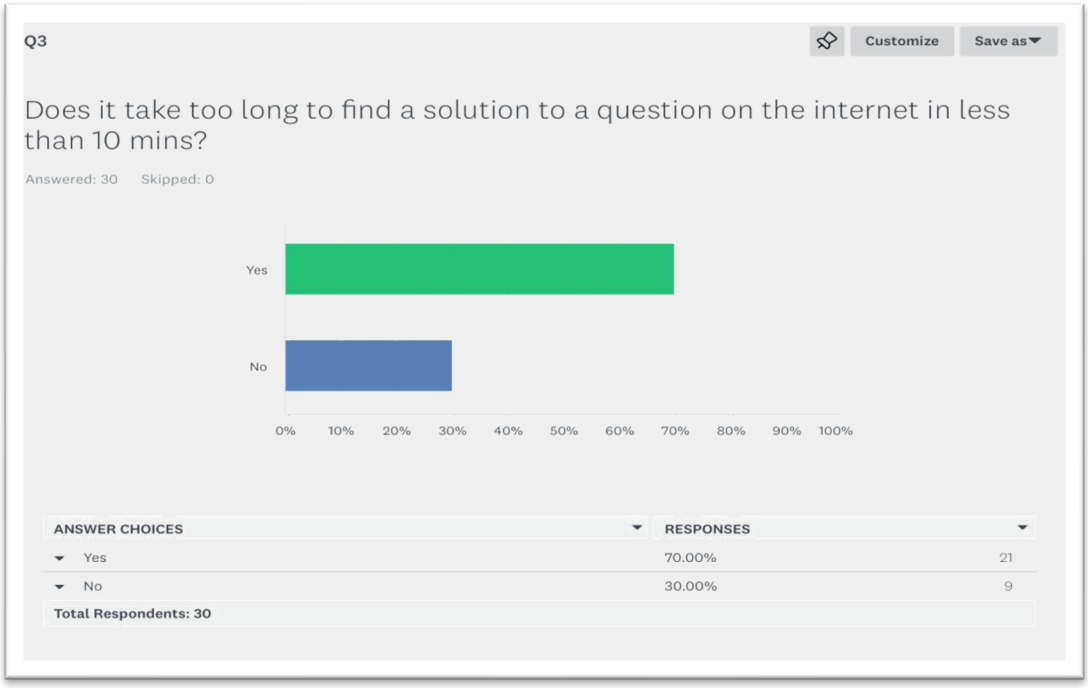


Figure 3-3 A Survey for question 3

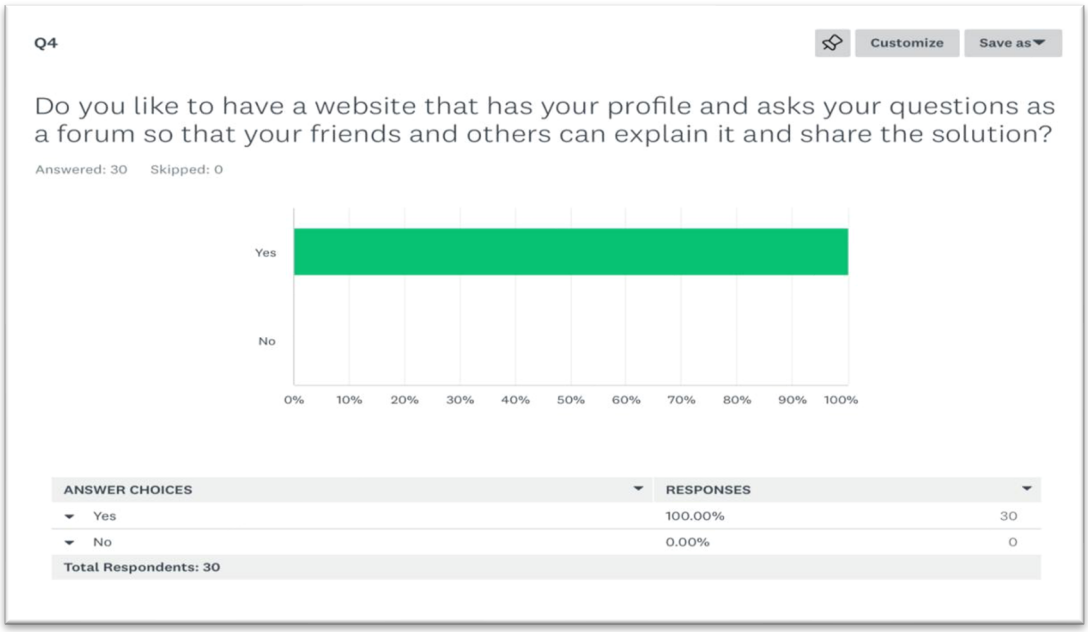


Figure 3-4 A Survey for question 4

According to this data in a small sample, it is illustrated that most of the students have a problem with finding a solution or an explanation in a faster way. It is also clear that they are not happy with the existing ones because it disappears after a while. Again, they still wonder about a website where they can share their issues and ideas.

3.3.3. Solution

Creating a website for all the students in the world regardless of their locations, universities, classes, or any categories. This will enable students to have his/her own profile that includes information about the student. On this website, the student can ask questions about any subject s/he may wish. This will lead to a discussion between the students to find a solution, or explain a topic, for a subject.

3.3.4. Gantt Chart

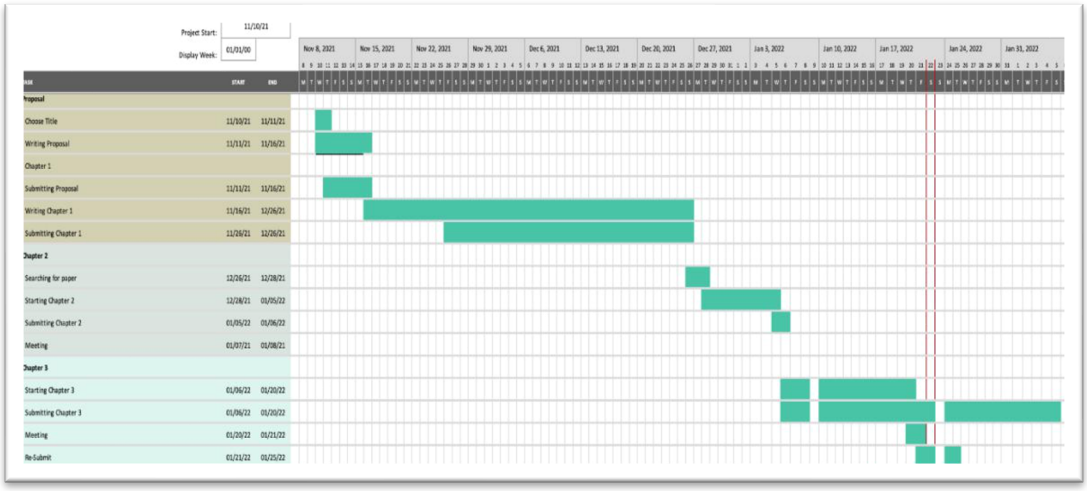


Figure 3-5 A Gantt chart explains the timeline and milestones of each submission

3.4. Technology to Develop the System

To build this system:

1. HTML: For the web content, it allows you to show data in a formatted way, such as paragraphs, links, and headings so that even non-technical people can read and engage with your product information/data.

2. CSS: For styling the content. It has the ability to control the layout of many web pages at the same time. CSS can be used to determine table cell padding, the padding around images, and color of a table's border or other objects.
3. Bootstrap: For making the website responsive and giving a friendly user interface. This works like a framework. Many ready tools and codes are prepared so that the developers do not put a lot of time into it to make it. For example, the grid systems and containers.
4. JavaScript: For some actions over the web. Allows users to create modern web applications that allow them to interact without having to reload the page every time.
5. PHP: As back-end of the website for processing data. It can be used to create interactive and dynamic web pages.
6. MySQL: The database server is MySQL. This will be used for storing all the data.

3.5. System Requirement Analysis

To run the program, it needs:

3.5.1. Software Requirements

- Sublime text: As a text editor to write the code. This would be a good option for this website as it has built-in syntax highlighting and error checker. Also, it is faster than most of the free versions of the text editors
- Xampp: As Apache server to host the website locally. This is most well-known server that holds Apache, PHP, MariaDB, MySQL. This would be the best fit for our project as it supports PHP and MySQL.

- Google Chrome: To run and check the result. It is fast, easy, and secure to use as our web browser. It also has developer's tool to check the responsiveness of our website on any platforms.

3.5.2. Hardware Requirements

Laptop: to build the web application

- CPU: Minimum core i5
- RAM: Minimum 4 GB

3.6. Chapter Summary

In conclusion, this paper is following the descriptive research method. In which, it describes the available problems and issues. Afterwards, it brings some data to illustrate the problem and clarifies the scope. Here, a survey was run among some students and collected the data. Based on the data, a solution is proposed which is a website that students make a profile and post questions or answer questions. Eventually, the requirements were shown to build and make the project work.

Chapter 4

Requirements Analysis and Design

4.1. Introduction

This chapter focuses on organizing requirements, modelling requirements and designs, validating data information, identifying data types, and the best preferred interface. The purpose of this chapter is to give a valid virtual view of the system in terms of modelling, requirements, data, and interfaces. This would be the plan of the project that will go through during the development stage to propose the final product.

4.2. Requirements Analysis

It is necessary for every project to sort out the requirements for having a functional system. In this section, all the stakeholders and functionality of the One Community will be shown and explained. This way, it makes it better to understand the theme and functionality.

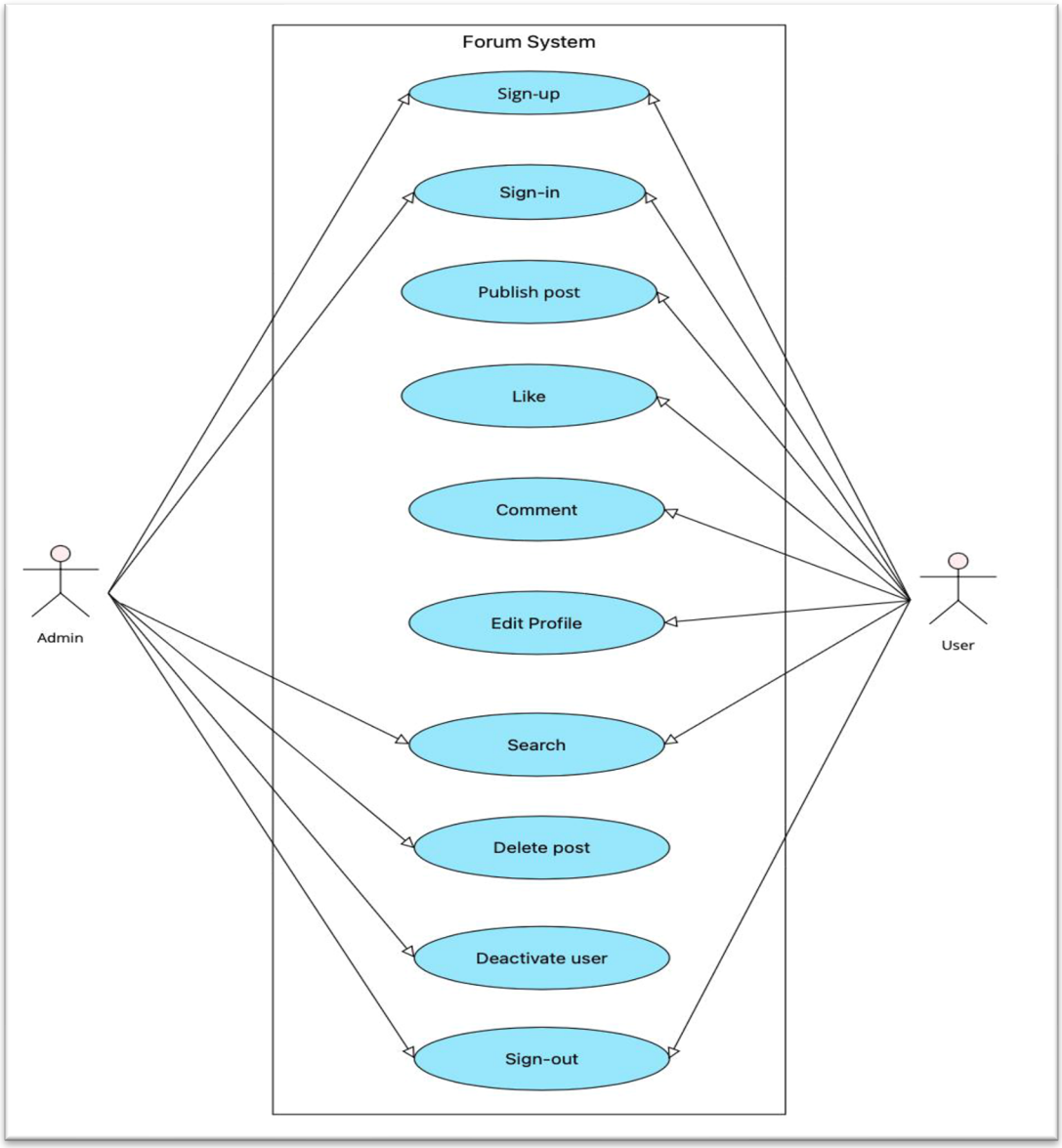


Figure 4-1 The use case of the system that explain. The functionality of each type of users

Based on this use case, there are 2 main types of users, admin and general users. Both of them have 3 mutual functionalities, which are sign up, sign in, sign out. However, for the rest they are different. An admin has the permissions for deleting a post. If a user post improper post, the admin can delete it. Also, the admin can deactivate, or block, a user. If a user behaves, the admin has the right to block his account to stop him from hurting people. In other side, a user after signing in can check the newest posts published by the community. Also, she/he can put likes and comments on the posts. Publishing a post is the main functionality of the website by the users. Editing the user information is another feature which enables the user to edited the

registered data. All of these requirements end up a website that serves a lot of students, teachers, and scholars.

4.3. Design

This section presents the design of the system. Here, all the program side possibilities get introduced.

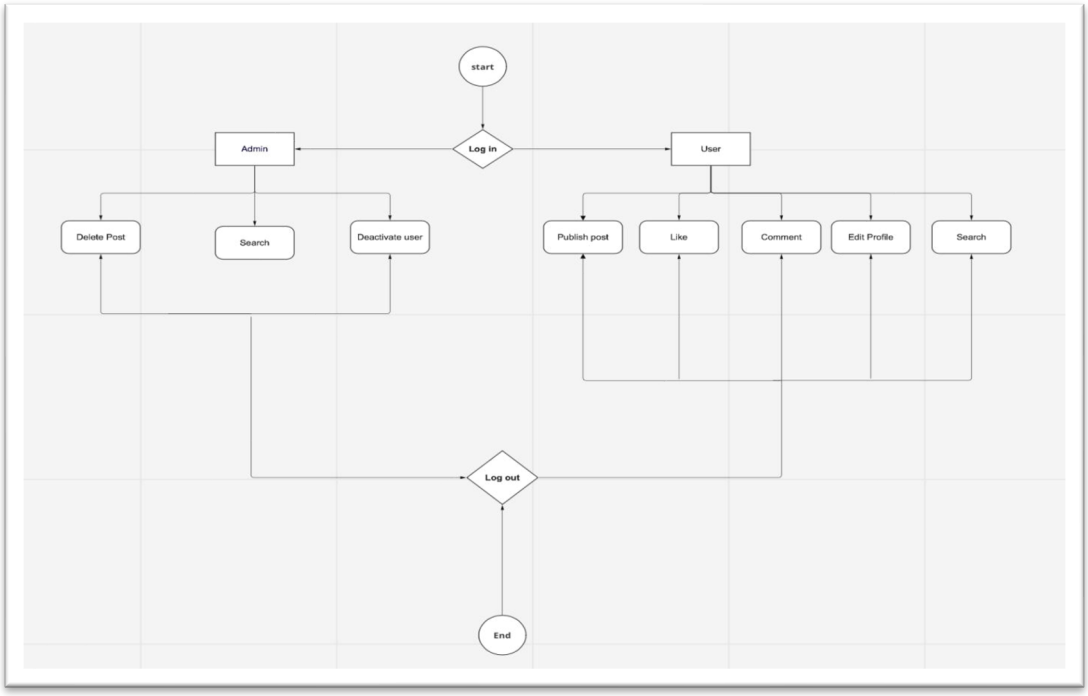


Figure 4-2 The flow chart of the One Community website that eases the process for the backend developer

Once a login happens, the system checks if the user type is admin or a general user. If it is admin, the interface of admin opens. Here the admin has two options, delete a post or deactivate a post. Finally, he can log out.

If the program checked the user type and it is not an admin type, the interface of normal users opens where the user can publish a post, like, comment on a post, or edit his profile details. Finally, he can log out of the system. All of these programming sides make up the system to function at its best.

4.4. Database Design

In this stage, the dataset design, tables, data types, and relationships will be shown. First, MySQL database management system would be used as Relational Database Management System (RDBMS). Here, there will be one database that consists of 4 tables.

4.4.1. Tables

1. User
2. Post
3. Comments
4. Category

Each of these tables hold different data to make the system functional. Here is the explanation of each table:

Table 4-1 table for user

User	
User_id	User_id
User_fname	User_fname
User_lname	User_lname
Use_bdate	Use_bdate
User_gender	User_gender
User_edu	User_edu
User_Email	User_Email
User_Password	User_Password
User_type	User_type
User_status	User_status
User_join_date	User_join_date
User_address	User_address
User_country	User_country
User_city	User_city

1- User table:

This table holds the credentials and any possible information of each user. This table will have direct connection with the Login page and other pages, like My Profile page.

User_id is the primary key of the table with the numeric data type. This ID is unique and useable for admins to work on the database based on this ID instead of names or email. Meaning, this field is required and should not be empty as well as incrementing.

User_fname holds the first name of the user. The data type is varchar as it contains text and characters. User_lname holds the last name of the user. The data type is varchar as it contains text and characters. User_bdate holds the birthdate of the user. The data type is the date.

- User_gender holds the gender of the user. The data type is varchar as it can be Male or female.
- User_edu holds the education details of the user. The data type is varchar as it can be a university name, which is long.
- User_Email is the email of the user where he has signed up with. This data is a varchar data type.
- User_Password is the encrypted password of the user who has set it from the sign-up page. Md5

function will be used to encrypt the password.

- User_type is the type of user, if it is a student, graduated, or maybe a teacher.
- User_status to check if the user is active or blocked.
- User_join_date it holds the join date of the user when signed up.
- User_country it holds the country where the user is from.

- User_city it holds the city where the user lives.
- User_address holds the address of the user.

2- Post tables:

This table holds all the information of each post that users will post on the website. This post has some information, like subject, content, date of published, and who has published it. The stored data in this table would be these:

Table 4-2 table for user

Post	
Post_id	Int(10)
P_Subject	Varchar(200)
P_Content	TEXT
P_Date	Date
User_id_Fk	int(10)
Cat_id_Fk	int(10)

- P_id is the primary key of the table with the numeric data type. This ID is unique and useable for admins to work on the database based on this ID to fetch data from the database or run queries. Meaning, this field is required and should not be empty as well as incrementing.
- P_Subject holds the subject of the post. Each post has a subject, the subject might be a word or a sentence that summarizes the content of the post.
- P_Content is the field that holds all the content of the post. This consists of a lot of sentences, or paragraphs.

The data type is TEXT which holds a string with a maximum length of 65,535 bytes.

- P_Date is the date when the post was published. The data type is the date. This is essential so that the users could know if the post is new or old by how many days, months, or years.

- User_id_Fk is the foreign key of the table. This field makes the connection between user and post tables.

This field enables the system to indicates who has published what and when.

Cat_id_Fk is another foreign key of the table. This field makes the connection between category and post tables. This field enables the system to indicates which category this post refers to.

3- Comment table

This table stores all the comments that user post on each post. Each comment has some information, like comment on which post, who has published the comment, when, and what is the content of the comment.

Table 4-3 table for user

Comment	
C_id	Int(10)
P_id_Fk	Int(10)
User_id_Fk	Int(10)
C_date	Date
C_content	TEXT

- C_id is the primary key of the table with numeric data type. This ID is unique and useable for admins to work on the database based on this ID to fetch data from the database or run queries. Meaning, this field is required and should not be empty as well as incrementing.
- P_id_Fk is the foreign key of the table. This field makes the connection between comment and post tables.

This field enables the system to indicates who has published what and when.

- User_id_Fk is the foreign key of the table. This field makes the connection between user and comment tables.

This field enables the system to indicates who has published which comment.

- C_date is the date when the comment was published on a post. The data type is date. This is essential so that the users could know if the comment is new or old by how many days, months, or years.
- C_Content is the field that holds the content of the comment. This may consist of a lot of sentences, or paragraphs. The data type is TEXT which holds a string with a maximum length of 65,535 bytes.

4- Category table

This table stores all the category that user posts about. This way, the posts will be categorized and becomes more user friendly to check the theme of the questions.

Table 4-4 table for user

Category	
Cat_id	Int(10)
Cat_name	Varchar(20)

Cat_id: is the primary key of the table with numeric data type. This ID is unique and useable for admins to work on the database based on this ID to fetch data from the database or run queries. Meaning, this field is required and should not be empty as well as incrementing.

Cat_name: This field stores the name of the category. The category might be Web development, biology, physics, math, marketing, HR, etc.

4.4.2. Table Relationships

Relationship in the database is an important aspect of the system. These relationships establish a connection between the tables that their data logically related to each other. It also helps to fetch data from multiple tables at once. It minimizes redundant data that should be avoided as much as possible by normalizing the data structure.

In this database, we have 3 relationships, which are between:

1- User and Post tables

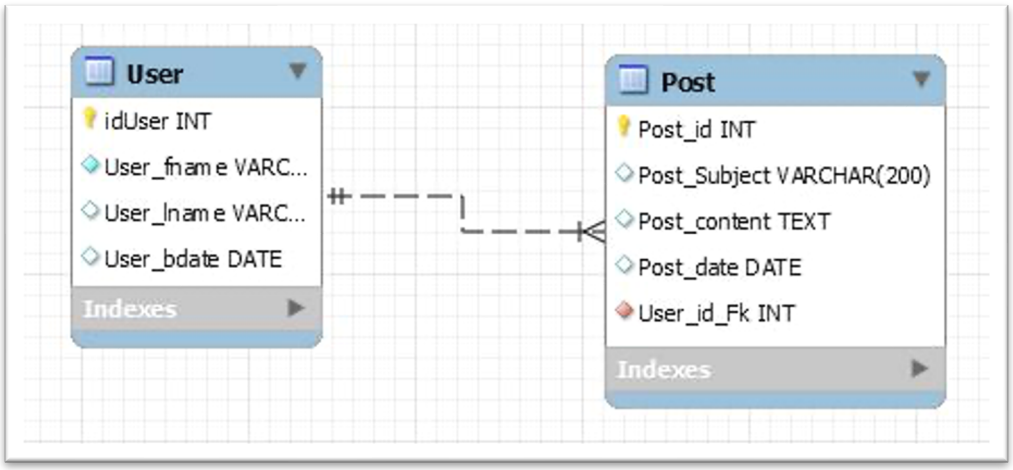


Figure 4-3 The One-to-many relationship between User table and Post table

There is a one-to-many relationship between User and Post tables. This relationship ensures that whenever a user publish a post, the post should hold the user's information about who is the user, what is the subject, what is the content of the post, and when he/she published the post. The relationship works in a way that

each post has to have at least a user id. Meaning, each post should refer to at least one person. Also, one person can have none or many posts.

2- Post to Comment tables

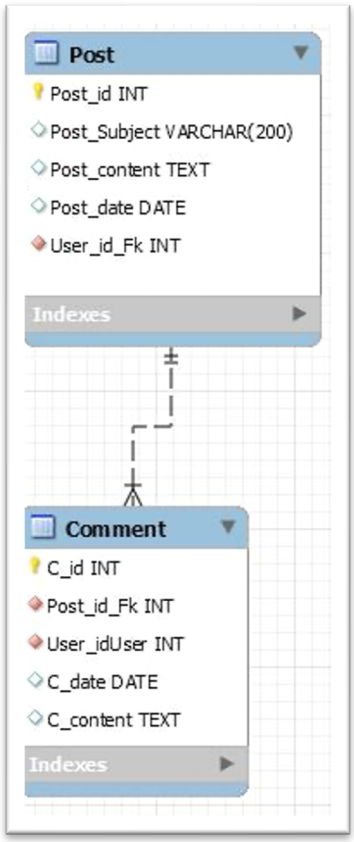


Figure 4-4 The One-to-many relationship between Comment table and Post table

There is a one-to-many relationship between Post and Comment tables. This relationship ensures that whenever there is a comment on a post, the comment should hold the post's information about on which post, what is the content of the comment, and when he/she published the comment. The relationship works in a way that each comment has to have at least a post id. Meaning, each comment should refer to at least one post. Also, a post can have none or many comments.

3- User and Comment table

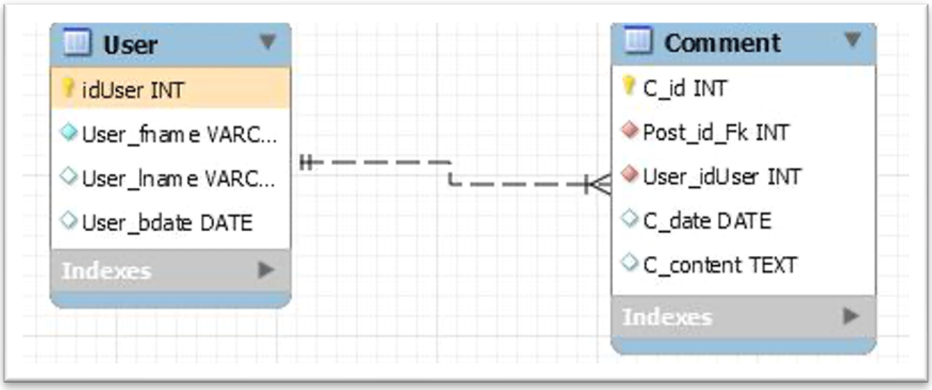


Figure 4-5 The One-to-many relationship between User table and Comment table

There is a one-to-many relationship between User and Comment tables. This relationship ensures that whenever there is a comment on a post, the comment should hold the user's information about on who is the user, what is the content of the comment, and when he/she published the comment. The relationship works in a way that each comment has to have at least a user id. Meaning, each comment should refer to at least one person. Also, a person can have none or many comments.

4- Post and Category table

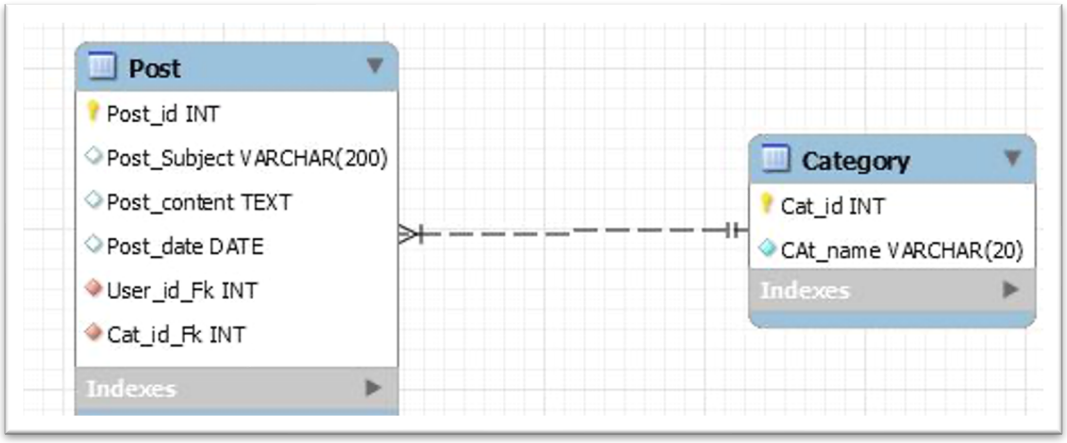


Figure 4-6 The One-to-many relationship between Post table and Category table

There is a one-to-many relationship between Post and Category tables. This relationship ensures that whenever there is a post, the post has to have a category. The relationship works in a way that each post has to have at least a category id. Meaning,

each post should refer to at least one category. Also, each category can refer to none or many posts.

4.5. Interface Design

This section needs much more focus than the other sections because this is where the final product presents. Here, the interface has been designed to let the user finds it easy and easy to follow. The overall layout, containers, navbar, foot bar, and other primary section of the website are colored with light grey and black. The reason is because this makes the user to feel comfier and healthier for eyes. Another positive side is that the users stay longer and spend more time on the website, seeking to solutions without getting annoyed due to the colors and design. Color, margins, buttons, font, and other attributes and elements are taken care of in a way that users unconsciously perform the tasks. For example, the delete and log out buttons are filled with solid red, or bordered in red, but sign in, publish, and edit buttons are filled with blue, or bordered in blue. The font is a standard font that makes it readable and easy to follow reading.

The interface design of this system is divided into three sections. The first one is the general one where everyone, admin and normal users, has to go through it and see it. The second one is for all normal users who are Student, teachers, or any. The third one is for admin users. The admin interface is different because he has more privileges and permissions on the system.

4.5.1. General Interface

This section consists of 2 main interfaces which are sign in page, where users log into the system, and the sign up, where the users start registering.

1- Sign in Page

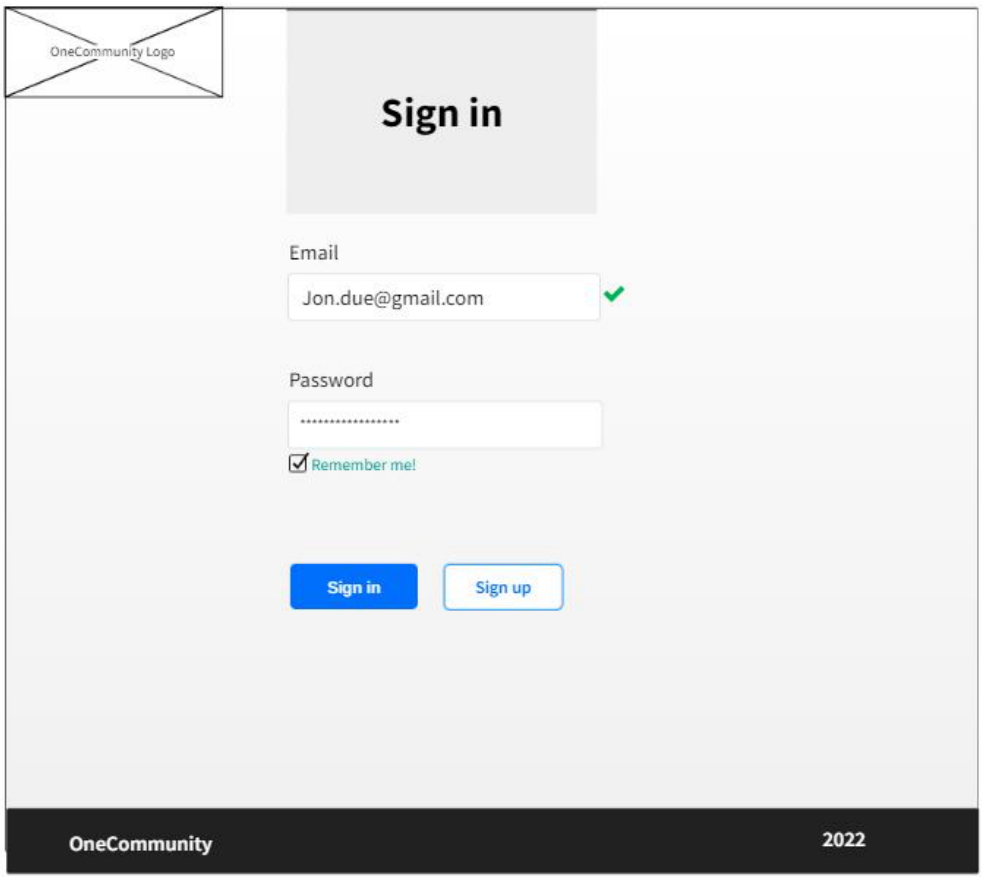


Figure 4-7 The possible front-end interface for Sign in page

The sign in page is designed as simple as the user can interact without paying too much attention to the fields and texts. The background color is light grey to make the black color appear clearly. There is a green check next to the email field which let the user to know if the email is right and registered, without mentioning in text. The green means successful, and the check means done. There is remember me check that enables the user to save his password and sessions for later. There are two buttons, sign in and sign up. The sign in is filled with blue because it means to submit, however, the sign up is empty but bordered in blue which indicates that if you are a new user and wants to register, you should fill the form.

2- Sign up Page

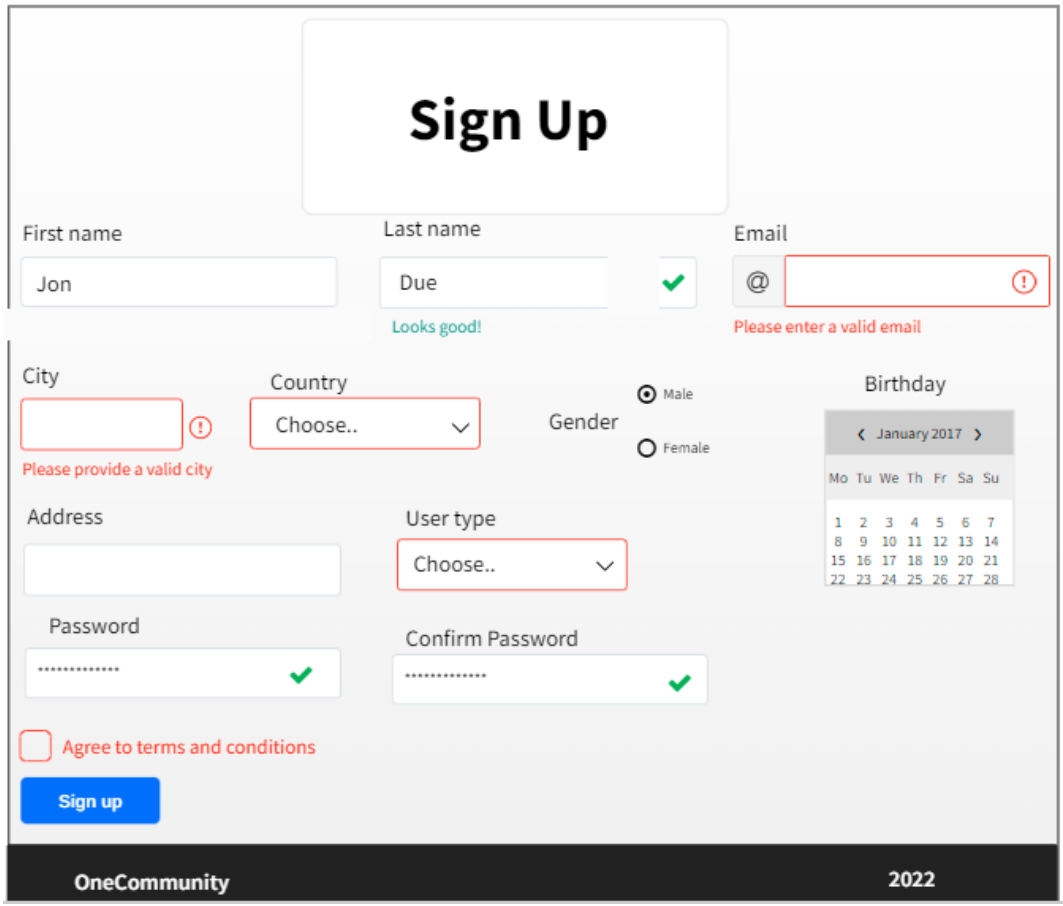


Figure 4-8 The possible front-end interface for Sign Up page

The sign-up page is designed as simple as the user easily find the fields and insert the right information. It is the best practice to minimize text fields so that users do not type everything. Here, radio buttons for gender selection, dropdown lists for country and user type selection, and ready calendar for choosing the right date for birthday are used to minimize user typing. This helps to have valid data in the right data format and type which are more useful for database and presentations. Also, there are green checks next to the email, password, and password confirmation fields which let the user to know if the entered data is right and acceptable, without mentioning in text. The green means successful, and the check means done. Some of the fields will be bordered in red in case of wrong data entry. For example, if the email is wrong or empty, or the dropdown list is not selected, the field's border will get solid red by 1 px. Last but not least, the sign-up button is filled with blue and big enough for the user to quickly make the decision and press it.

4.5.2. Standard Pages

This section consists of 4 main interfaces which appear after signing in. These pages are global as they are the same for all the users inside the system.

1- Home Page

After signing in, the first page open is the home page where the newest post shows up here. First, there is a navigation bar that separates all the pages. The bolded and underlined text is where you are as a user. The log out button is in the right side, colored red so that easy to for user to find out the log out functionality. The posts are listed in the container of the page. Each one is separated by its block and borders.

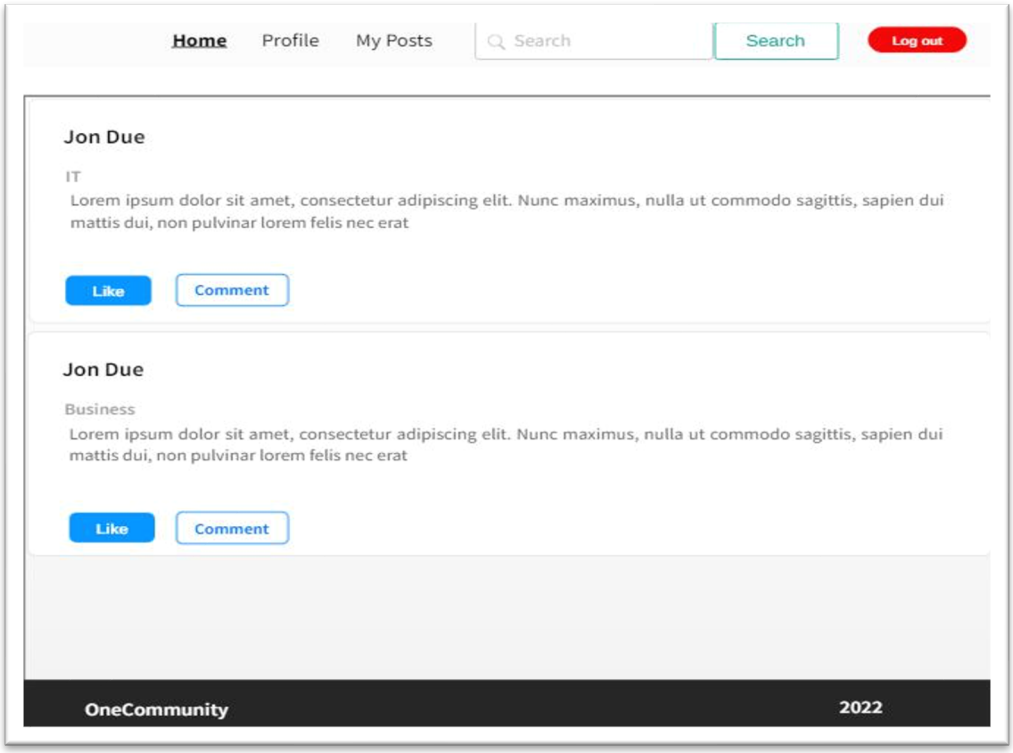


Figure 4-9 The possible front-end interface for home page for general user type

2- Profile

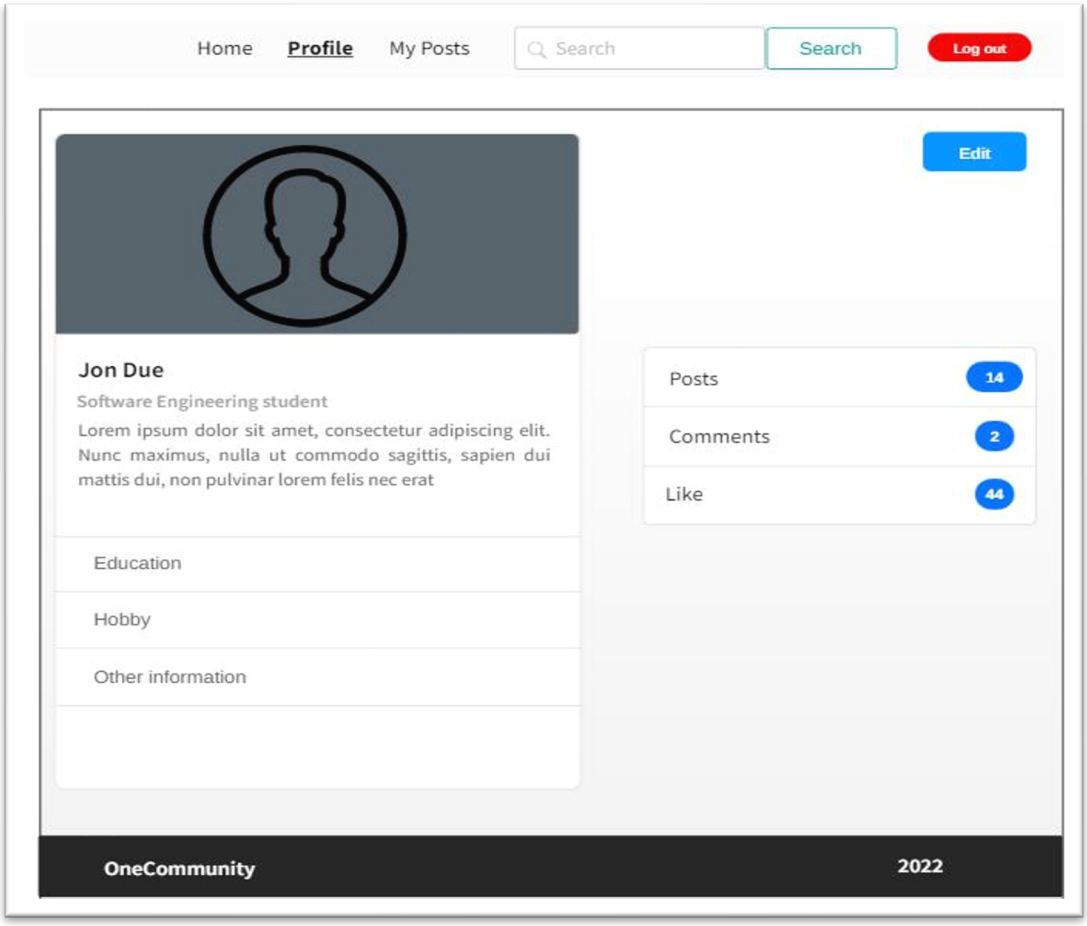


Figure 4-10 The possible front-end interface for Profile page for general user type

If Profile clicked, another page opens which is the profile of the user. This page consists of all the details about the user. The data of this page all fetch back from the User table to here. Also, the user can edit them. Also, there is a statistic in the right side, that shows the number of posts the user has made, the comments, and likes.

3- My Posts Page

This page is indicated for those post that the user has published to check for answers and comments. From here, there is a button “Publish a post”. This enables the user to publish a post. After clicking on this button, it takes the user to the next page.

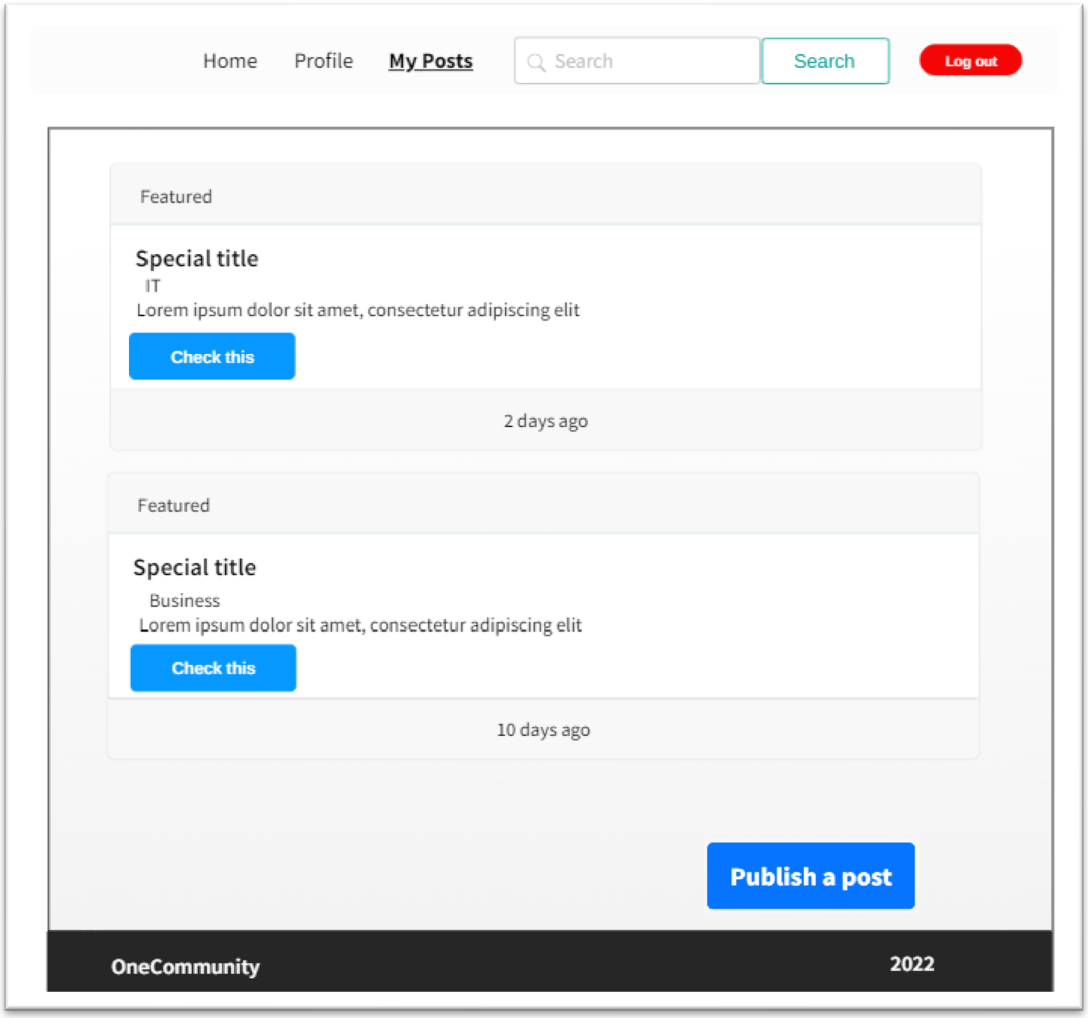


Figure 4-11 The possible front-end interface for My post page for general user type

4- Publish a post

This page is where the user publishes a post. First, there is Category dropdown list. Here, the user chooses the category of his post. For example, if it is related to website, he chooses Web Development.

The second line is the Subject. Here the user put a subject of his question. This would give a better view for the other users to expect what kinds of questions should be expected. For example, the user can say CSS for Dropdown list. The big text area is where the whole post and explanation goes. Finally, the user press Publish button to publish it.

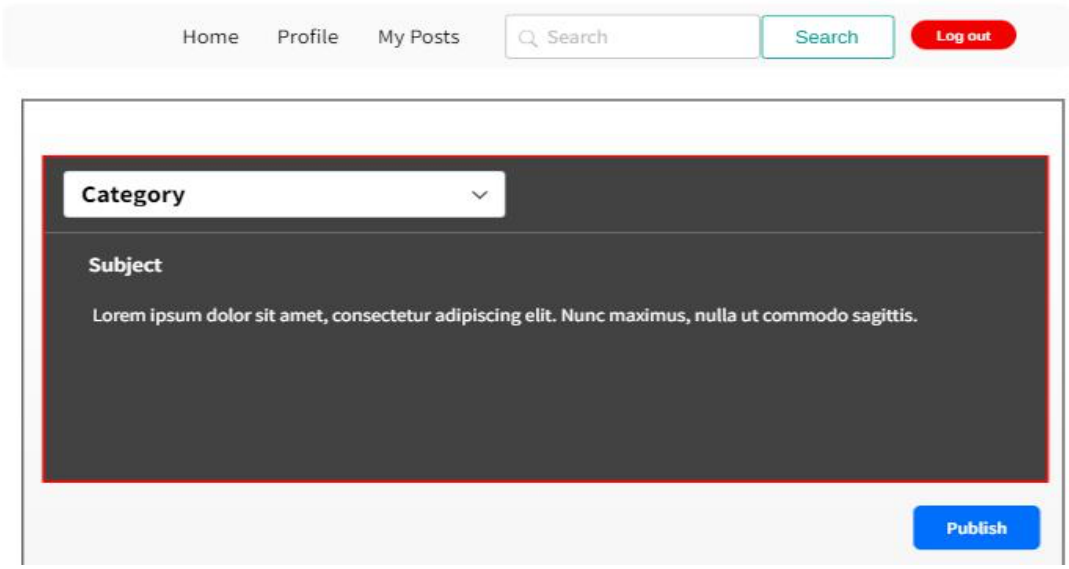


Figure 4-12 The possible front-end interface for Publish a Post page for general user type

4.5.3. Admin Interfaces

The same as most of the systems, in this system there is admin user. This user has more and special permissions that the normal users do not have it. Of course, the user interface will be different as well to enable the admins procced some more actions. First, there is a navigation bar that repeats on all the pages. The bolded and underlined text is where you are as an admin.

1- Users

Figure 4-13 The possible front-end interface for Users page for admin user type

After signing in as admin, the first page open is the User page where all the registered users show up here. There are some details about the users. An action button is there which enables the admin to deactivate or re-activate a user.

2- Posts

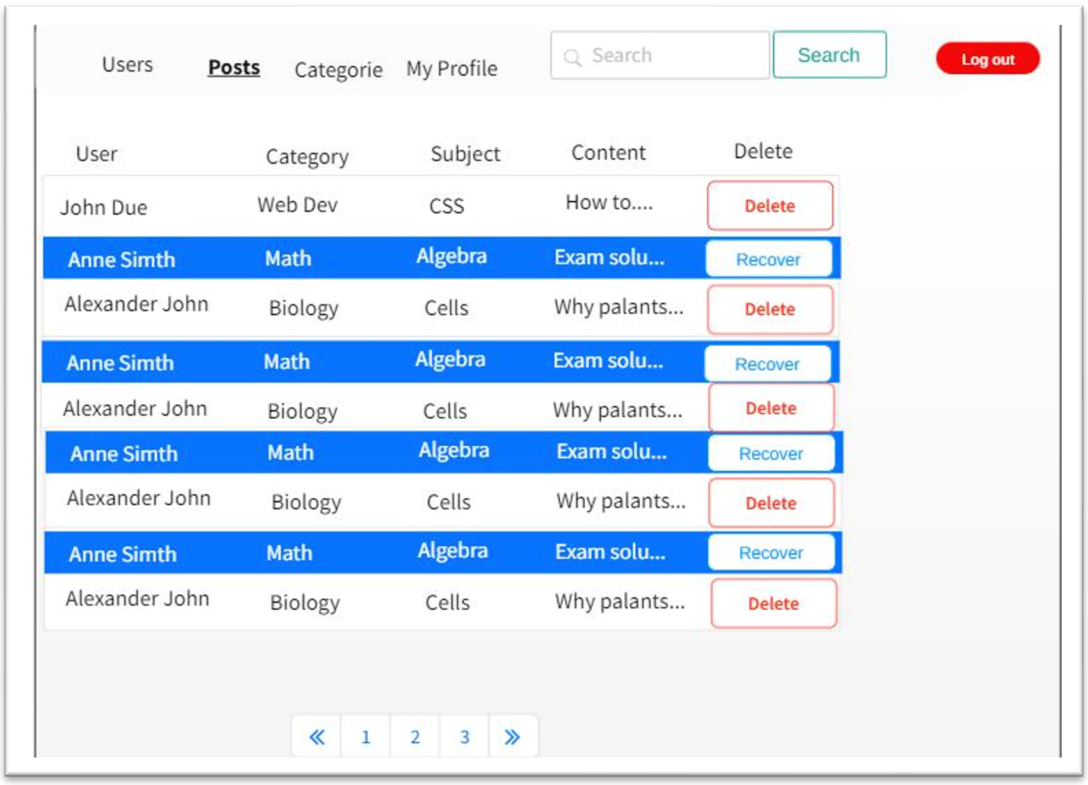


Figure 4-14 The possible front-end interface for Posts page for Admin user type

The next page is the Posts page where all the published posts by the users are listed here. Each post has some details and they are colored. For example, the first one is light grey, but the second one is blue, and this pattern repeats as even and odds. This gives a good interaction for the admin to not mixed up. Also, there is an action button which is Delete functionality. If a user post something that is not acceptable for the community, the admin has the power to delete the post. Also, at the bottom of the page there is a pager where enables the admin to go back to the old posts.

3- Categories

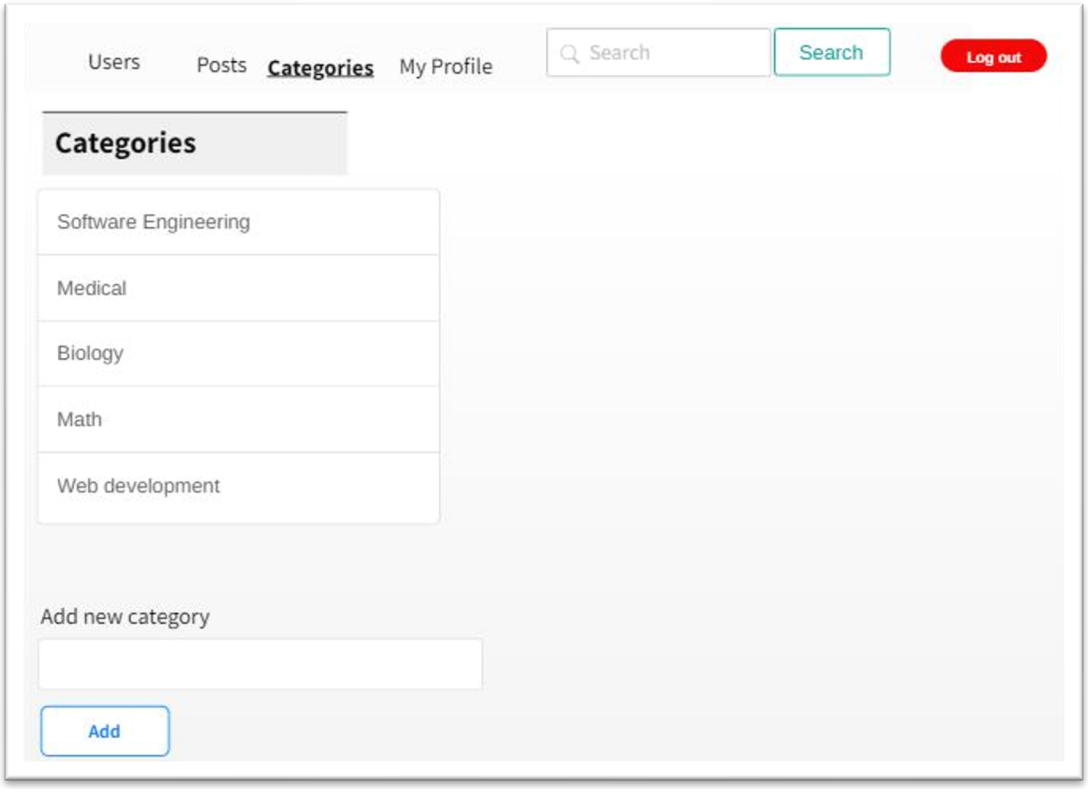


Figure 4-15 The possible front-end interface for Categories page for Admin user type

This page enables the admin to add more categories to the list. In this page all the available categories are listed. Also, there is a field where the admin types the new category, and there is an Add button once pressed, the category will be added to the list. The button is filled with white and bordered in blue.

4.6. Summary

For this project design, different steps have been taken. These steps will be practically taken once the development stage starts. First, the use case explained how the system will interact with the users and who are the expected users. It is also mentioned that the database should have some tables with specifications, types, and relationships between them. The big part is the user interfaces which divided into 3 main sections for two different user roles. In order to develop this web application, it is necessary to take these steps to function as it supposed to be.

Chapter 5

Implementation, and Testing

5.1. Introduction

This section will provide a brief explanation of the system architecture, including the architecture that was chosen as well as its features and components.

5.2. Architecture Style and Rationale

MVC stands for "Model-view-controller," meaning an application is divided into three interrelated pieces. The view, which is the screens and may be thought of as the software's interface, is what the user sees and interacts with. The link between the view and the controller is known as the model. For both online and mobile applications, the model-view-controller is frequently used.

5.3. Architecture Model

In the architecture model following table, there are two sides: the front end and the back end. At the front of the end, as we previously mentioned, we could see the view layer in the MVC, which is the interface layer that the user can view and interacts with. For this justification, we placed the website interface on the view layer. The user requests data and information or actions to interact with the controller side; in the example provided, the user can post a blog from the controller. To pass and obtain the necessary data or information, the model is connected to the database.

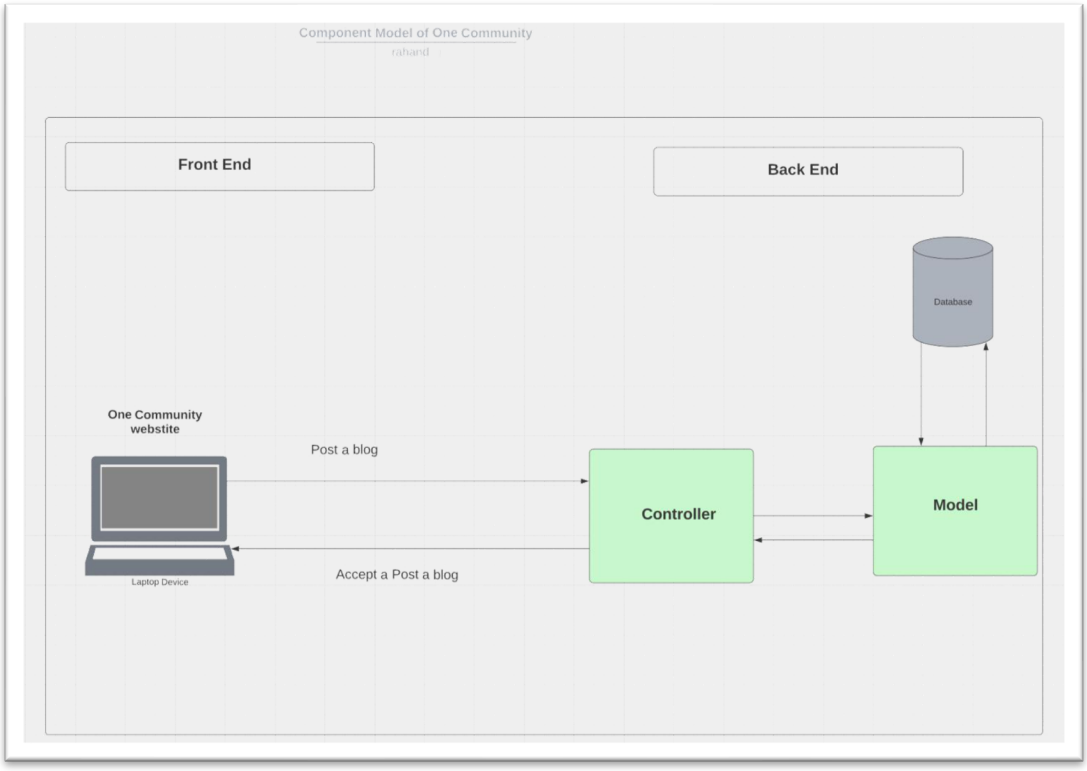


Figure 5-1 Component model of one community project

5.4. Detailed Description of Components

5.4.1. Component Diagram

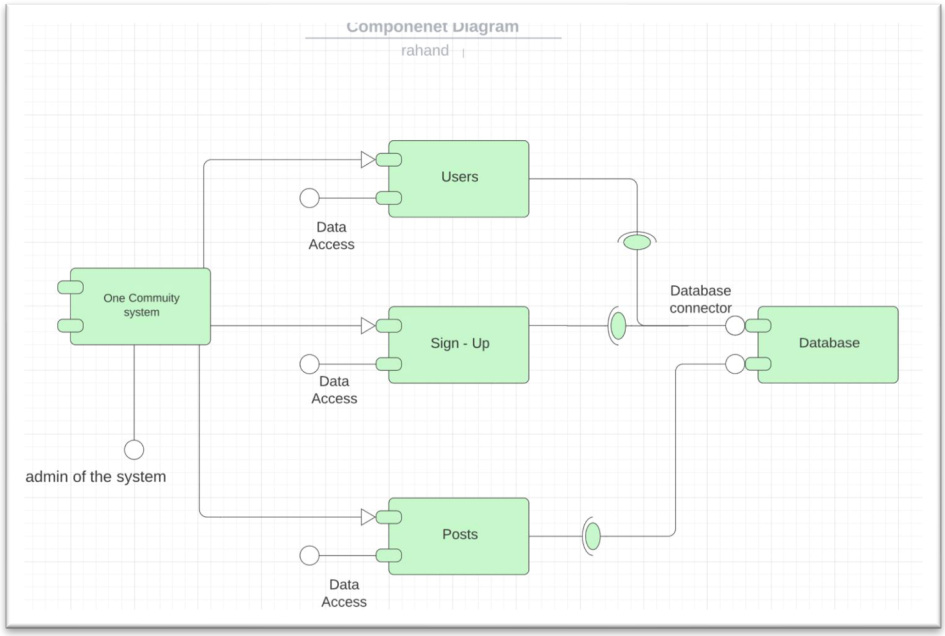


Figure 5-2 Component Diagram of One Community project

5.4.2. Detailed Description

Here is the class diagram:

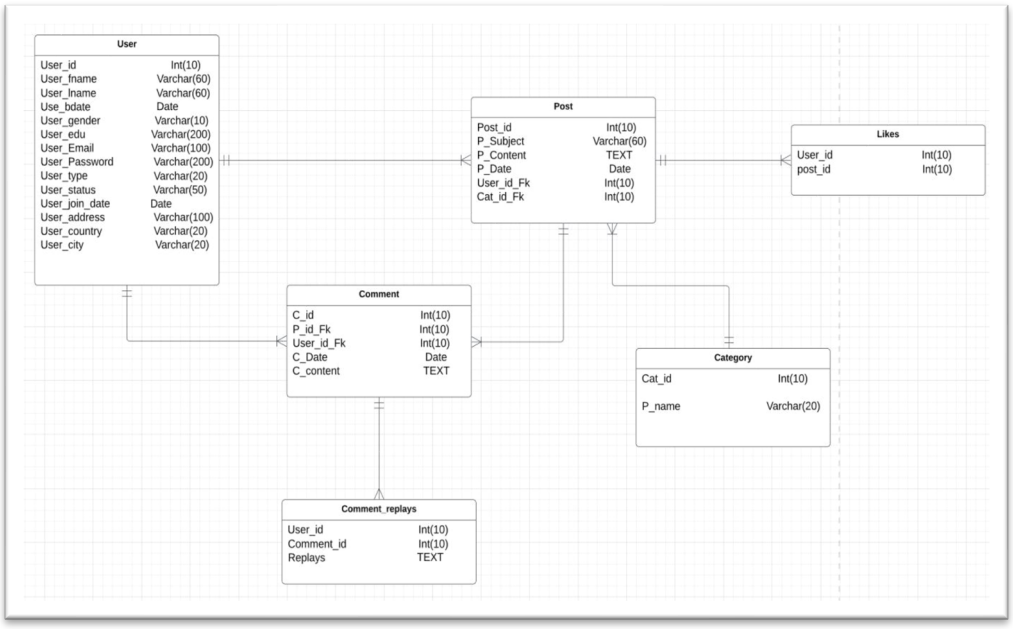


Figure 5-3 Class Diagram of One Community project

Here is the sequence diagram:

For Admin

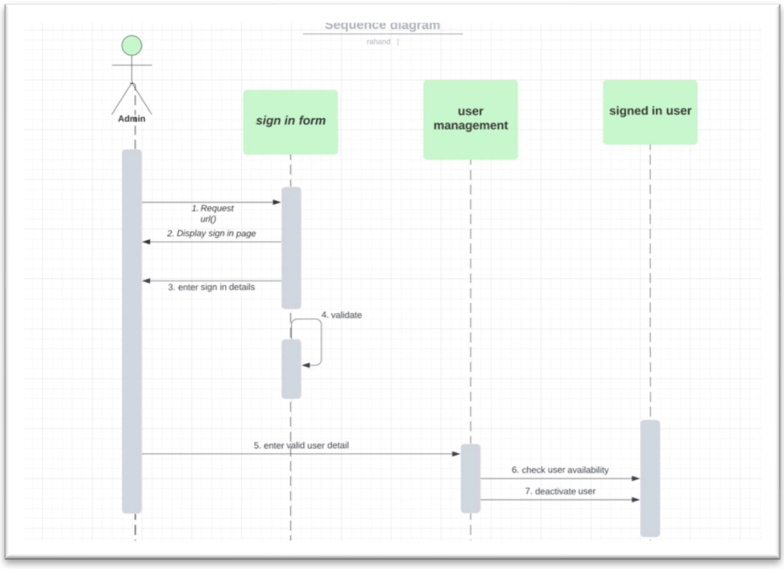


Figure 5-4 Sequence diagram for admin

For Non-admin

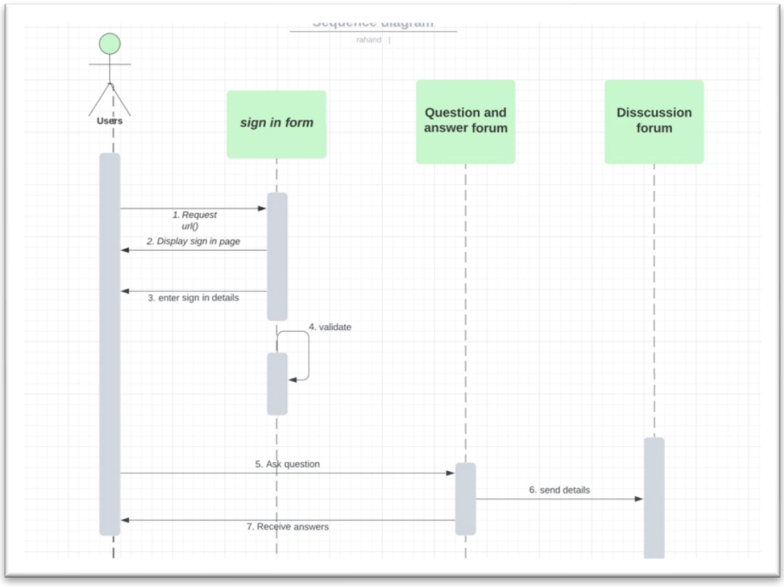


Figure 5-5 Sequence diagram for non-admin

5.5. Data Design

5.5.1. Data Dictionary

Table 5-1 Database normalization table for User Data

Name	Type	Default	Index
User_id	Int	Null	Primary key
User_fname	Varchar	Null	Not Null
User_lname	Varchar	Null	Not Null
User_bdate	Date	Null	Not Null
User_gender	Varchar	Null	Not Null
User_edu	Varchar	Null	Not Null
User_Email	Varchar	Null	Not Null
User_Password	Varchar	Null	Not Null
User_type	Varchar	Null	Not Null
User_status	Varchar	Null	Not Null
User_join_date	Date	Null	Not Null
User_address	Varchar	Null	Not Null
User_country	Varchar	Null	Not Null
User_city	Varchar	Null	Not Null

Table 5-2 Database normalization table for Post

Name	Type	Default	Index
Post_id	Int	Null	Primary key
P_Subject	Varchar	Null	Not Null
P_Content	TEXT	Null	Not Null
P_Date	Date	Null	Not Null
User_id	Int	Null	Foreign key
Cat_id	Int	Null	Foreign key

Table 5-3 Database normalization table for Comment

Name	Type	Default	Index
C_id	Int	Null	Primary key
P_id	Int	Null	Foreign key
User_id	Int	Null	Foreign key
C_date	Date	Null	Not Null
C_content	Text	Null	Not Null

Table 5-4 Database normalization table for Category

Name	Type	Default	Index
Cat_id	Int	Null	Primary Key
Cat_name	Varchar	Null	Not Null

Table 5-5 Database normalization table for Like

Name	Type	Default	Index
User_id	Int	Null	Primary Key
Post_id	Int	Null	Not Null

Table 5-6 Database normalization table for Comment_replays

Name	Type	Default	Index
User_id	Int	Null	Primary Key
Comment_id	Int	Null	Not Null
Replays	Text	Null	Not Null

Chapter 6

Conclusion

6.1. Introduction

In Conclusion the One Community project is a website that enhances the engagement of students in a way of forum style. The main objective of this website is to provide a place for all the students in the world to ask and answer questions of other students and stay forever. This would be different from most of the existing e-learnings as they are specified for students of a university, and more specifically for students of a class, not all the university. Also, their forum does not last forever as it will be deleted after finishing the course. On One Community, regardless of their locations, universities, classes, or any category, everyone can engage by posting, commenting, or liking a post that would stay forever and can be used as a reference for others.

6.2. Achievement

After reviewing some papers of experts, they agree on one point which is having a place where all the students can engage and ask meaningful questions to get a brief explanation regardless of the subject. This point can be easily found on the One Community project. As having a chatting system on an educational platform is not helpful due to the direct connection created between two students, and this does not give benefits to the community, One Community avoided having a chat system to make it a more engageable environment. Also, categorization is another feature added for the posted. This makes the users be more active and quicker in making decisions on the posts. This way users would know which post is in his domain to answer and which one is not.

6.3. Suggestions Plans

To develop this website at its best, two main things should be considered, plan and working functionality. These two would be the main core for successfully implementing the website. A good plan should be considered in a way that this project would take around 3 months to build. To start with, it should be started from database design. A good database is the backbone of any system. After designing and creating all the tables and relationships, the interface design should be sampled and tested. Later on, the backend should be developed to make the connection between the database and the interface as well as all the programming sides. Last but not least, the functionality of the website is very crucial. Testing all the pages and possibilities should be done by 3 different people and get back some feedback from them. If the feedback is valid and making scenes, based on them the website should be adjusted.

REFERENCES

M L H Khan and A Setiawan 2019 *J. Phys.: Conf. Ser.* 1375 012084

Oproiu G C 2015 A Study about Using E-learning Platform (Moodle) in University Teaching Process Procedia - Soc. Behav. Sci. 180 426–32

Kübra Çelen F, Çelik A and Seferoğlu S S 2013 Analysis of Teachers'Approaches to Distance Education Procedia -Social Behav. Sci. 83 388–92

Rajabalee BY, Santally MI, Rennie F. A study of the relationship between students' engagement and their academic performances in an eLearning environment. E-Learning and Digital Media. 2020;17(1):1-20. doi:10.1177/2042753019882567

A. Adetimirin, "An Empirical Study of Online Discussion Forums by Library and Information Science Postgraduate Students using Technology Acceptance Model 3," *Journal of Information Technology Education: Research,* vol. 14, pp. 257 - 269, 2015.

D. Coetzee, A. Fox, M. A. Hearst and B. Hartmann, "Should Your MOOC Forum Use a Reputation System?" *CSCW,* vol. 15, no. 19, pp. 1176 - 1187, 2014.

D. H. Spencer and S. R. Hiltz, "A Field Study of Use of Synchronous Chat in Online Courses," in *Proceedings of the 36th Hawaii international Conference on System Sciences*, 2002.

M. A. Gerosa, H. Fuks and C. J. P. de Lucena, "Use of Categorization and Structuring of Messages in order to Organize the Discussion and Reduce Information Overload in Asynchronous Textual Communication Tools," *IEEE,* pp. 136 - 141, 2001.

A. F. Wise, Y. Cui and J. Vytasek, "Bringing Order to Chaos in MOOC Discussion Forums with Content-Related Thread Identification," in *ACM*, Edinburugh, 2016.

R. N. Conaway, S. S. Easton and W. V. Schmidt, "Strategies for enhancing student interaction and immediacy in online courses," *Business Communication Quarterly,* vol. 68, no. 1, pp. 23 - 35, 2005.

Appendix A

SRS

Software Requirements Specification

(SRS)

Software Engineering

Project Name: One Community

Prepared by: Rahand Mahdi Ahmed

Metrics Number: QU182SCSJ057

Date: 6/22/2022

Revision Page

a. Overview

We will be explaining and going deeper into the system's detail to know more about the One Community project and see where we have gotten to so far. The part of the report is about software requirement specification, which would be mainly used to make sure that the quality, as well as the system meet the requirements specified for this project.

b. Target Audience

One Community project focuses on providing certain forum services in which the students will be able to exchange their ideas and ask for help to find solutions for a certain problem, regardless of what the subject is whether it is IT, HR or accounting.

c. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
<Current Version> Version 1	Rahand Mahdi Ahmed	The SRS of the system developed is explained inn this version	22/6/2022

Table of Contents

1 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms and Abbreviations

1.5 Overview

2 Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 User Interfaces

2.1.3 Hardware Interfaces

2.1.5 Communication Interfaces

2.1.6 Memory

2.2 Product Functions

2.3 User Characteristic

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.4 Communication Interfaces

3.2 System Features

3.2.1 Module <Users >

3.2.1.1 UC001: Use Case <Sign up >

3.2.1.2 UC002: Use Case <Sign in >

3.2.1.3 UC003: Use Case <Publish Post>

3.2.1.4 UC004: Use Case <Like>

3.2.1.5 UC005: Use Case <Comment>

3.2.1.6 UC006: Use Case <Edit profile>

3.2.1.7 UC007: Use Case <Sign out>

3.2.2 Module <Admin>

3.2.2.1 UC008: Use Case <Sign up>

3.2.2.2 UC009: Use Case <Sign in>

3.2.2.3 UC010: Use Case <Delete Post>

3.2.2.4 UC011: Use Case <Deactivate user>
3.2.2.5 UC012: Use Case <Sign out>
3.4 Design Constraints
3.6 Other Requirements

1. Introduction

The One Community project's software requirement definition will be explained in this chapter. To better understand the project's purpose and justification, more clearly understand will be used to describe it.

1.1 Purpose

The entire project of an organization is built on an SRS. It lays forth the guidelines that all development teams will stick to. It ensures that the teams are in agreement by providing crucial information to all of the teams, including development, management, quality assurance, and maintenance. Making decisions about the lifespan of their product, such as when to retire a feature, and verifying that the criteria are met are both made easier by using the SRS. Additionally, by creating an SRS, developers can minimize the time and effort needed to accomplish their objectives.

1.2 Scope

This project focuses on providing certain forum services in which the students will be able to exchange their ideas and ask for help to find solutions for a certain problem, regardless of what the subject is whether it is IT, HR or accounting. It could be a solution for an exam question. One of the limitations this project will have is the fact that it does not have newsfeed features. The student only gets to see the newest posts rather than on their own desires. The second limitation is the absence of chatting. This is due to the fact that this feature needs some programming skills in which we have not studied so far, therefore I have only used the resources that I have studied at the university.

1.3 Definitions, Acronyms and Abbreviation

Definitions of all terms, acronyms and abbreviation used are to be defined here.

SRS	Software Requirement Specification	SRS is a document that describes how the system should work.

1.4 Overview

This paper provides an explanation of the SRS for the One Community project. The essentials of what we have to do and would have needed to develop this project will essentially be shown. It allows us to determine whether or not the project has effectively met the requirements. One Community project is a web-based to provide an education for students and improve communication between students, also allow students to check the user profiles to evaluate the answers based on the background of the user, and help the students to find a solution to a problem at the university level, and this system will be accessible at any time and not be deleted and other students with the related issue can take advantage of it whenever they want This document will cover the specifics of how the project will operate as well as the requirements necessary to provide the best idea and long-term strategy for the project.

2. Overall Description

The main factors that have an effect on the product and its needs are described in this section of the SRS.

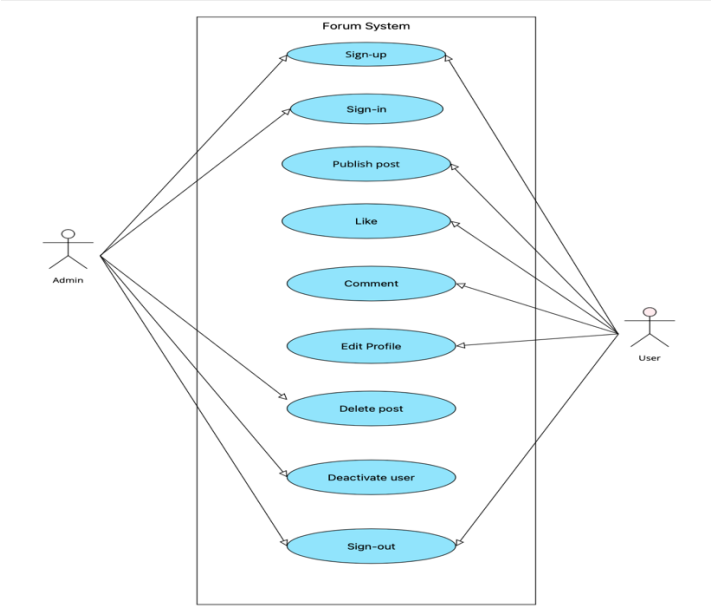


Figure 2.1: Use case diagram for <One Community> Project

2.1 Product Perspective

I decided to create this system because it allows students to submit questions and receive replies without having to sift through an inbox full of emails. Instead of searching through an inbox full of emails, you can just pose a question in the forum and view everyone's responses in one place.in this system students can post and share their own ideas, also students can post questions or thoughts, and the communicate between students. The idea of a forum made many terms much more understandable and easier to follow and improve. It is another way you learn the material through teaching it to someone else

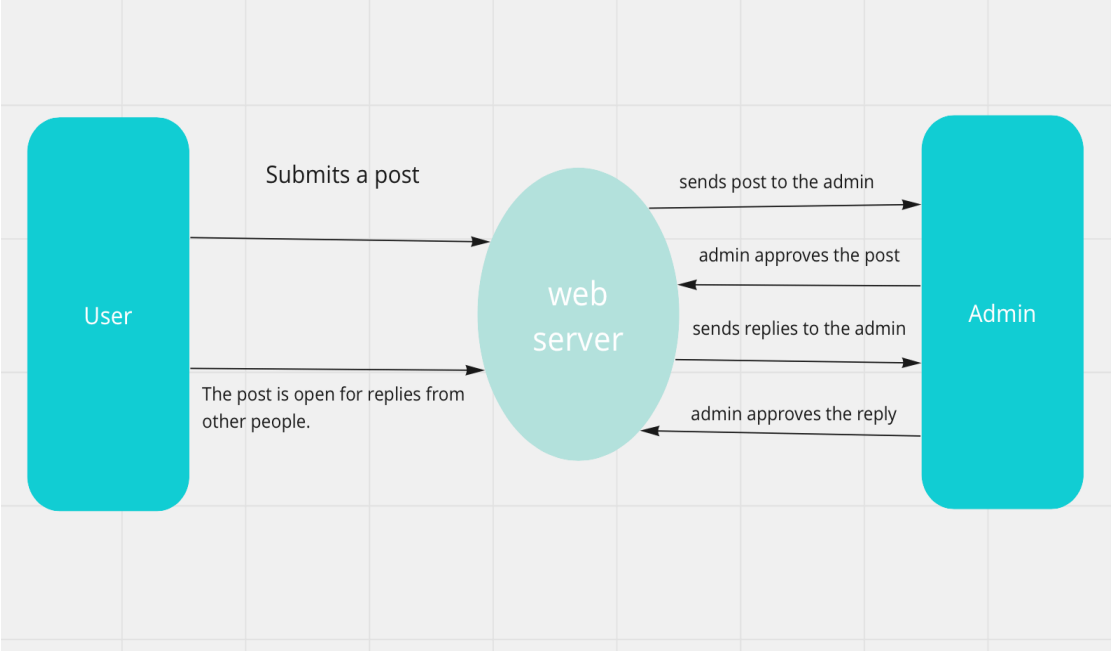


Figure 2.2: A context diagram

2.1.1 System Interfaces

In this stage, the dataset design, tables, data types, and relationships will be shown. First, MySQL database management system would be used as Relational Database Management System (RDBMS). Here, there will be one database that consists of 6 tables.

1. User
2. Post
3. Comments
4. Category
5. Likes
6. Comment_replays

Each of these tables hold different data to make the system functional

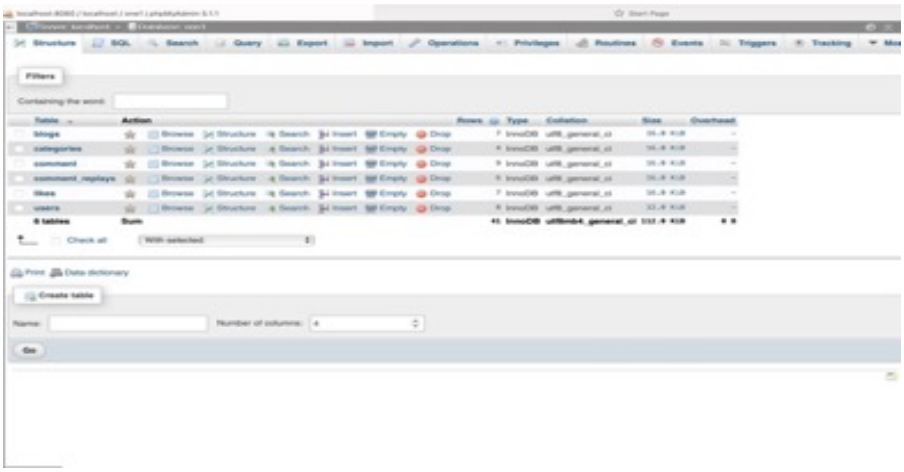


Figure 2.3: system interface for one community

2.1.2 User Interfaces

The interface design of this system is divided into three sections. The first one is the general one where everyone, admin and normal users, has to go through it and see it. The second one is for all normal users who are Student, teachers, or any. The third one is for admin users. The admin interface is different because he has more privileges and permissions on the system.

1. Sign-in Page:

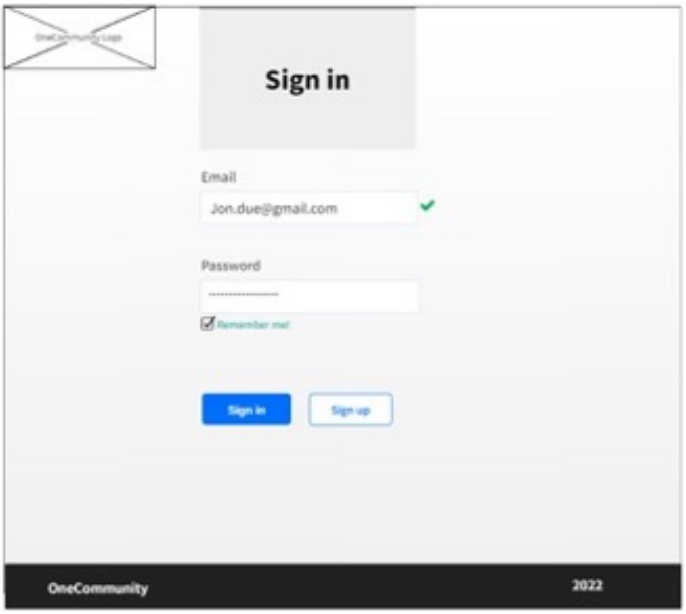


Figure 2.4: The possible front-end interface for Sign in page

2. Sign up page:

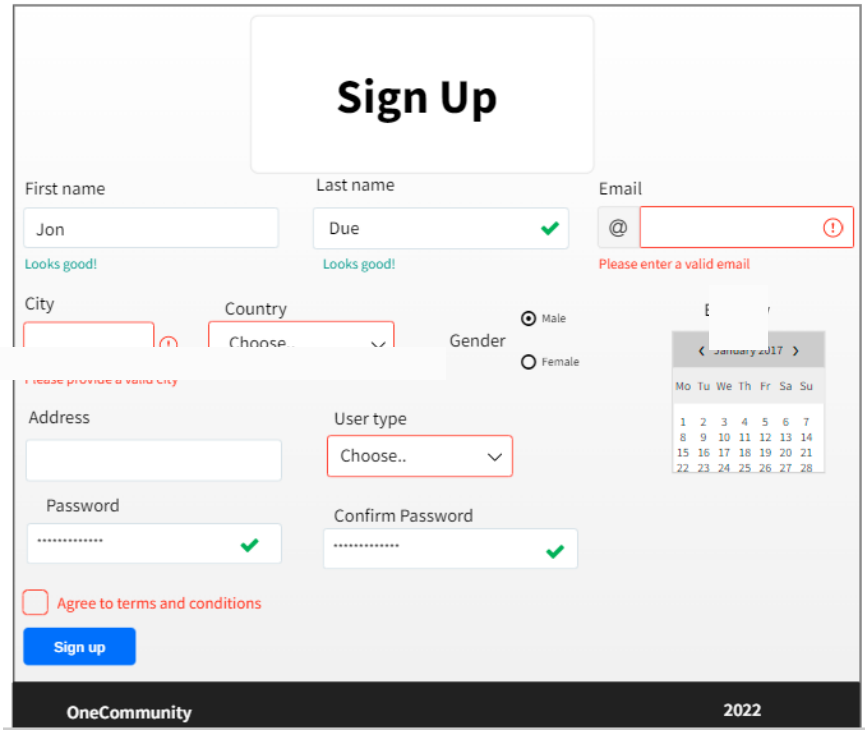


Figure 2.5: The possible front-end interface for Sign Up page

3. Home page:

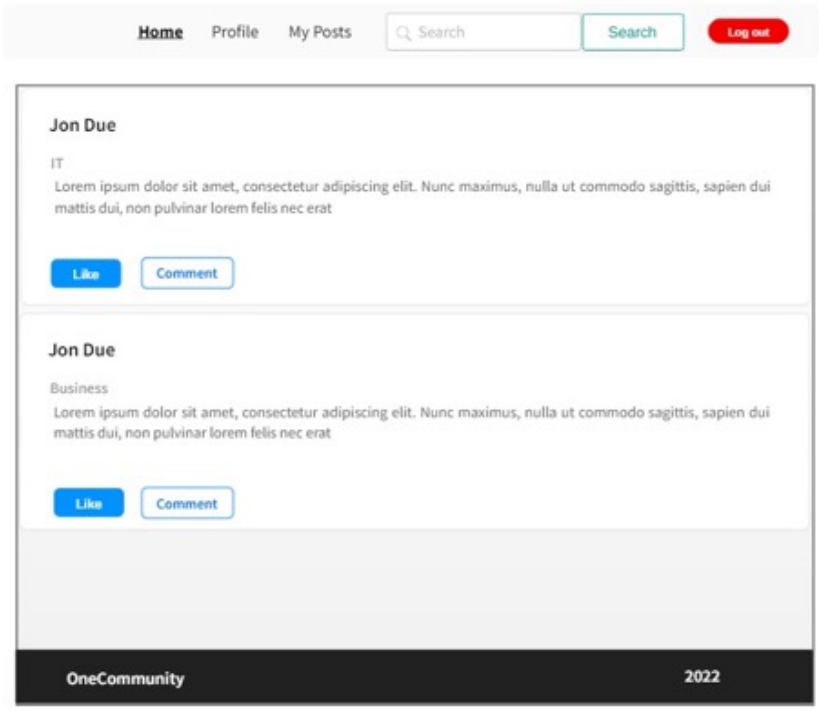


Figure 2.6: The possible front-end interface for Home page for general user type

4. Profile

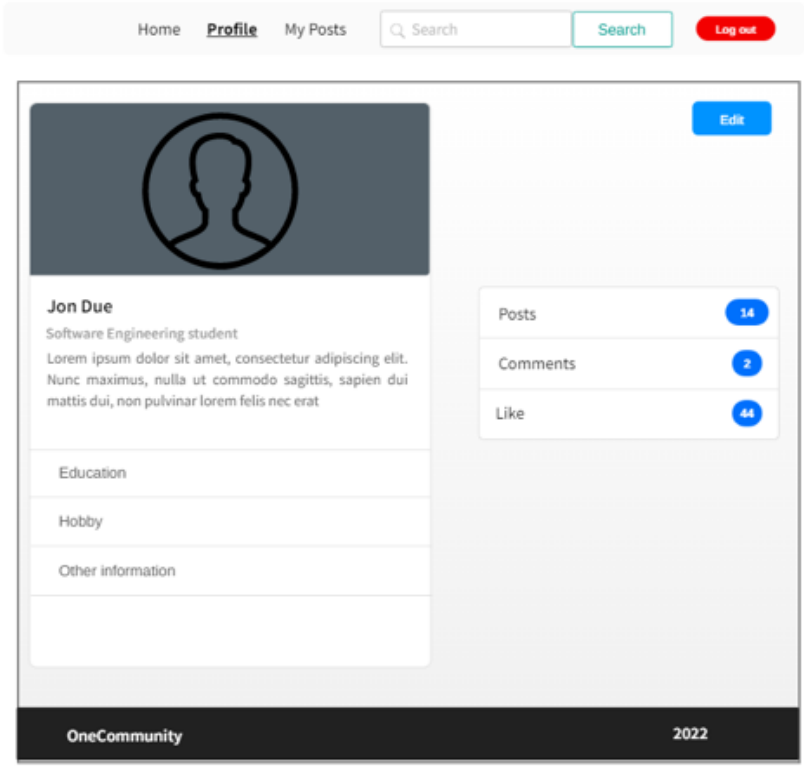


Figure 2.7: The possible front-end interface for Profile page for general user type

5. My Posts page:

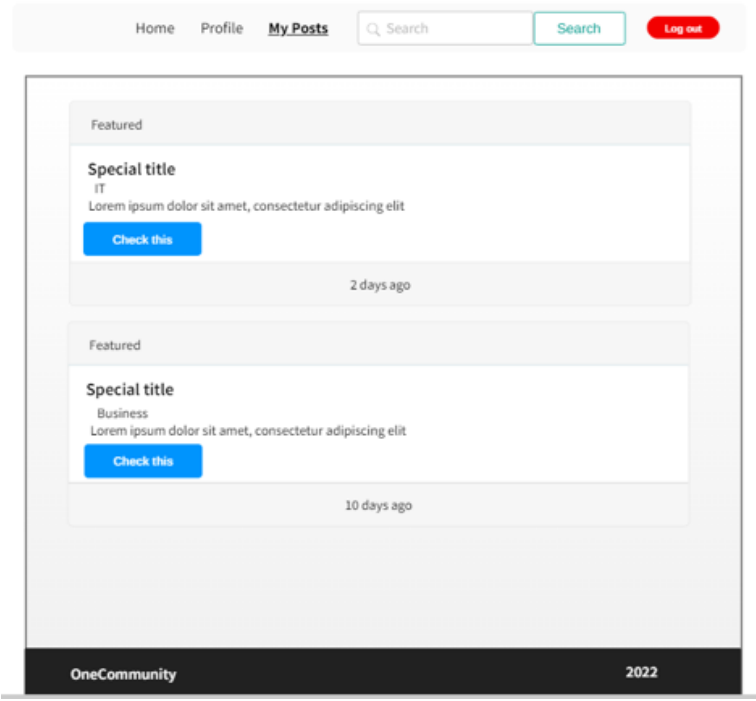


Figure 2.8: The possible front-end interface for My post page for general user type

6. Publish a post

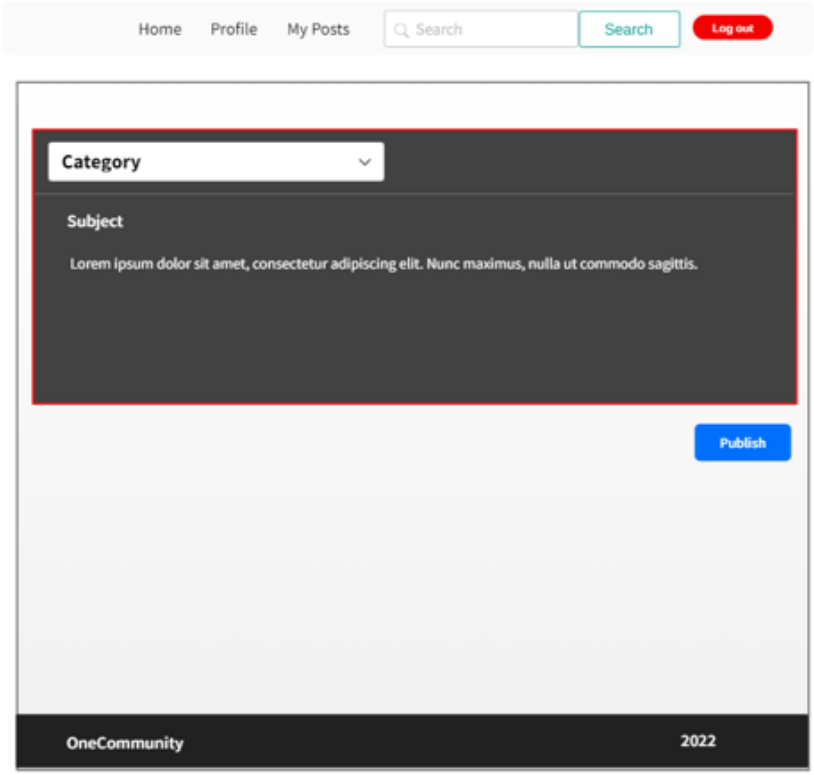


Figure 2.9: The possible front-end interface for Publish a Post page for general user type

7. Users

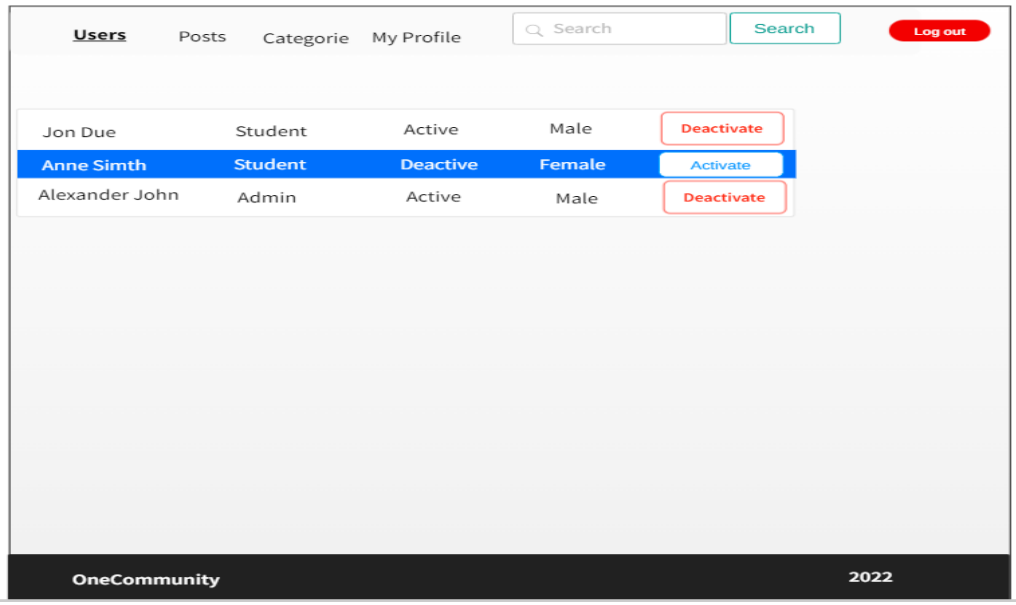


Figure 2.10: The possible front-end interface for Users page for admin user type

8. Posts:

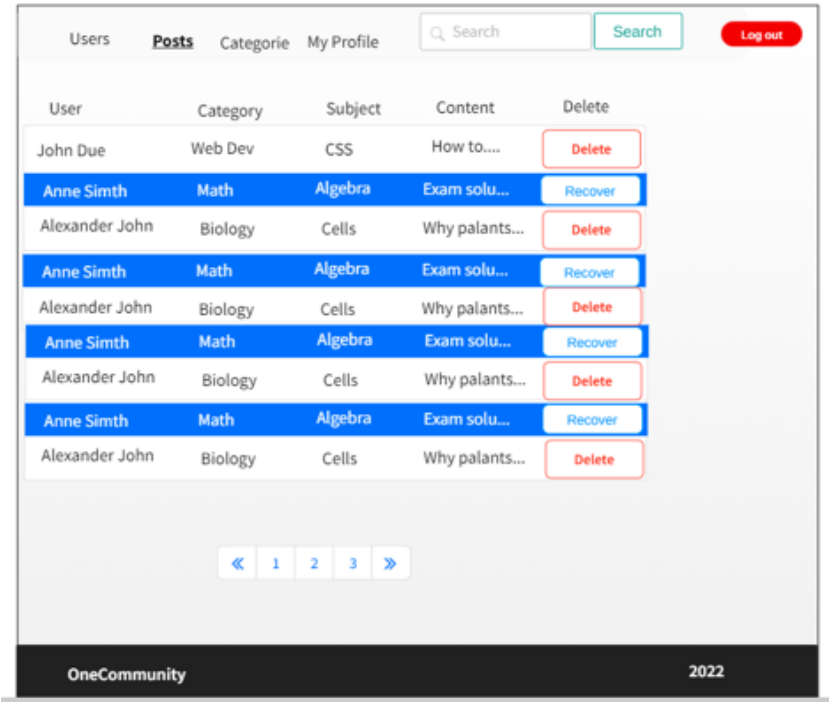


Figure 2.11 The possible front-end interface for Posts page for Admin user type

9. Categories

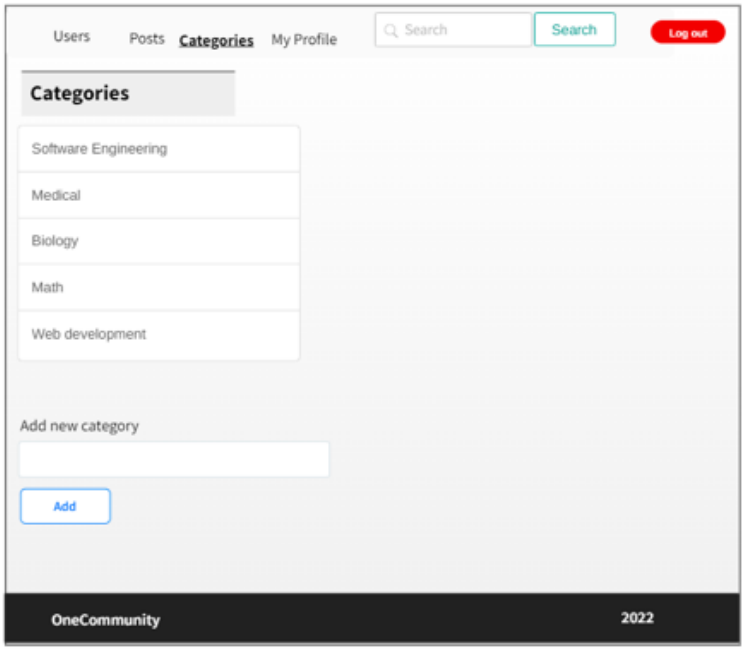


Figure 2.12: The possible front-end interface for Categories page for Admin user type

2.1.3 Hardware Interfaces

Laptop: to build the web application

2.1.4 Communication Interfaces

In order to One Community website, you need to connect the device to the Wi-Fi

2.1.5 Memory

- CPU: Minimum core i5
- RAM: Minimum 4 GB

2.2 Product Functions

Here are 9 use cases that show the key tasks carried out by the proposed system.

NO.	Use Case	Description
1	Sign-in	Users enter their email and password to sign in to the system.
2	Sign-up	Users Sign-up by filling and inserting the right information that is required.
3	Publish post	user can publish their post
4	Like	The post is open for like from other people
5	Comment	The post is open for replies from other people
6	Edit Profile	Allow user to edit their own information
7	Delete Post	Admin can delete a post
8	Deactivate User	Admin can deactivate user
9	Sign-out	Allow user and admin to sign out to the website

2.3 User Characteristics

Based on the following table we have two users in this system, each user has a specific role in this system. The three users in the system are:

1. Non-admin
2. Admin

NO.	Actor	Role
1	Non-admin	This user is able to perform all functions in the system. They can add friends, send notifications, set schedule, search and view recipes.
2	Admin	This user has the ability to deactivate user, and delete a post

3. Specific Requirements

This section of the SRS contains all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

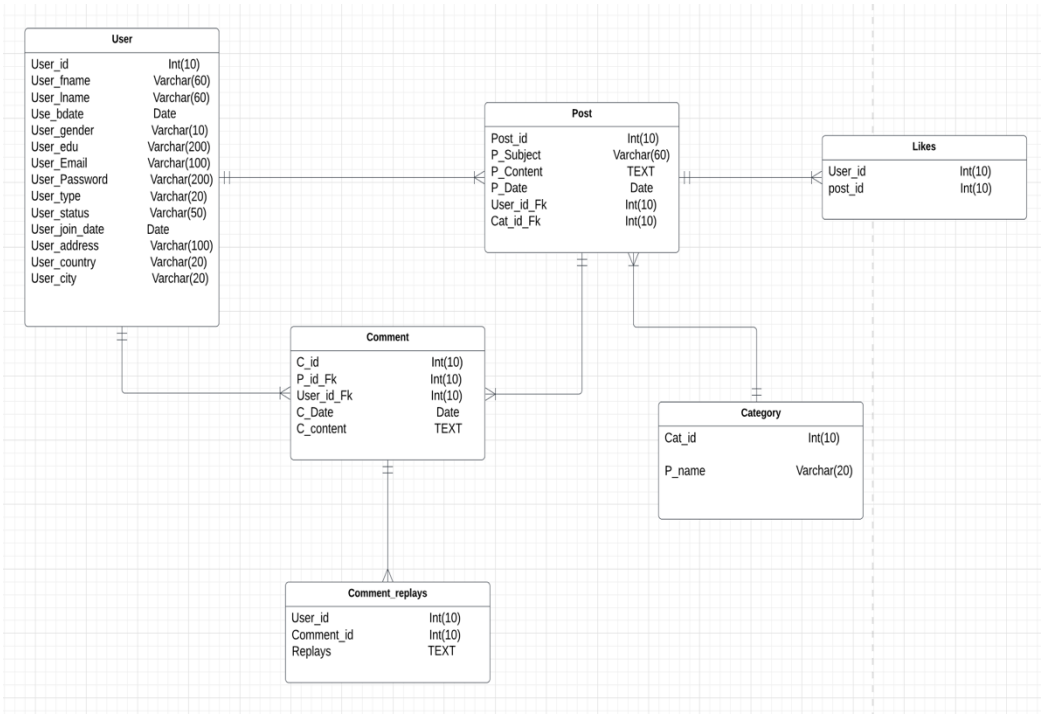


Figure 3.1: domain model for <One Community> project

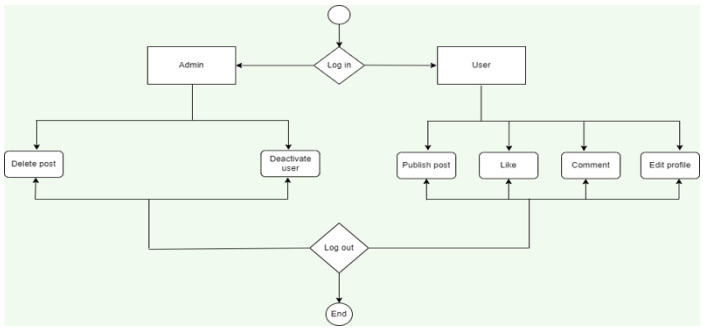


Figure 3.2: The flow chart of the OneCommunity website that eases the process for the backend developer

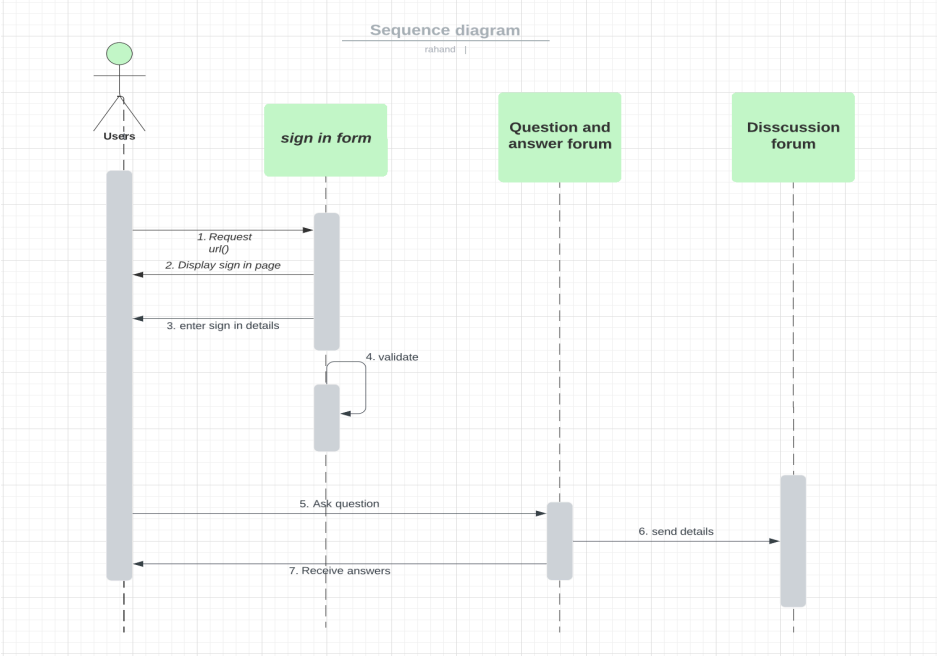


Figure 3.3: the sequence diagram for the users

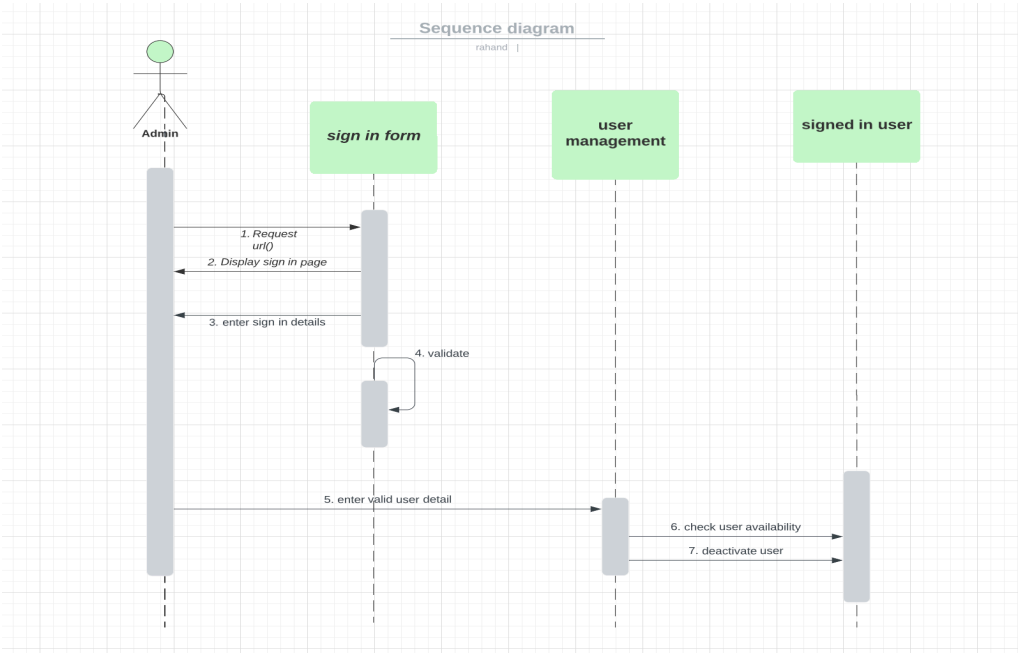


Figure 3.4: sequence diagram for admin

3.1 External Interface Requirements

3.1.1 User Interfaces

Based on this user interfaces , there are 2 main types of users, admin and general users. Both of them have 3 mutual functionalities, which are sign up, sign in, sign out. However, for the rest they are different. An admin has the permissions for

deleting a post. If a user post improper post, the admin can delete it. Also, the admin can deactivate, or block, a user. If a user behaves, the admin has the right to block his account to stop him from hurting people. In other side, a user after signing in can check the newest posts published by the community. Also, she/he can put likes and comments on the posts. Publishing a post is the main functionality of the website by the users. Editing the user information is another feature which enables the user to edited the registered data. All of these requirements end up a website that serves a lot of students, teachers, and scholars.

3.1.2 Hardware Interfaces

For the hardware interface we have said earlier that there is only one hardware interface in which the website can run on, this interface is laptop.

3.1.3 Communication Interfaces

The website requires WIFI so it can perform the actions the user needs.

3.2 System Features

3.2.1 Module < Users >

In this module, the user is able to sign-up, sign in, publish the post, like on the post, comment on other people's posts, edit their profile, and also sign out. The user module has seven use cases

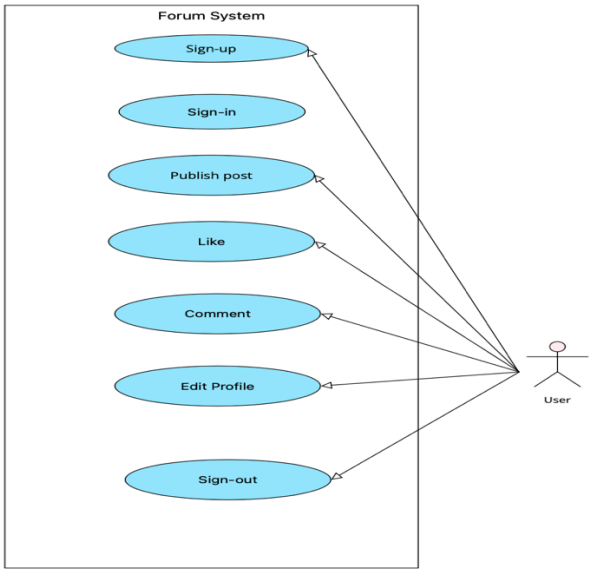


Figure 3.5: Module for <User>

3.2.1.1 UC001: Use Case <Sign up>

Table 3.1: Use Case Description for < Sign up >

Use case id: UC001	Sign-up
Brief description	The user create a new account and insert their information
Primary Actors	Users
Preconditions	Users must be sign up in the system
Postconditions	The user's profile page is shown by the system
Scenario	1. the user selects the sign up button to initiate the use case. 2. the system displays the sign - up on which users can enter their information. 3. The user's profile page is shown by the system.
Alternative Flow	When logged in with their information, an existing user can access the system.

Both sign up, as shown in the use case specification below, but they must first make sure that they do not already have that account in the system. The user must complete certain fields during signup in order to effectively complete the sign-up process.

3.2.1.2 UC002: <Sign in>

Table 3.2: Use Case Description for < Sign in>

Use case id: UC002	Sign in
Brief description	the user sign-in to the system by using their account
Primary Actors	Users
Preconditions	The user must sign-up before
Postconditions	User can sign-in to the system
Scenario	1. the system displays the sign in page that allow users to use their information 2. The database's data is checked by the system.
Alternative Flow	The user has access to the system

3.2.1.3 UC003: <Publish Post>

Table 3.3: Use Case Description for <Publish Post>

Use case id: UC003	Publish Post
Brief description	User publish their post
Primary Actors	Users
Preconditions	The user must sign-up before
Postconditions	Check answers and comments
Scenario	1. Choose category of this post 2. Put subject of his question 3. User press publish button to publish it

Alternative Flow	

3.2.1.4 UC004: <Like>

Table 3.4: Use Case Description for <Like>

Use case id: UC004	Like
Brief description	Users can like on other people post
Primary Actors	Users
Preconditions	The user must sign-up before
Postconditions	-
Scenario	When someone post the question then the other user can like on that post
Alternative Flow	-

3.2.1.5 UC005: <Comment>

Table 3.5: Use Case Description for <Comment>

Use case id: UC005	Comment
Brief description	Users can comment on other people post
Primary Actors	Users
Preconditions	The user must sign-up before
Postconditions	-
Scenario	When someone post the question then the other user can comment on that post
Alternative Flow	-

3.2.1.6 UC006: <Edit profile>

Table 3.6: Use Case Description for <Edit Profile>

Use case id: UC006	Edit Profile
Brief description	User can edit their profile
Primary Actors	Users
Preconditions	The user must sign-up before
Postconditions	User view their profile and their information
Scenario	User sign In to the system then they can edit their profile and views their information
Alternative Flow	User can see the number of posts the user has made also the comments and likes

3.2.1.7 UC007: <Sign out>

Table 3.7: Use Case Description for<Sign out >

Use case id: UC007	Sign out
Brief description	Allow users to sign out to the website
Primary Actors	Users
Preconditions	-
Postconditions	user will be sign out of the website
Scenario	User click on sig out button and the system and the user is signed out
Alternative Flow	

3.2.2 Module <Admin>

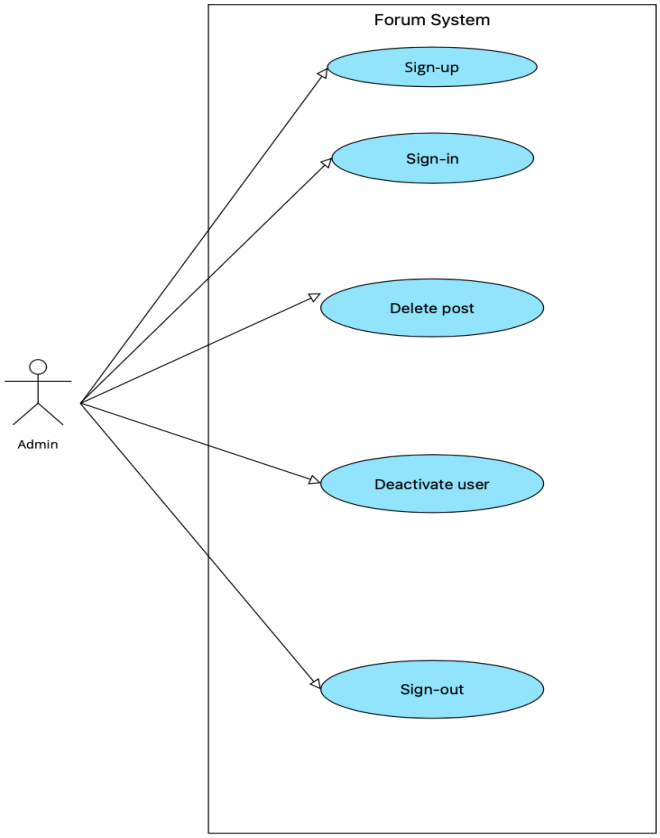


Figure 3.6: Module for <Admin>

3.2.2.1 UC008: Use Case <Sign up>

Use case id: UC008	Sign-up
Brief description	The user create a new account and insert their information
Primary Actors	Admin
Preconditions	Admin must be sign up in the system
Postconditions	The admin's profile page is shown by the system
Scenario	1. the admin selects the sign up button to initiate the use case. 2. the system displays the sign - up on which admin can enter their information. 3. The admin's profile page is shown by the system.
Alternative Flow	When logged in with their information, an existing admin can access the system.

3.2.2.2 UC009: <sign in>

Table 3.9: Use Case Description for < sign in>

Use case id: UC009	Sign in
Brief description	the admin sign-in to the system by using their account
Primary Actors	Admin
Preconditions	The admin must sign-up before
Postconditions	Admin can sign-in to the system
Scenario	1. the system displays the sign in page that allow admin to use their information 2. The database's data is checked by the system.
Alternative Flow	The admin has access to the system

3.2.2.3 UC010: <Delete Post>

Table 3.10: Use Case Description for <Delete post**>**

Use case id: UC010	Delete post
Brief description	Allows admin to delete a post
Primary Actors	Admin
Preconditions	The admin must sign-up before
Postconditions	The post is deleted
Scenario	If a user post something that is not acceptable for community the admin can delete the post
Alternative Flow	The system deletes the post from database

3.2.2.4 UC011: <Deactivate User >

Table 3.11: Use Case Description for <Deactivate User**>**

Use case id: UC011	Deactivate User
Brief description	Allows admin to deactivate user
Primary Actors	Admin
Preconditions	The admin must sign-up before
Postconditions	The user is deactivated
Scenario	If a user post something that is not acceptable for community the admin can deactivate user
Alternative Flow	The system deactivate the user from database

3.2.2.5 UC012: <sign out>

Table 3.12: Use Case Description for<sign out>

Use case id: UC012	Sign out
Brief description	Allow admin to sign out to the website
Primary Actors	Admin
Preconditions	-
Postconditions	Admin will be sign out of the website
Scenario	Admin click on sig out button and the system and the user is signed out
Alternative Flow	-

3.3 Design Constraints

- The user interface needs: There is nothing difficult about the user interface, which is why I thought about creating a simple design that everyone could use.
- The programming language used: To build this system I use HTML, CSS, bootsrap, JavaScript, PHP, and MySQL.
- Execution environment: As we already said this website can run on a laptop.

3.4 Other Requirements

This project can fulfil several significant requirements and features, from first and most crucial of which is:

1. Reliability: Any product must also be dependable or accessible to the customer at all times.

2. Portability: Any system must have the capacity to be moved about and used whenever and however necessary.

3. Maintainability: The project must be able to be maintained by either adding new features or addressing flaws or defects that were previously present.

4. Usability: It is crucial to design a product that

Appendix B

SDD

SCSJ3323: Software Design and Architecture

Software Design Document

(SDD)

Software Engineering

Project Name: One Community

Prepared by: Rahand Mahdi Ahmed

Metrics Number: QU182SCSJ057

Date: 6/22/2022

Revision Page

d. Overview

This paper will provide more diagrams to help illustrate the project in considerable detail and help readers understand how the system is separated.

e. Target Audience

The developers and software designers that create software projects are the intended audience for this document.

f. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
<Current Version> Version 1	Rahand Mahdi Ahmed	This version explains the software design documentation.	22/6/2022

Table of Contents

1 Introduction
- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms and Abbreviations
- 1.5 System Overview

2 System Architectural Design
- 2.1 Architectural Style and Rationale
- 2.2 Architecture Model
- 2.3 Use Case Diagram

3 **Detailed Description of Modules**
- 3.1 Component Package Diagram
- 3.2 Modules Detailed Descriptions
  - 3.2.1 Module <Name of Module 1>
    - 3.2.1.2 Class Diagram
    - 3.2.1.3 Sequence Diagrams

4 **Data Design**
- 4.2 Data Dictionary

5 **User Interface Design**
- 5.1 Overview of User Interface
- 5.2 Screen Images

1. Introduction

1.1 Purpose

The goal of the Software Design Documentation (SDD) is to give readers a clear understanding of how the One Community website will be constructed. This document will also describe the architectural style, and interface design. and database description.

1.2 Scope

The aim of this project is that it allows the forums to be accessible even after classes get deleted and the students will be able to go back to it even if the course is finished.

Another aim is that it allows other students, from any part of the world, to seek the solution to the same issues that has faced other students as well with suitable solutions.

1.3 Definitions, Acronyms and Abbreviation

Abbreviation	Definition
SDD	Software Design Documentation

1.4 Overview

This article covers the software design documentation for the One Community project. It describes the software developed to make decision-making, planning, and analysis easier. The following is how this documentation is set up:
System architectural design and component descriptions in detail
Designing data and user interfaces

2. System Architectural Design

This section will provide a brief explanation of the system architecture, including the architecture that was chosen as well as its features and components.

2.1 Architecture Style and Rationale

MVC stands for "Model-view-controller," meaning an application is divided into three interrelated pieces. The view, which is the screens and may be thought of as the software's interface, is what the user sees and interacts with. The link between the view and the controller is known as the model. For both online and mobile applications, the model-view-controller is frequently used.

2.2 Architecture Model

In the architecture model following table, there are two sides: the front end and the back end. At the front of the end, as we previously mentioned, we could see the view layer in the MVC, which is the interface layer that the user can view and interacts with. For this justification, we placed the website interface on the view layer. The user requests data and information or actions to interact with the controller side; in the example provided, the user can post a blog from the controller. To pass and obtain the necessary data or information, the model is connected to the database.

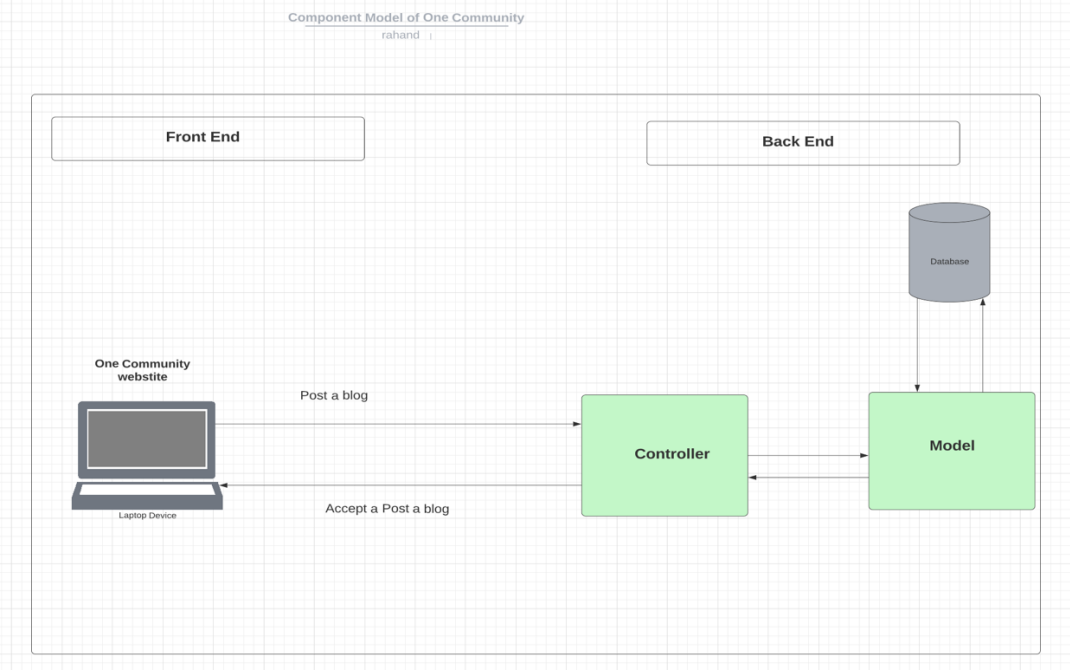


Figure 2.3: Component model of one community project

2.3 Use Case Diagram

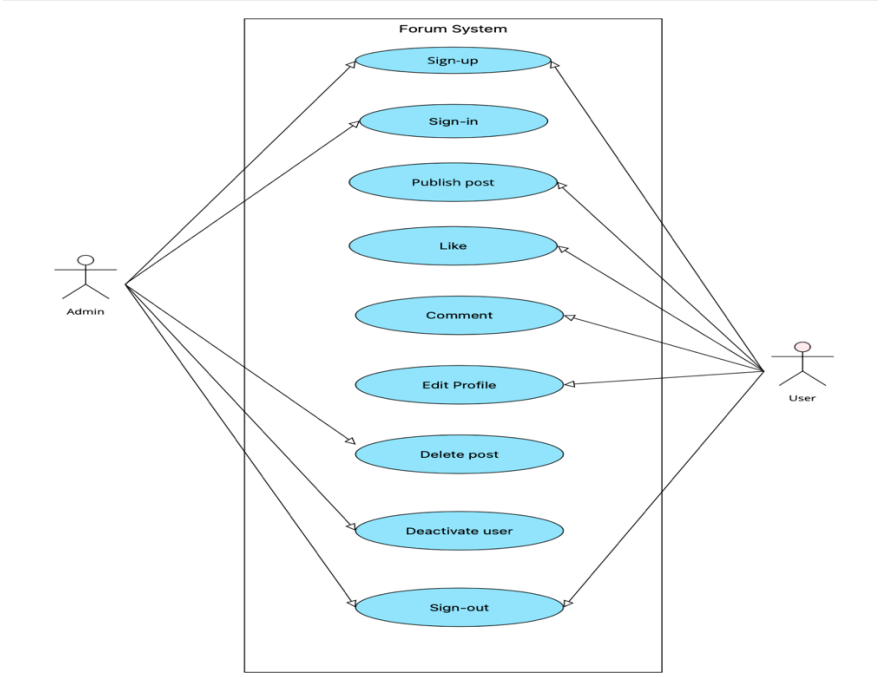


Figure 2.3: Use case diagram for One Community Project

3. Detailed Description of Components

3.1 Component Diagram

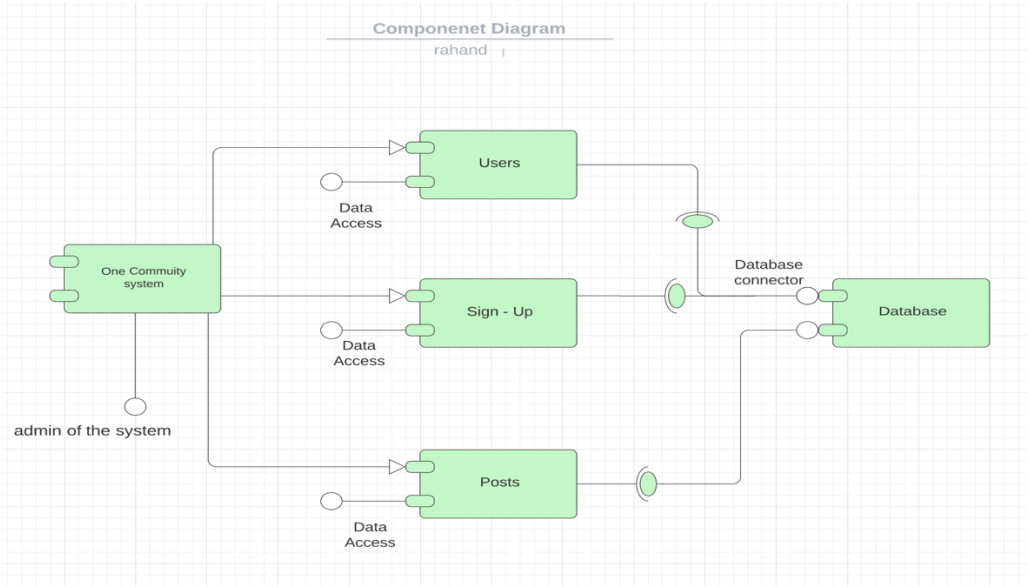
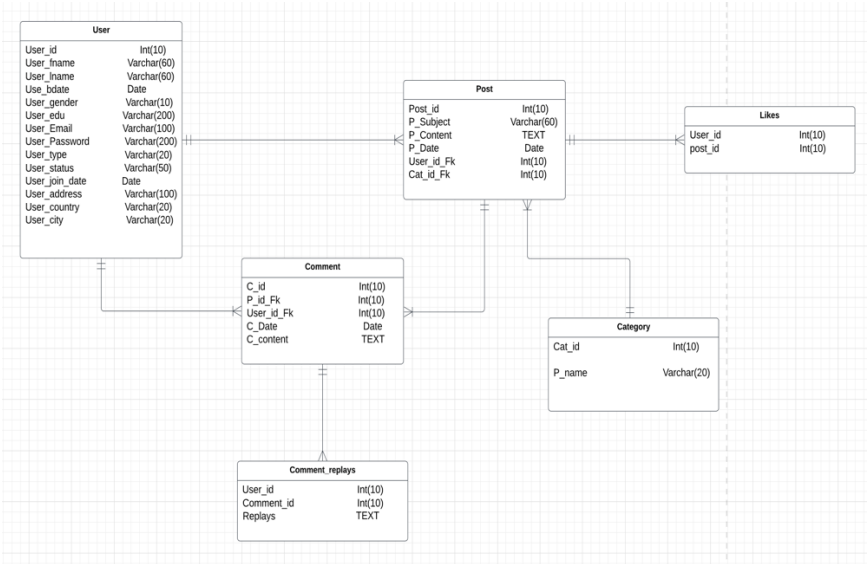


Figure 2.4: Component Diagram of One Community project

3.2 Detailed Description

Here is the class diagram :



Here is the sequence diagram:

For Admin

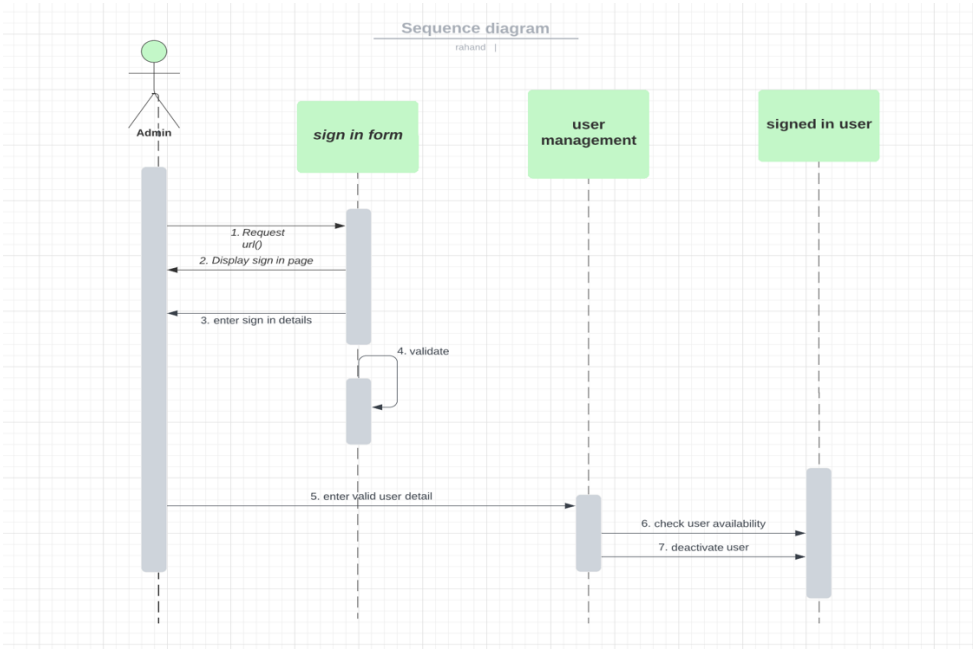


Figure 3.2.1: Sequence diagram for admin

For Non-admin

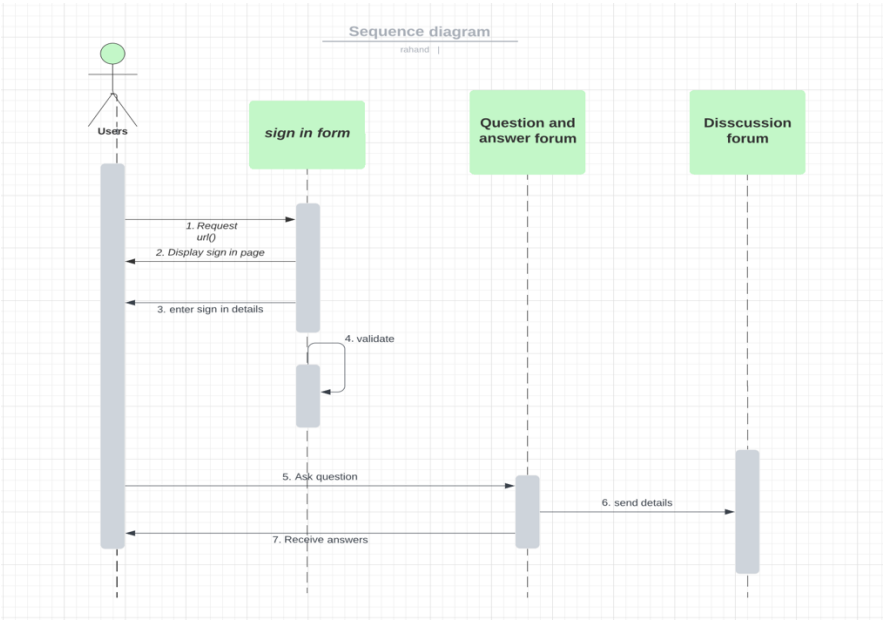


Figure 3.2.2: Sequence diagram for non-admin

4. Data Design

4.1 Data Dictionary

Table 3.1.1 Database normalization table for User Data

Name	Type	Default	Index
User_id	Int	Null	Primary key
User_fname	Varchar	Null	Not Null
User_lname	Varchar	Null	Not Null
User_bdate	Date	Null	Not Null
User_gender	Varchar	Null	Not Null
User_edu	Varchar	Null	Not Null
User_Email	Varchar	Null	Not Null
User_Password	Varchar	Null	Not Null
User_type	Varchar	Null	Not Null
User_status	Varchar	Null	Not Null
User_join_date	Date	Null	Not Null
User_address	Varchar	Null	Not Null
User_country	Varchar	Null	Not Null
User_city	Varchar	Null	Not Null

Table 3.2.2 Database normalization table for Post

Name	Type	Default	Index
Post_id	Int	Null	Primary key
P_Subject	Varchar	Null	Not Null
P_Content	TEXT	Null	Not Null
P_Date	Date	Null	Not Null
User_id	Int	Null	Foreign key
Cat_id	Int	Null	Foreign key

Table 3.2.3 Database normalization table for Comment

Name	Type	Default	Index
C_id	Int	Null	Primary key
P_id	Int	Null	Foreign key
User_id	Int	Null	Foreign key
C_date	Date	Null	Not Null
C_content	Text	Null	Not Null

Table 3.2.4 Database normalization table for Category

Name	Type	Default	Index
Cat_id	Int	Null	Primary Key
Cat_name	Varchar	Null	Not Null

5. User Interface Design

5.1 Overview of User Interface

This section needs much more focus than the other sections because this is where the final product presents. Here, the interface has been designed to let the user finds it easy and easy to follow. The overall layout, containers, navbar, foot bar, and other primary sections of the website are coloured light grey and black. The reason is because this makes the user feel comfier and healthier for the eyes. Another positive side is that the users stay longer and spend more time on the website, seeking to solutions without getting annoyed due to the colors and design. Color, margins, buttons, font, and other attributes and elements are taken care of in a way that users unconsciously perform the tasks.

5.2 Screen Images

Interface for non-admin

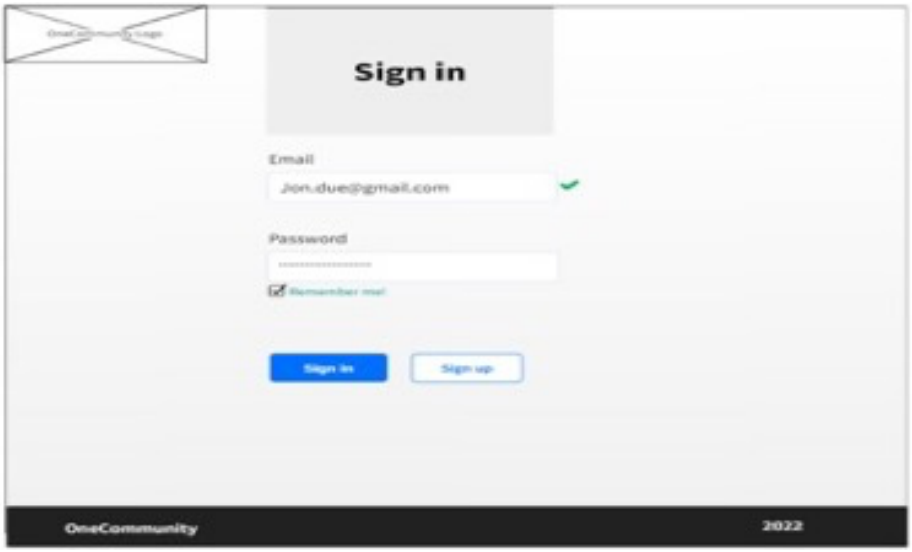


Figure 5.1: The possible front-end interface for Sign-in page

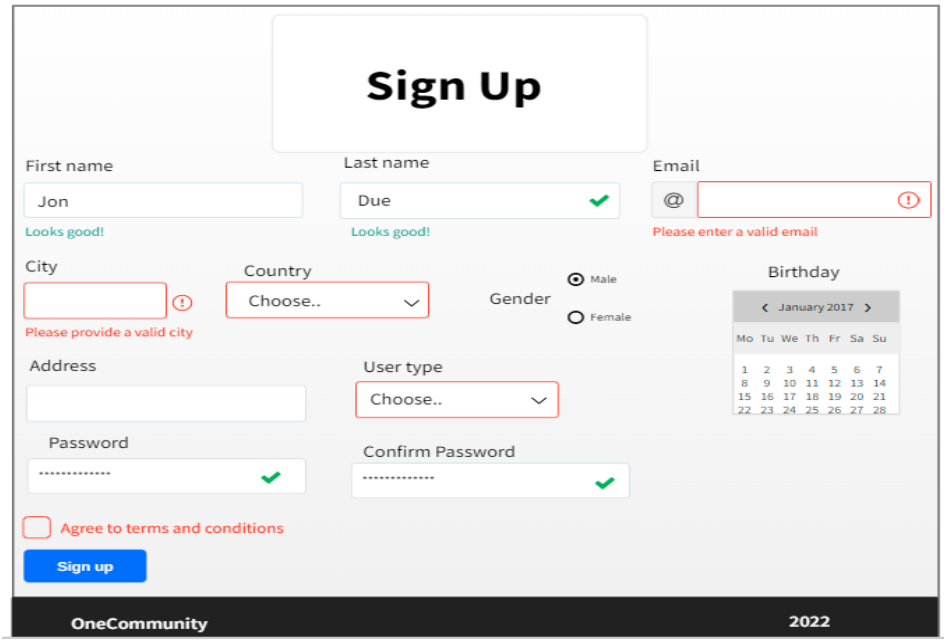


Figure 5.2: The possible front-end interface for Sign-up page

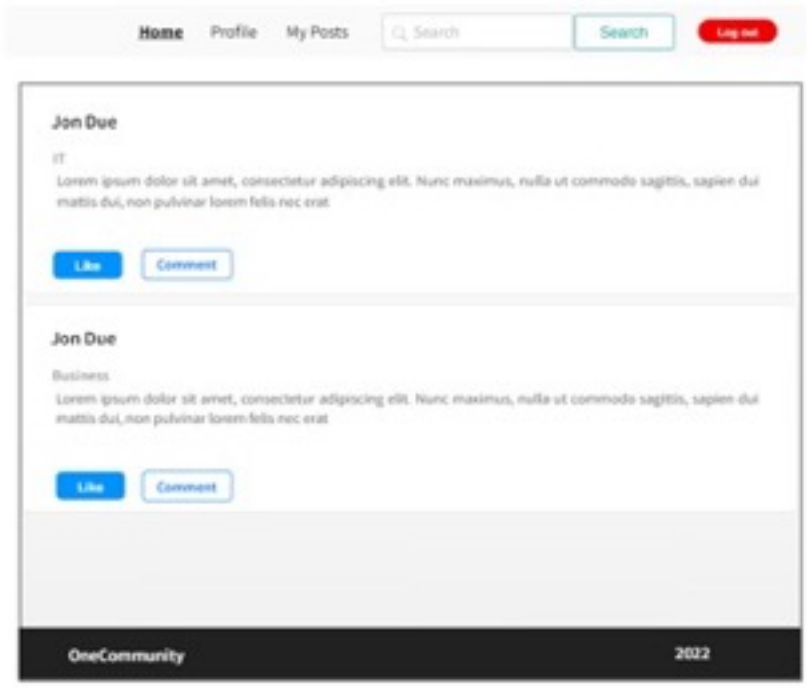


Figure 5.3: The possible front-end interface for Home page

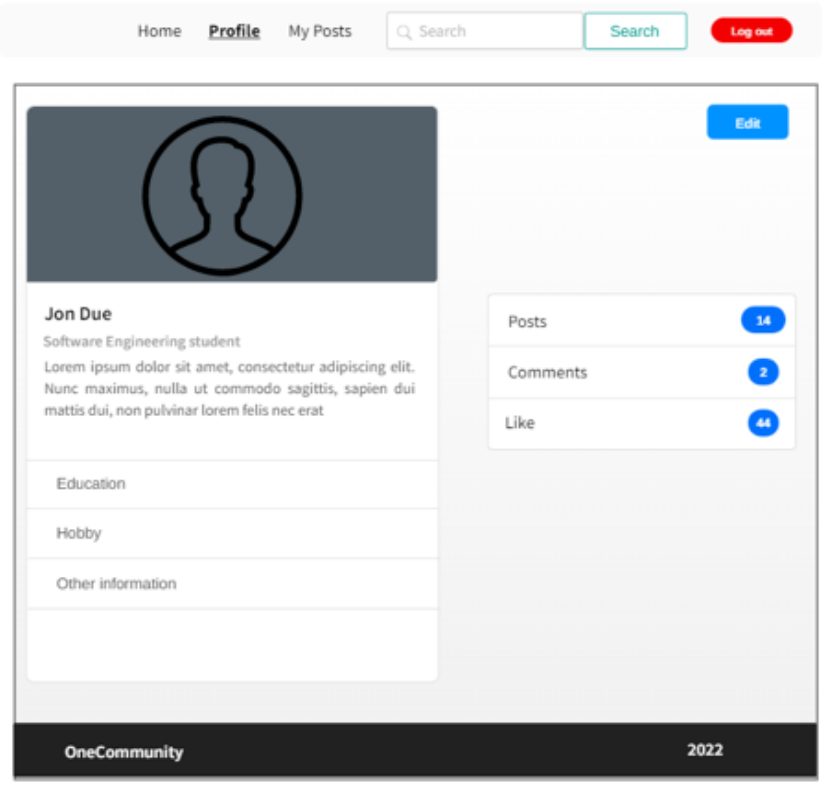


Figure 5.4: The possible front-end interface for Profile page

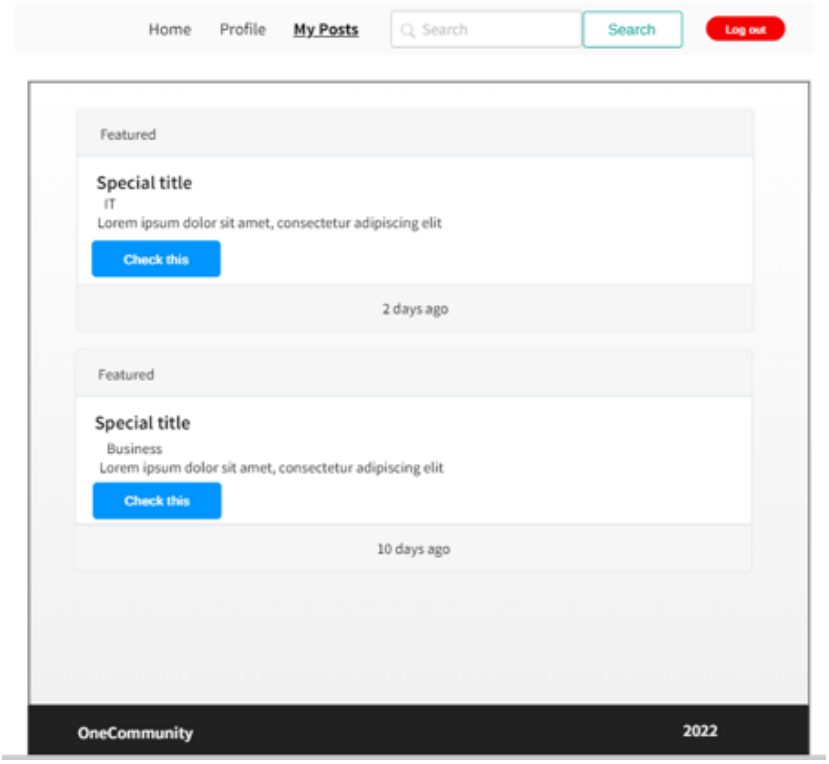


Figure 5.5: The possible front-end interface for My posts page

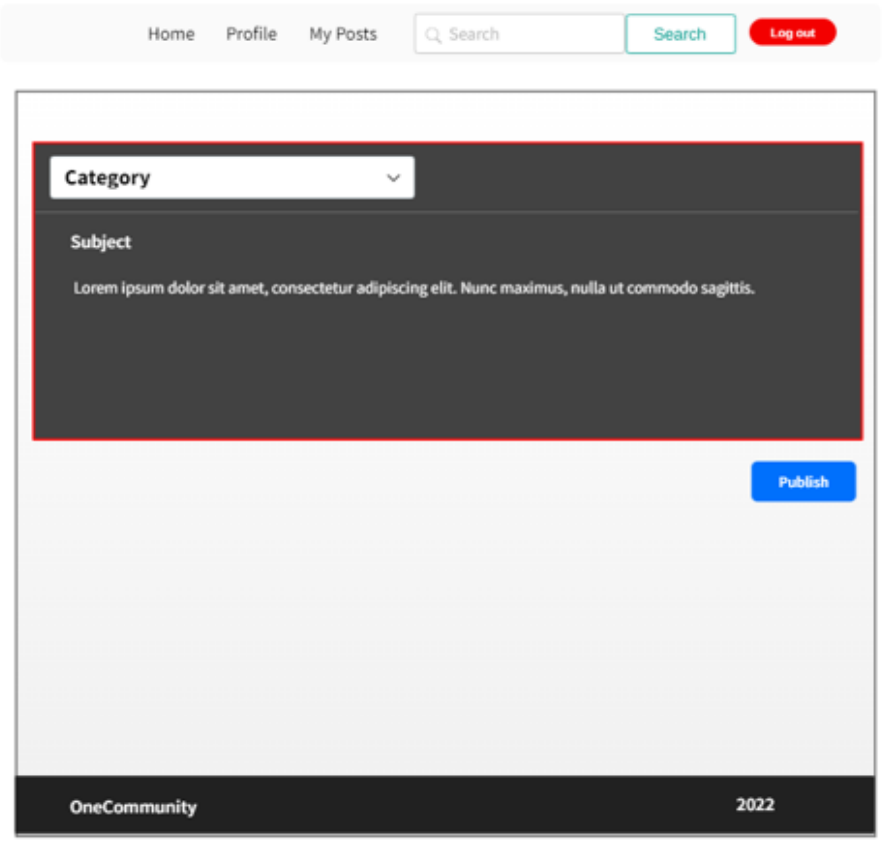


Figure 5.6: The possible front-end interface for Category page

Admin Interface

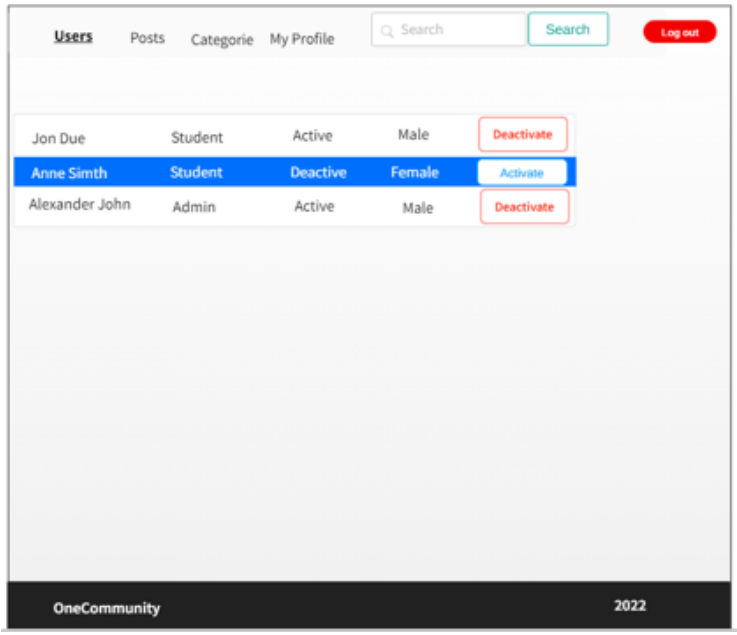


Figure 5.7: The possible front-end interface for User page

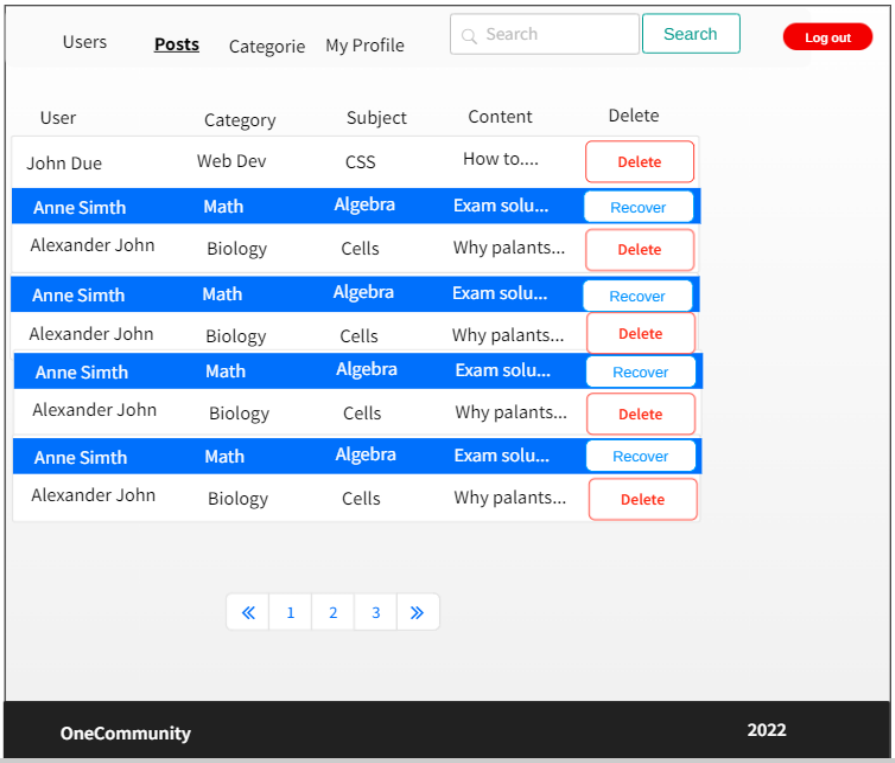


Figure 5.8: The possible front-end interface for Posts page

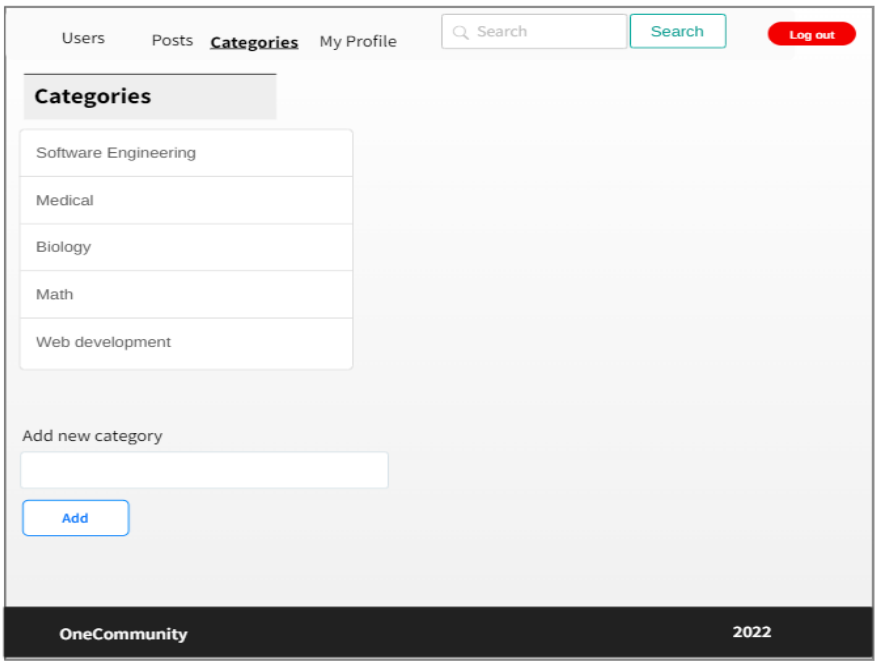


Figure 5.9: The possible front-end interface for Categories page

Appendix C

STD

Software Testing Documentation

(STD)

Software Engineering

Project Name: One Community

Prepared by: Rahand Mahdi Ahmed

Metrics Number: QU182SCSJ057

Date: 6/22/2022

Revision Page

g. Overview

In this new version of Software Test Development, we will run different attempts of tests against the software to ensure the quality and functionality of the system.

h. Target Audience

The target audience for this section are the testers of the web application who wish to run tests on the system.

i. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	**Rahand Mahdi**	**This version explains the software testing documentation of smart kitchen.**	**June 22, 2022**

Table of Contents

1 Introduction

1.1 Purpose
1.2 Scope
1.3 Definitions, Acronyms and Abbreviations
1.4 System Overview

2 Test Cases

2.1 Test TC001 for Module <Sign up>: <First name>
2.2 Test TC001 for Module <Sign up>: <Password>
2.3 Test TC001 for Module <Sign up>: <Email>
2.4 Test TC001 for Module <Sign in>: <Email>

3 Test Approach Analysis

1. Introduction

In this paper, the evaluating process of the One Community web application is introduced. Here, all the errors, bugs, technical issues and other areas of human errors are detected and fixed. The tests have been through all the pages and subpages of the web application, One Community. The aim is to minimize errors and faults of the system for better performance and functionality.

1.1 Purpose

This software test documentation provides the necessary information about testing activities that include test description and test results for One Community system.

1.2 Scope

One Community web application provides a community for every student and teacher in the world to ask questions and get answers regardless of any subjects. The key objectives are:

To provide an educational environment for students and improve communication between students.

To be accessible at any time and not be deleted and other students with the related issue can take advantage of it whenever they want.

To allow students to check the user profiles to evaluate the answers based on the background of the user.

To make it easier to find a solution to a problem at the university level.

1.3 Definitions, Acronyms and Abbreviation

Abbreviation	Definition
Web APP	Website Application

1.4 System Overview

In this STD, multiple tests are being attempted on several pages to evaluate the web APP. Doing so, provides better functionality for the system. For each test, results of the tests with descriptions are shown.

2. Test Cases

2.1 Test TC001 for Module <Sign up>: <First name>

UC001_01: Sign up (First Name)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_01_01	R	One word is not accepted	Invalid name	Fail
TC001_01_02	Test	Okay, no issue	Okay, no issue	Pass

2.2 Test TC001 for Module <Sign up>: <Password>

UC002_01: Sign up (password)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC002_01_01	1234	Password is too short, try again	Password should not be less than 6 characters	Fail
TC002_01_02	123@123R	Password OK	Password OK	Pass
TC002_01_13	(empty)	Password cannot be empy	Invalid password	Fail

2.3 Test TC003 for Module <Sign up>: <Email>

UC003_01: Sign Up (Email)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC003_01_01	Rahand	Email is invalid	Not an email style	Fail
TC003_01_02	Rahand.com	Email is invalid	Not an email style	Fail
TC003_01_03	Rahand@gmail.com	Email is valid	Successful	Pass

2.4 Test TC004 for Sign in: <email>

UC003_01: Sign in (Email)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC004_01_01	R.gmail.com	Email is invalid	Not an email style	Fail
TC004_01_02	r@gmail.com	Email is not in the list	Invalid email address	Fail
TC004_01_03	Rahand@gmail.com	Email is valid	Successful	Pass

3. Test Approach Analysis

UC001: Sign up <first name>

First name

EP class 1 (invalid): 1 < first name

EP class 2 (valid): 2 < first name < 20

Password

EP class 1 (invalid): Password < 6

EP class 2 (valid): password > 6

EP class 3 (invalid): password =! (Empty)

Email

EP class 1 (invalid): email is not following email style (It does not contain @)

EP class 2 (invalid): email is not following email style (It does not contain @)

EP class 3 (valid): email follows the standards of email style

UC002: Sign in <email>

EP class 1 (invalid): email is not following email style (It does not contain @)

EP class 2 (invalid): email has not registered in the database

4. EP class 3 (valid): email can be found in the database