UNIVERSITI SAINS MALAYSIA

SCHOOL OF COMPUTER SCIENCES

SEMESTER I, ACADEMIC SESSION 2023/2024

CSE441 – SOFTWARE PROCESS AND QUALITY ASSURANCE

INDIVIDUAL ASSIGNMENT

YAP YOU QUAN

153484

For the attention of:

Ts. Dr. Mohd Heikal Husin

Dr. Mohammad Ali Sarvghadi

Submission Date:

15th December 2023

Table of Contents

List of Tables

List of Figures

1. Project Background

1.1 Problem Background

For university lecturers and graduates, the process of teaching and learning is vital. In order to promote personal development, it entails two-way interaction and communication between university instructors and students in an organized setting. At Universiti Sains Malaysia, there are more than 20,000 enrolled students and more than 2,300 faculty members in year 2023. In the following problem statements, only the teaching and learning procedure at Universiti Sains Malaysia Computer Science—referred to as USM CS in the following paragraphs—will be covered and discussed.

In the USM CS community, all the members will utilize the online platforms such as USM E-learning platform to access all the necessary information. However, the E-learning platform has its own limitations. For example, the final year project coordinators are the main person-in-charge that are responsible for uploading the final year project toolkits such as marking rubrics, report guidelines and report templates on the e-learning platform for the final year students. However, the coordinators will disseminate the toolkits to the lecturers separately by using OneDrive, Moodle, or emails. This is because the lecturers do not have permission to access to the final year E-learning platform. If the lecturers are assigned as examiners, this will immediately affect their efficiency in locating the relevant materials because they will have to go through several emails and Onedrive links in order to find the respective document to refer back when grading students' work.

Referring to the above paragraph, there is not a direct channel of communication between the lecturers and the coordinator of the final year project regarding the exchange and sharing of materials. The third-year students face the same issue since the internship coordinator must upload all internship paperwork, including insurance forms, report duty forms, and company lists, via the Padlet platform. These are the principal problems and causes that led to the creation of this system. In order to improve the exchange of knowledge and resources among community members, we intend to create a centralized one-stop platform that will assist USM CS academics in sharing information more effectively.

1.2 System Overview

1.2.1 System Objectives

1. To develop a centralized portal system for all the USM Computer Science students, lecturers, and admins to access all the resources and materials in a more effective way.
2. To develop a reporting module that contains displaying dashboards in the system for all groups to allow the group of people in accessing the respective files, materials, or resources.
3. To develop event management module in the system to allow the lecturers and staffs to create and disseminate important announcements.
4. To develop user permissions module for the respective groups of users to access only specific information.

1.2.2 System Module Breakdown

The following diagram shows the module breakdown diagram for the project.

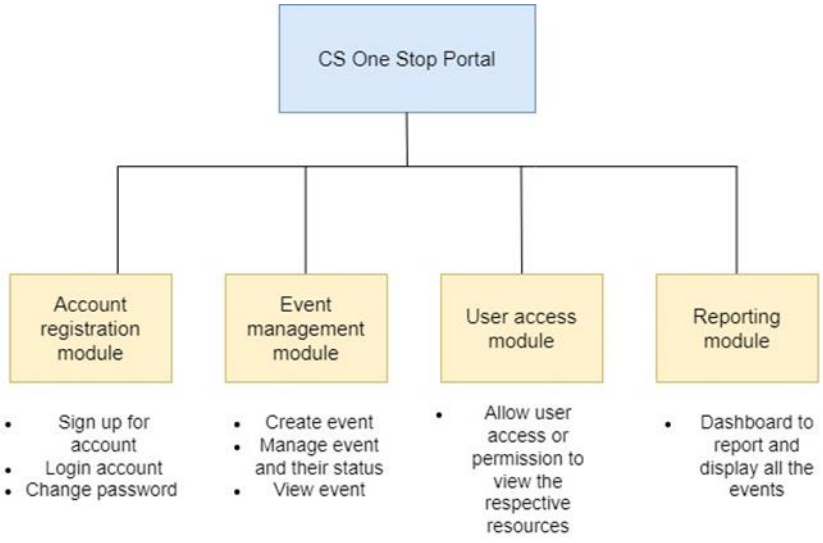

Figure 1: Module breakdown diagram for the project

1.2.3 System description and function

To implement the USM CS one stop portal website system, a few possible modules will be implemented according to the module breakdown diagram above. I will next give a brief explanation on each module. The project will be implemented by using the waterfall method.

The first module that will be implemented in the system is the account registration module. In this module, when a person visits the website for the first time, the person will be asked to sign up for an account in the system. The details needed in creating an account will be name, matric number for students, staff ID for staffs and lecturers, and their email addresses. After the user creates an account, the user should receive a message that shows their account creation is successful. They should be able to log in to the system. They are also allowed to change account password. It is necessary to take note that only staffs and student from USM Computer Science course are allowed to create account in the system as it is only designed for USM CS community.

The second module that will be included in this system will be event management module. In this module, the lecturers or staffs are able to create events. By creating events, the lecturers are able to upload and share the important resources, learning materials and information. After this, lecturers should be able to view the event they have created. The edit and save buttons should be made available for the lecturers if they wish to edit or modify any resources or documents. The buttons should only be made visible and accessible only for lecturers and staffs. Besides, the lecturers can also utilize the system by informing ongoing competitions to the students. In this case, event status is needed. If the competition is already ended or the registration deadline is passed, the competition should be deleted, and the students cannot access to the competition details. Notifications should be sent to the students or lecturers when an event is created on the portal.

Next, the user access module will be implemented in the system as well. In this module, users should be given access or permission to view and download respective resources only. For example, all the internship documents such as insurance form, certification letter form and list of company should be made visible to third year students only. Students from other year of study are not allowed to view or access the internship documents. In this case, restrictions should be made so that every group of students should only access the respective information. However, general information or learning resources will be made available for all group of users in USM CS communities.

Lastly, a reporting module will be implemented in the system. In this module, dashboards will be developed for all events. On the dashboard, events will be displayed in which list of resources, materials or documents will be shown to the students. For the competition part, a list of participants will be made available as well. Every event will have one dashboard that displays the respective resources. Dashboards should be made user-friendly and easily accessible to all users.

2. Software Process Model

Agile development process will be the software process model that will be used in the development of my final year project.

The agile methodology was selected due to its user-centered approach, which facilitates uncertainty management and allows for quick responses to the environment's rapidly changing requirements. The software development process is divided into iterations by the agile technique, which also uses an iterative approach. During these iterations, project components are developed and tested. Every iteration has a completion date by which specific deliverables need to be made. The procedures will be repeated in a loop during agile iterations until an optimal outcome is obtained. Agile methodology is also a systematic model to develop the project because it involves several phases such as project planning, system design, system develop, testing, deploying the project and project review. The following figure shows the overall cycle of agile methodology.

Figure 2: Agile methodology phases [3]

A total of 10 weeks is given to develop the whole project. My final year project consists of four major modules, each of which should take two weeks to finish. The first two weeks will be used in system planning and design, in which it involves few phases such as requirements gathering, user interface design and also develop the system requirement and design which contains overall use case diagram, use case descriptions and architecture diagram.

Each module will be divided into one iteration. Each iteration will last around two weeks, meaning that it will take about eight weeks to finish all of the modules. The remaining weeks will be devoted to project closure and documentation. At the end of the agile process, system testings shall be performed, and project deployment should be done. A shippable and complete system with minimal defects should be produced. It is imperative that we consistently ensure the delivery of functional software, with a focus on sustainable development through user collaboration and quick response to dynamic requirements.

3. Software Requirements Specifications

3.1 System Requirements

3.1.1 Functional requirements

The system is developed based on the functional requirements below:

Functional requirements	Descriptions
FR-001	The system will allow users to register as a new user of the system so that he/she can access all functionalities in the website.
FR-002	The system will allow users to login to the website using the registered username and password.
FR-003	The system will allow users to view dashboard of the website
FR-004	The system will allow users to change and update password
FR-005	The system will allow users to upload resources and materials
FR-006	The system will allow users to view list of resources and materials
FR-007	The system will allow users to update list of resources and materials
FR-008	The system will allow users to search for contents in the website from search bar
FR-009	The system will allow users to view notifications when there are any updates
FR-010	The system will allow the logged in user (lecturers that act as coordinators) to create access for the lecturers to view resources and materials.
FR-011	The system will allow the logged in user (lecturers that act as coordinators) to remove lecturers access which prevents the lecturers from viewing the resources and materials.
FR-012	The system will allow the logged in user (lecturers) to create access for the students to view resources and materials.
FR-013	The system will allow the logged in user (lecturers) to remove students access which prevents students from viewing the resources and materials.
FR-014	The system will allow users to create list of events that contains competitions for other users(students) to join
FR-015	The system will allow users to view list of events that contains competitions
FR-016	To allow users to update list of events that contains competitions

3.1.2 Non-functional requirements

The website application should be developed and evaluated based on the non-functional requirements below:

a) Product Evaluation

Evaluation List	**Descriptions**
Performance	• Every user event request and response time should be less than 3 seconds. • The application start-up time must be less than 3 seconds. • The application should be capable of handling the application process without delay even if the user opens multiple applications at the same time. • The application should be able to save current progress and return to the saved point after being interrupted. • The application should support the latest stable version for Google Chrome, Firefox, Microsoft Edge and Safari browser.
Dependability	• The application must not have downtime during normal working hours. • The application must not have more than 3 minutes of downtime per day outside working hours. • The mean time of failure of the application should be 5 minutes. • The application update process must finish within 3 hours, so data is available by 8 a.m. local time after an overnight update.
Space	• The application should be able to support the sudden spikes of the user without risking crashing the system
Usability	• The application must have a user-friendly and easy-to-understand interface in which the user errors are minimized. • The application must have a user-friendly and easy-to-understand interface in which the user errors are minimized.

	• The application feature should be able to run similarly in different mobile operating systems
Portability	• The application could enable users to use the system either on mobile devices, tablets, or computers.
Recoverability	• The application failure fix should be completed within 1 hour when it happens. • The application should allow users to manually backup their data from time to time. • The application should allow users to download all the personal data in the application.
Security	• The personal information and sensitive data of the user must only be accessed by authorized users.

a) Organizational Evaluation

Evaluation List	**Description**
Operational	• The authentication process for user.
Development	• The software developed using Angular, Express.js, JavaScript, HTML, and CSS.

b) External Evaluation

Evaluation List	Description
Regulatory	• The application should comply with the The Computer Crimes 1997
Environment	• The application should be able to run at normal capacity without downtime during raining season

3.2 Use case modelling

3.2.1 Use case diagram

The following shows the overall use case diagram of CS One Stop Portal.

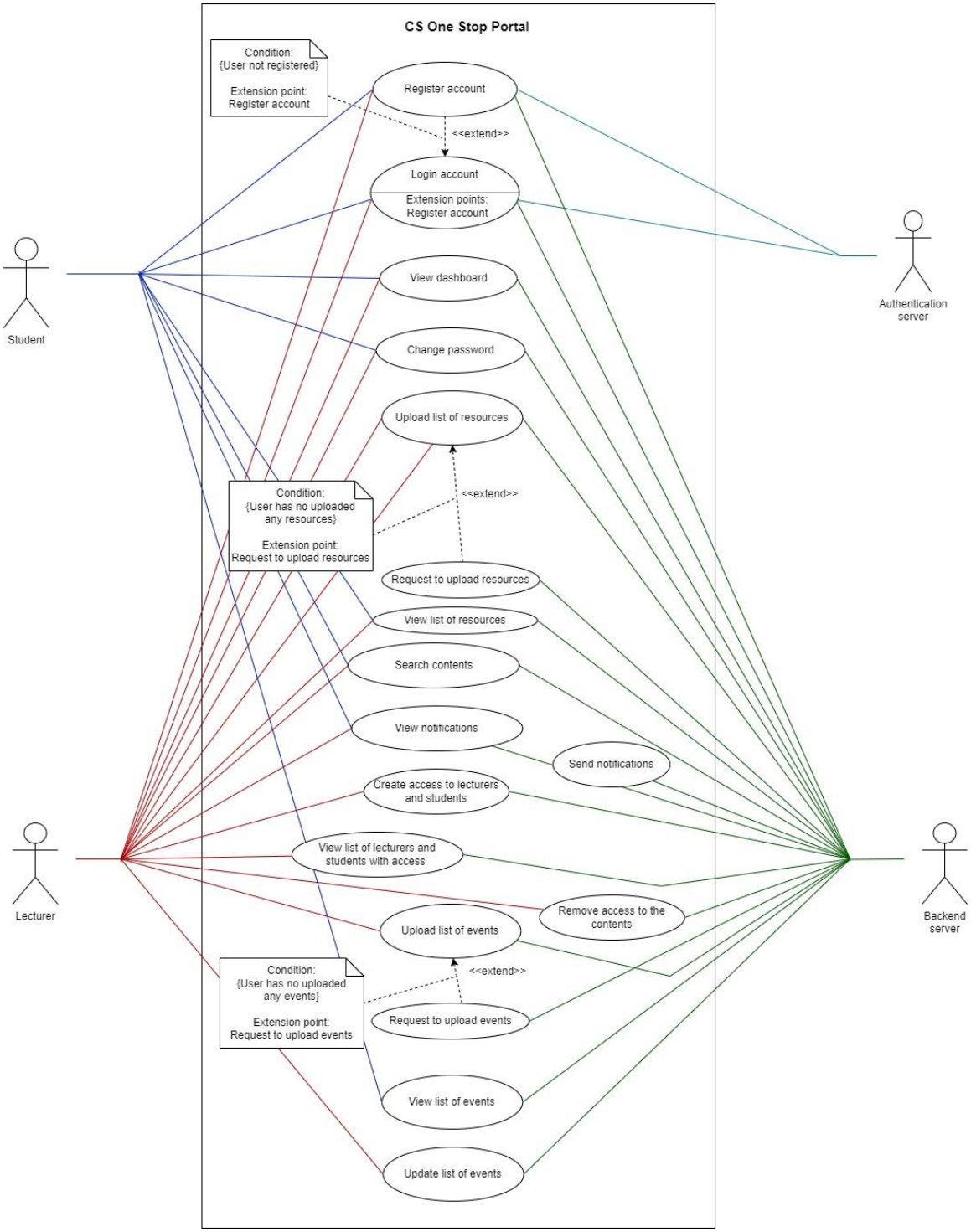

Figure 2 : Use case diagram

3.2.2 Use case descriptions

U001: Register account

<table>
<tr><td>Use case name:</td><td colspan="2">Register account</td></tr>
<tr><td>Scenario:</td><td colspan="2">Register an account to use the system</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Users want to register account to use the system</td></tr>
<tr><td>Brief description:</td><td colspan="2">Users fill in the form respectively that includes all the important details such as email address and password.</td></tr>
<tr><td>Actors:</td><td colspan="2">Students, lecturers, staffs</td></tr>
<tr><td>Related use cases:</td><td colspan="2">-</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Students, lecturers, staffs, developer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Users have not registered an account on the system before</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Users have accounts on the system</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Users click on “register as ‘role’” button on the login page.

2. Users fill in the details in the form.

3. Users click the “submit” button.</td><td>1. The system will display the login page.

2. The system will prompt a form for users’ details after users click on the register button.

3. The system will collect, record and save information from the user.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. System cannot display the form.

2. Users enter wrong input in the form.

3. System cannot collect, record and save the information.

4. The login page is not displayed.</td></tr>
</table>

U002: Login Account

<table>
<tr><td>Use case name:</td><td colspan="2">Login account</td></tr>
<tr><td>Scenario:</td><td colspan="2">Login to the system using the created account</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Users want to log into the system by using the created accounts</td></tr>
<tr><td>Brief description:</td><td colspan="2">Users fill in email address and password that used to create their account in the given bar so that they can log into the system.</td></tr>
<tr><td>Actors:</td><td colspan="2">Students, lecturers, staffs</td></tr>
<tr><td>Related use cases:</td><td colspan="2">U001</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Students, lecturers, staffs, developer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Users have already created an account.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Users log into the system successfully and are able to see dashboard.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actors</td><td>System</td></tr>
<tr><td>1. Users fill in their email addresses and password.

2. Users click on the “Login to system” button.

3. Users log into the system.</td><td>1. The system will display the login page.

2. The system will display the dashboard page if users enter correct email address and password.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. System cannot display the login page.

2. Users prompt wrong inputs for email address and password.

3. System cannot display dashboard page even users enter correct email address and password.</td></tr>
</table>

U003: View dashboard

Use case name:	View dashboard	
Scenario:	View dashboard page of the website after log into the website.	
Triggering event:	Users want to view the dashboard page of the system.	
Brief description:	Users can view dashboard page of the website after they successfully log into the website.	
Actors:	Students, lecturers, staffs	
Related use cases:	U002	
Stakeholders:	Students, lecturers, staffs, developer	
Preconditions:	1. Users have already registered an account on the system. 2. System display login page.	
Postconditions:	Users enter correct email address and password.	
Flow of activities:	**Actors**	**System**
	1. Users enter the correct email address and password. 2. Users click the "login to system" button. 3. Users are navigated to the dashboard page of the system. 4. Users are able to view all the functionalities that are available on the dashboard.	1. The system displays a login page. 2. The system displays the dashboard page that contains all the functionalities of the system for the users to view.
Exception conditions:	The system is not displaying dashboard page even if users enter correct email addresses and passwords.	

U004: Change password

Use case name:	Change password	
Scenario:	Change and update password to log into the system	
Triggering event:	Users want to change password to log into the system	
Brief description:	Users can change and update password on the login page of the system	
Actors:	Students, lecturers, staffs	
Related use cases:	-	
Stakeholders:	Students, lecturers, staffs, developer	
Preconditions:	The system display login page.	
Postconditions:	Users enters new password in the given space.	
Flow of activities:	**Actors**	**System**
	1. Users click on the "change password" button. 2. Users key in the new password in the "New Password" space. 3. Users key in new password again in the "Confirm New Password" space. 4. Users click on "submit" button and new password is updated.	1. System displays "change password" button. 2. System collects and records the latest password updated by users.
Exception conditions:	1. "Change password" button is clicked but it is not functioning. 2. Users fail to update password after clicking on the "submit" button.	

U005: Upload list of resources

<table>
<tr><td>Use case name:</td><td colspan="2">Upload list of resources</td></tr>
<tr><td>Scenario:</td><td colspan="2">Upload list of resources and materials on the system</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Users want to upload resources and materials on the system.</td></tr>
<tr><td>Brief description:</td><td colspan="2">Users can upload resources such as report templates, marking rubrics and marking guidelines on the system.</td></tr>
<tr><td>Actors:</td><td colspan="2">Lecturers</td></tr>
<tr><td>Related use cases:</td><td colspan="2">-</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Lecturers, students, developer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">1. Users have registered accounts on the system.

2. Users are able to log into the system and view the dashboard page.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Users can click on “list of materials” on the sidebar of the website and perform uploading action.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Users click on the “list of materials” on the sidebar menu of the dashboard.

2. Users click on the “upload” button.

3. Users browse for the documents that they wish to upload.

4. Users click on the “save and submit” button.</td><td>1. The system displays dashboard page and sidebar menu.

2. The system displays a section that allows the users to upload documents after they click on the “upload” button.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. The “list of resources” sidebar menu is not displayed.

2. The uploading section is not displayed after users click on “upload” button.</td></tr>
</table>

U006: View list of resources

Use case name:	View list of resources	
Scenario:	View list of resources and materials on the system	
Triggering event:	Users want to view resources and materials on the system.	
Brief description:	Users can view resources such as report templates, marking rubrics and marking guidelines on the system.	
Actors:	Lecturers, students	
Related use cases:	U005	
Stakeholders:	Lecturers, students, developer	
Preconditions:	1. Users have registered accounts on the system. 2. Users are able to log into the system and view the dashboard page.	
Postconditions:	Users can click on "list of materials" on the sidebar of the website and view all the available materials.	
Flow of activities:	**Actor**	**System**
	1. Users click on the "list of materials" on the sidebar menu of the dashboard. 2. Users view all the available materials on the website.	1. The system displays dashboard page and sidebar menu. 2. The system displays a list of resources after users click on the "list of materials" menu on the sidebar.
Exception conditions:	1. The sidebar menu is not displayed. 2. The list of materials is not displayed after users click on the sidebar menu.	

U007: Update list of resources

<table>
<tr><td>Use case name:</td><td colspan="2">Update list of resources</td></tr>
<tr><td>Scenario:</td><td colspan="2">Update list of resources and materials on the system</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Users want to update resources and materials on the system.</td></tr>
<tr><td>Brief description:</td><td colspan="2">Users can add, reupload or remove the materials on the system</td></tr>
<tr><td>Actors:</td><td colspan="2">Lecturers</td></tr>
<tr><td>Related use cases:</td><td colspan="2">U005, U006</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Lecturers, students, developer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">1. The list of resources is available on the website.

2. The edit button is available, and it is only visible and accessible for specific users (lecturers).

3. The uploaded documents should be in specific documents and the given range of size.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Users are able to edit and update list of resources successfully.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actors</td><td>System</td></tr>
<tr><td>1. Users click on “list of materials” on the sidebar menu of the dashboard.

2. Users click on the “edit” button on the page.

3. Users perform editing action on the resources. They can click on the dustbin icon to remove the content and plus button to add or reupload the documents.

4. Users click on “Save and Submit” button after editing the documents.</td><td>1. System display dashboard page that contains sidebar menu.

2. System prompt editing page for the users after they click on the editing button.

3. System records the latest resources updated by users.

4. System displays the latest list of resources.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. The editing page is not displayed after users click on “edit” button.

2. Users failed to save and submit the updated documents.</td></tr>
</table>

U008: Search contents

<table>
<tr><td>Use case name:</td><td colspan="2">Search contents</td></tr>
<tr><td>Scenario:</td><td colspan="2">Search available contents on the website</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Users want to search for specific documents or contents on the website</td></tr>
<tr><td>Brief description:</td><td colspan="2">Users wants to search for specific documents by entering keywords in the given search bar on the page.</td></tr>
<tr><td>Actors:</td><td colspan="2">Students, lecturers, staffs</td></tr>
<tr><td>Related use cases:</td><td colspan="2">-</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Students, lecturers, staffs, developer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">1. The list of documents is available on the website.

2. The search bar is displayed on the website.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">1. The search bar is functioning.

2. The documents named with the keywords entered by the users will be filtered out and displayed.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. User enters keyword in the given search bar.

2. Users click on the search button.</td><td>1. System displays search bar.

2. The system gets the input from the users and performs filtering action.

3. System displays the filtered documents.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. The search bar is not displayed on the website.

2. The correct documents are not displayed after users enter the keyword and click on the search button.</td></tr>
</table>

U009: View notifications

Use case name:	View notifications	
Scenario:	View available notifications on the website	
Triggering event:	Users want to view notifications on the website.	
Brief description:	Users want to view the latest notifications of the latest information and resources on the website.	
Actors:	Students, lecturers	
Related use cases:	U007	
Stakeholders:	Students, lecturers, staffs, developers	
Preconditions:	1. Users receive the latest notification. 2. The notification list is displayed on the website.	
Postconditions:	Users are able to click and view available notification	
Flow of activities:	**Actors**	**System**
	1. Users access to the system dashboard page. 2. Users click on the "ring" icon on top of the website. 3. Users view all the available notifications, including the latest ones.	1. System displays dashboard page. 2. System prompt a notification list for the users to view.
Exception conditions:	1. The users do not receive any notifications. 2. The notification list is not displayed.	

U010: Give lecturers access

Use case name:	Give lecturers access	
Scenario:	Give access to the other lecturers in accessing the document	
Triggering event:	Lecturers want to give access to other lecturers.	
Brief description:	Lecturers that act as coordinators grant access to other lecturers to view the documents and resources.	
Actors:	Lecturers	
Related use cases:	-	
Stakeholders:	Lecturers, developers	
Preconditions:	Lecturers that want to grant access to other lecturers should act as coordinators in CS related fields (internship, final year project, researches, students related affairs etc.)	
Postconditions:	Lecturers are able to grant access to other lecturers.	
Flow of activities:	**Actors**	**System**
	1. Users upload materials on the portal. 2. Users choose the name of lecturers that they wish to grant access to in the provided name list in the "grant access to" section. 3. Users click the "save and submit" button.	1. System displays list of materials page. 2. System prompts the "grant access to" section and a name list of lecturers for the users to choose. 3. System prompts "save and submit" button.
Exception conditions:	1. The list of materials page is not displayed. 2. The name list of lecturers is not displayed in the "grant access to" section.	

U011: Remove lecturers access

<table>
<tr><td>Use case name:</td><td colspan="2">Remove lecturers access</td></tr>
<tr><td>Scenario:</td><td colspan="2">Remove other lecturers' access in accessing the documents</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Lecturers want to remove other lecturers' access in viewing the documents.</td></tr>
<tr><td>Brief description:</td><td colspan="2">Lecturers that act as coordinators want to remove other lecturers' access to view the documents and resources.</td></tr>
<tr><td>Actors:</td><td colspan="2">Lecturers</td></tr>
<tr><td>Related use cases:</td><td colspan="2">U010</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Lecturers, developers</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Lecturers that want to remove other lecturers' access should act as coordinators in CS related fields (internship, final year project, researches, students related affairs etc.)</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Lecturers are able to remove other lecturers' access in viewing the documents.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actors</td><td>System</td></tr>
<tr><td>1. Users view the list of materials.

2. Users choose the documents and click on the edit button.

3. Users go to the "grant access to" section and choose the name of lecturers that they wish to remove access.

4. Users click on the cross button shown beside the lecturers' names.

5. Users click on "save and submit" button.</td><td>1. System displays list of materials page.

2. System prompts the "grant access to" section and a name list of lecturers for the users to remove access.

3. System prompts a cross button beside the lecturers' name.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. Users cannot access to the list of materials page.
2. Users cannot remove other lecturers' access even they already clicked on the cross button and "save and submit" button.</td></tr>
</table>

U012: Give students access

<table>
<tr><td>Use case name:</td><td colspan="2">Give students access</td></tr>
<tr><td>Scenario:</td><td colspan="2">Give access to the students in viewing the documents</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Lecturers want to grant access to the students in viewing the available documents.</td></tr>
<tr><td>Brief description:</td><td colspan="2">Lecturers give access to students in viewing the available resources.</td></tr>
<tr><td>Actors:</td><td colspan="2">Lecturers, students</td></tr>
<tr><td>Related use cases:</td><td colspan="2">-</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Lecturers, students, staffs, developers</td></tr>
<tr><td>Preconditions:</td><td colspan="2">1. Lecturers have the latest students' name list.

2. All students are assigned the correct batch name. (example, Year 3 students are assigned as "Undergrad Year 3 PPSKOMP)</td></tr>
<tr><td>Postconditions:</td><td colspan="2">Lecturers are able to grant access to students.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Users upload materials on the portal.

2. Users choose the name of lecturers that they wish to grant access to in the provided batch list in the "grant access to" section.

3. Users click the "save and submit" button.</td><td>1. System displays list of materials page.

2. System prompts the "grant access to" section and a batch list for the users to choose.

3. System prompts "save and submit" button.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. The list of materials page is not displayed.

2. The batch list is not displayed in the "grant access to" section.</td></tr>
</table>

U013: Remove students access

Use case name:	Remove students access	
Scenario:	Remove other students' access in accessing the documents	
Triggering event:	Lecturers want to remove other students' access in viewing the documents.	
Brief description:	Lecturers remove other lecturers' access to view the documents and resources.	
Actors:	Lecturers, students	
Related use cases:	U012	
Stakeholders:	Lecturers, students, developers	
Preconditions:	1. Lecturers have the latest students' name list. 2. All students are assigned the correct batch name. (example, Year 3 students are assigned as "Undergrad Year 3 PPSKOMP)	
Postconditions:	Lecturers are able to remove students' access.	
Flow of activities:	**Actors**	**Systems**
	1. Users view the list of materials. 2. Users choose the documents and click on the edit button. 3. Users go to the "grant access to" section and choose the name of batch list that they wish to remove access. 4. Users click on the cross button shown beside the batch name. 5. Users click on "save and submit" button.	1. System displays list of materials page. 2. System prompts the "grant access to" section and a name list of lecturers for the users to remove access. 3. System prompts a cross button beside the lecturers' name.
Exception conditions:	1. Users cannot access to the list of materials page. 2. Users cannot remove other students' access even they already clicked on the cross button and "save and submit" button.	

U014: Create list of event

Use case name:	Create list of event	
Scenario:	Create list of event that contains competition	
Triggering event:	Lecturers want to create a list of event that contains competition.	
Brief description:	Lecturers create a list which contains various type of competitions that student can participate in.	
Actors:	Lecturers, students	
Related use cases:	-	
Stakeholders:	Lecturers, students, developer	
Preconditions:	1. The competition is valid. 2. Lecturers must include posters, terms and conditions and links to participate the competition.	
Postconditions:	1. Lecturers are able to create a list of event. 2. The list of event is available on the website.	
Flow of activities:	**Actor**	**System**
	1. Users click on the "list of events" on the sidebar menu of the website. 2. Users click on the "upload" button. 3. Users upload posters, terms and conditions, link to participation and other relevant competition resources. 4. Users click on "Save and submit" button.	1. System display "list of event" on the sidebar menu. 2. System display "upload button". 3. System requires user to upload all the relevant competition resources in the uploading section.
Exception conditions:	1. Users cannot access to the list of events page. 2. The uploading section is not displayed after users click on "upload" button. 3. Users can create event without uploading any relevant competition resources.	

U015: View list of event

Use case name:	View list of event	
Scenario:	View list of event that contains competition	
Triggering event:	Users want to view list of event that contains competition.	
Brief description:	Users view a list which contains various type of competitions that student can participate in.	
Actors:	Lecturers, students	
Related use cases:	-	
Stakeholders:	Lecturers, students, developer	
Preconditions:	The list of competition is displayed on the website.	
Postconditions:	1. Users are able to view the list of event. 2. User are able to click on their interested event and perform further actions (read terms of conditions, register via the given link etc)	
Flow of activities:	**Actor**	**System**
	1. Users click on the "list of events" on the sidebar menu of the website. 2. Users view the list of event that contains various type of competitions.	1. System display "list of event" on the sidebar menu. 2. System prompt list of competitions after users click on the "list of events" sidebar menu.
Exception conditions:	1. Users cannot access to the list of events page. 2. No list of event is displayed after users click on the sidebar menu, despite they received notifications on a list of competitions is uploaded.	

U016: Update list of event

<table>
<tr><td>Use case name:</td><td colspan="2">Update list of event</td></tr>
<tr><td>Scenario:</td><td colspan="2">Update list of event that contains competition</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Users want to update list of event that contains competition.</td></tr>
<tr><td>Brief description:</td><td colspan="2">Users update or remove the list of event which contains various type of competitions.</td></tr>
<tr><td>Actors:</td><td colspan="2">Lecturers</td></tr>
<tr><td>Related use cases:</td><td colspan="2">-</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Lecturers, students, developer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">1. There are updates on the competition resources. (example: change of registration links, updated poster etc)

2. The event is expired.</td></tr>
<tr><td>Postconditions:</td><td colspan="2">1. Users are able to update the list of event.

2. User are able to view the latest list after they perform updating action on the website.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Users click on the “list of events” on the sidebar menu of the website.

2. Users click on “update” button beside the list of event that contains various type of competitions.

3. Users perform updating action.

4. Users click “Save and submit” button after updating.</td><td>1. System display “list of event” on the sidebar menu.

2. System prompt “update” button beside list of competitions.

3. System prompt “Save and submit” button.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1. Users cannot access to the list of events page.

2. The update button is not displayed beside the list of event.

3. The old list is displayed even after users have performed updating action and clicked “Save and submit” button.</td></tr>
</table>

4. Requirements Traceability Matrix

Requirement ID	Requirement description	Justification	Test case ID	Test result	Notes
REQ-001	Login page	Users need to login to the system by using the created accounts	TC-001: No input is entered. TC-002: Only email address is entered TC-003: Only password is entered TC-004: Wrong email password or password or combinations is entered TC-005: Correct combination of email password and password is entered	TC-001, TC-002, TC-003, TC-004: System prompts an error page to prevent users from entering the dashboard page. TC-005: System prompts loading page and allow users to access dashboard page.	All requirements are done designed properly and sent to user acceptance test.
REQ-002	Loading page	It is the starting page for the users to access the website.	TC001: Loading page is showing. TC002 : Loading page is not showing.	TC001: Login page will be displayed. TC002: Dashboard page will be displayed.	
REQ-003	View dashboard	Users are able to view dashboard that display all functionalities and relevant information.	TC001: Dashboard page is not displaying. TC002: Dashboard page is displayed but the sidebar menu is missing. TC003: A complete dashboard page with sidebar menu is displayed.	TC001,TC002: The implementation of the dashboard page needs to be revised again. TC003: Users can access to the functionalities on the dashboard page.	

REQ-004	Change password	Users want to change and update password.	TC001: The new password contains only small letters. TC002: The new password contains only capital letters. TC003: The new password contains only password. TC004: The new password did not contain any special characters. TC005: The new password is less than 8 letters. TC006: The new password contains at least one small letter, one capital letter, one special character, one number, and its length are equal or more than 8 letters.	TC001, TC002, TC003, TC004, TC005: The users will be asked to reenter password. TC006: Users can update the password successfully.	
REQ-005	Search functionality	Users use the search bar displayed and search for the available contents.	TC001: Keywords are not entered inside the search bar and search icon is clicked. TC002: Keywords are entered in the search bar and search icon is clicked.	TC001: Search function is not triggered, and no contents will be displayed. TC002: Document that content the specific keywords will be filtered out and displayed.	

REQ-006	Resources management	Users are able to manage the resources and materials they uploaded on the website.	TC001: Users click on the “update” button, but they cannot perform further updating action. TC002: Users updated the resources but did not click on the “save and submit” button. TC003: Users clicked on the “save and submit” button after they updated the materials.	TC001, TC002: An error message will be prompted, and the users cannot update the list of resources. TC003: The list of resources is updated successfully, and the new updated list is displayed on the website.	
REQ-007	Event management	Users are able to manage the list of competition they uploaded on the website.	TC001: Users click on the “update” button, but they cannot perform further updating action. TC002: Users updated the resources but did not click on the “save and submit” button. TC003: Users update the list of competition without including the relevant information such as posters, terms and conditions, and registration link. TC004: Users uploaded every relevant competition resources and click on the “save and submit” button.	TC001, TC002, TC003 : An error message is displayed on the page and users are asked to redo the updating action. TC004: Users successfully manage all the resources on the list of event.	

REQ-008	Access management	Users are able to manage the access that granted to other users in accessing the resources.	TC001: The name list is not displayed. TC002: The name list is displayed, but the cross button is not displayed beside them. TC003: The name list with cross button is displayed.	TC001, TC002: Users cannot add or remove access for the other users. TC003: Users can manage the access successfully.	

5. Software Metrics

5.1 Product metrics

Product metrics are the quantitative measures that are used to access and evaluate the software's characteristics and performance throughout the whole software development cycle [1]. Product metrics help the development team better understand their project and keep improving the system since they offer insightful recommendations on a range of software product attributes, including efficiency, consistency, and reliability. Usually, product metrics are related to software features only and are categorized in two main classes which are dynamic metrics and static metrics. Dynamic metrics are collected by the measurements made from a program in execution, whereas static metrics are collected by measurements made from system representations such as design, programs or documentation [2].

There are some categories that will be considered in measuring the performance of the software product, which are provided in the following table:

No.	Categories	Description
1.	Code metrics	This is the measure that will be taken to measure size of code, which includes length of code and code complexity. The larger the number of code, the more complex the system and the possibility of error prompting also higher.
2.	Fan-in/fan-out	The number of functions that call another function is called fan-in which is referred to as X. The number of functions that function X calls is known as the fan-out. A high fan-in value indicates that X is closely related to the other components of the design and that altering X will have significant ripple effects. A high fan-out number indicates that the called components' coordination requires a high level of overall control logic complexity.
3.	Cyclomatic complexity	Cyclomatic complexity controls the overall understandability of the project. The higher the cyclomatic complexity, the more time is needed to develop whole system and it is more complex.
4.	Depth of conditional nesting	This measures the total number and depth of nestings of the if statements in a program. The higher the index, the more difficult to understand the whole program and the depth of the if statements nesting is high.
5.	Fog index	The fog index measures the total length of words and sentences in the documentation of the project. The higher the fog index, the more difficult to understand the documentation.

5.2 Project metrics

Project metrics are used to monitor progress, adjust for overall project overflow, and assess overall project performance. Project metrics are used to monitor team productivity and give the development team a better understanding of what is actually completed or ongoing [4]. There are several project metrics that will be taken into account for measuring the overall performance, which are given in the following list:

No.	Categories	Description
1.	Development model	It is important for the team to decide which development model is the most suitable to develop the whole project as different projects have vary requirements and might suit to different development model. Example of development model is waterfall method and agile method.
2.	Time	Time is another metrics that will be considered in project development. The larger the project, the more time will be taken to complete the whole project.
3.	Cost	Cost will be another metrics in the measurement of project metrics. The cost will increase if the project is big, and many complex features are required.
4.	Schedule	It is necessary to make a schedule so that every member in the team contributes to the project. For example, milestones with several deadlines are provided to measure every team member's performance on the progress of the deliverables and their effectiveness.

5.3 Process metrics

Metrics used to assess the software development process from its inception to its completion are known as process metrics. Process metrics are typically used to track software dependability in order to learn more about how the intangible software product is created. The International Standard Organization (ISO) created a generic reference known as ISO-9000 Certification, or Quality Management Standards, which is the basis for which process metrics are referenced to [5]. Process metrics are gathered during the software development process; Boehm's Constructive Cost Model, which is applied for software project cost prediction, is one example of process metrics. There are several types of process metrics that will be used in the development process, listed in the following table:

No.	Categories	Description
1.	Static process metrics	Related to the defined process in software development
2.	Dynamic process metrics	Related to properties of process performance
3.	Process evolution metrics	Related to process of making adjustments over a period of time

6. System Screenshots

The following shows the possible system screenshots of my project.

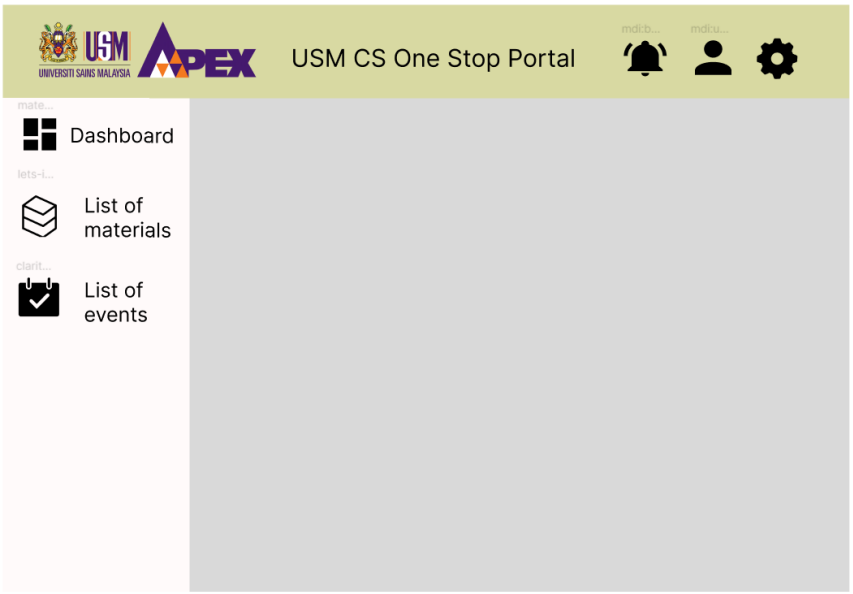

Figure 4: System Screenshot

7. Github Screenshots

The following shows the Github screenshots.

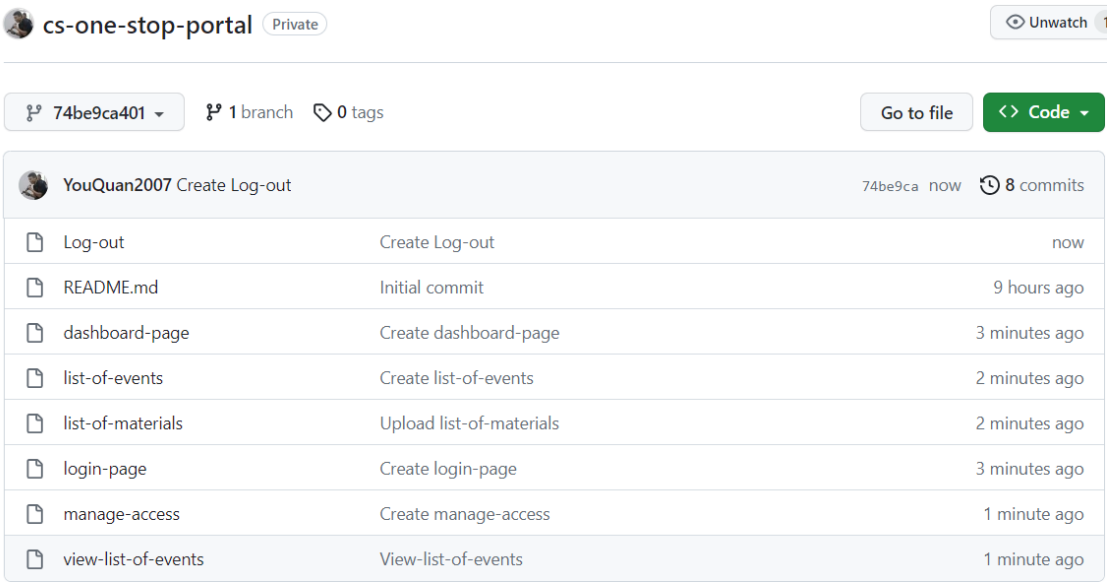

Figure 5: Github screenshot

8. Conclusion and future work

In conclusion, the creation of the CS One Stop Portal is a major step in improving the effectiveness with which all members of the CS community can obtain the necessary materials, including report templates, marking guideline, marking rubrics, and other pertinent information. Students and lecturers can quickly access the documents they need using this centralised one-stop gateway, eliminating the need to search through numerous emails, social media groups, Padlet link, and OneDrive links. Additionally, the portal has a notification system that allows all members to be informed of any updates. Also, without having to search through emails or groups, students may quickly check the list of contests or competitions they can participate in on the portal. By consolidating various information and resources on a centralized portal, I strongly believe that it can promote a more efficient information sharing between the CS community members. It enhances the convenience and accessibility of the CS community and promotes a more collaborative learning environment.

In future, I hope to implement more functionalities in the portal, such as integrating advanced data analytics and machine learning algorithms in the system, in which a personalized recommendations of information can be provided to all CS members. Analysis of the user behaviours is also a possible future work that can be accomplished, in which the portal can offer suggestions for the lecturers for relevant resources. I believe it can maximize the learning experience of every CS members if these idea are successfully implemented in the future.

References

[1] Dallaire, K. (2023, October 26). *Product metrics in software engineering*. Jellyfish. https://jellyfish.co/library/product-metrics-in-software-engineering/ (Accessed 13 December, 2023)

[2] GeeksforGeeks. (2022, July 18). *Product metrics in software engineering*. GeeksforGeeks. https://www.geeksforgeeks.org/product-metrics-in-software-engineering/ (Accessed 13 December, 2023)

[3] *The Agile Process 101: Understanding the Benefits of Using Agile Methodology*. Nvisia. (2020, September 16). https://www.nvisia.com/insights/agile-methodology (Accessed 13 December, 2023)

[4] Littlewood, W. (2023, December 5). 6 essential project metrics for agencies to track performance. The All In One Project Management Platform - Teamwork.com. https://www.teamwork.com/blog/project-metrics/ (Accessed 13 December, 2023)

[5] GeeksforGeeks. (2023, July 13). People metrics and process metrics in software engineering. GeeksforGeeks. https://www.geeksforgeeks.org/people-metrics-and-process-metrics-in-software-engineering/ (Accessed 13 December, 2023)

Appendix

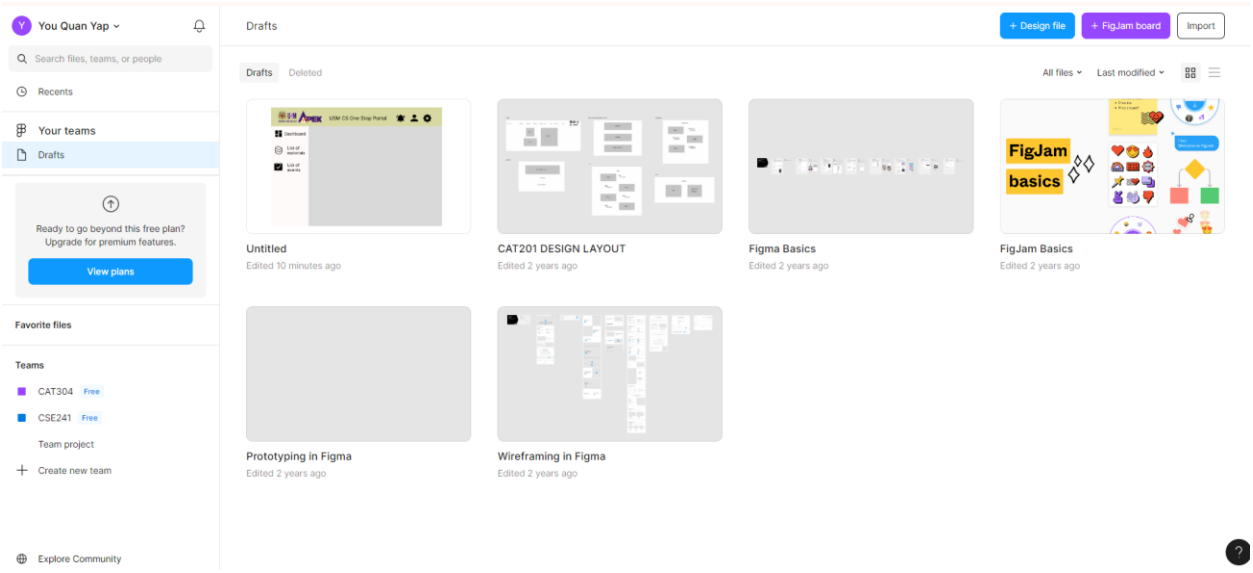

Figure 6: Figma platform is used in my design of system prototyping.