



## SECP2523 : DATABASE

SEMESTER 1 2023/2024

---

### DATABASE LOGICAL DESIGN & SQL FOR *<Youth Ventures Student Portfolio Management System (StuPort)>*

**Group Name:** EXPLORER

**Members:**

1. KOH LI HUI (GROUP LEADER)
2. KOH SU XUAN
3. LEE YIK HONG
4. VINESH A/L VIJAYAKUMAR

**Stakeholder:** Youth Ventures Asia

Representative Name	Date Interviewed
Mr. Hanif Marzuki Mohd Saupi	23/10/23
	8/11/23
Mr. Aniq Amrez	23/10/23

Printing Date: 7/2/2024

# TABLE OF CONTENTS

---

<b>CHAPTER 1.....</b>	<b>4</b>
Introduction.....	4
1.1 Overview about the company.....	4
1.2 Problems.....	5
1.3 Proposal.....	7
1.4 Definitions, Acronyms and Abbreviations.....	8
<b>CHAPTER 2.....</b>	<b>10</b>
Overview of Project.....	10
<b>CHAPTER 3.....</b>	<b>12</b>
Database System Requirements (Updated Business Rule).....	12
3.1 System Overview.....	12
3.2 Module 1 : Authentication.....	13
3.2.1 Module Description.....	13
3.2.2 Transaction Requirement: Data Entry.....	14
3.2.3 Transaction Requirement: Data Update/Deletion.....	14
3.2.4 Transaction Requirement: Data Queries.....	14
3.2.5 Local Conceptual Data Model.....	15
3.3 Module 2 : Profile.....	16
3.3.1 Module Description.....	16
3.3.2 Transaction Requirement: Data Entry.....	16
3.3.3 Transaction Requirement: Data Update/Deletion.....	16
3.3.4 Transaction Requirement: Data Queries.....	16
3.3.5 Local Conceptual Data Model.....	17
3.4 Module 3 : Dashboard.....	18
3.4.1 Module Description.....	18
3.4.2 Transaction Requirements: Data Entry.....	18
3.4.3 Transaction Requirements: Data Update/Deletion.....	19
3.4.4 Transaction Requirements: Data Queries.....	19
3.4.5 Local Conceptual Data Model.....	19
3.5 Module 4 : Activity.....	20
3.5.1 Module Description.....	20
3.5.2 Transaction Requirement: Data Entry.....	22
3.5.3 Transaction Requirement: Data Update/Deletion.....	22

3.5.4 Transaction Requirement: Data Queries.....	22
3.5.5 Local Conceptual Data Model.....	23
3.6 Module 5 : Registration.....	24
3.6.1 Module Description.....	24
3.6.2 Transaction Requirement: Data Entry.....	25
3.6.3 Transaction Requirement: Data Update/Deletion.....	25
3.6.4 Transaction Requirement: Data Queries.....	25
3.6.5 Local Conceptual Data Model.....	25
3.7 Module 6 : Feedback.....	26
3.7.1 Module Description.....	26
3.7.2 Transaction Requirements: Data Entry.....	26
3.7.3 Transaction Requirements: Data Update/Deletion.....	26
3.7.4 Transaction Requirements: Data Queries.....	26
3.7.5 Local Conceptual Data Model.....	27
3.8 Module 7 : Reward.....	28
3.8.1 Module Description.....	28
3.8.2 Transaction Requirement: Data Entry.....	28
3.8.3 Transaction Requirement: Data Update/Deletion.....	28
3.8.4 Transaction Requirement: Data Queries.....	28
3.8.5 Local Conceptual Data Model.....	28
3.9 Module 8 : Resume.....	29
3.9.1 Module Description.....	29
3.9.2 Transaction Requirement: Data Entry.....	29
3.9.3 Transaction Requirement: Data Update/Deletion.....	29
3.9.4 Transaction Requirement: Data Queries.....	29
3.9.5 Local Conceptual Data Model.....	30
3.10 Module 9 : Personal Activity.....	31
3.10.1 Module Description.....	31
3.10.2 Transaction Requirement: Data Entry.....	33
3.10.3 Transaction Requirement: Data Update/Deletion.....	34
3.10.4 Transaction Requirement: Data Queries.....	34
3.10.5 Local Conceptual Data Model.....	35
<b>CHAPTER 4.....</b>	<b>36</b>
<b>Database Conceptual Design.....</b>	<b>36</b>
4.1 Global Conceptual Design.....	36

4.2 Verified Global Conceptual Design.....	37
Database Logical Design.....	39
4.3 Logical Data Model (Logical ERD).....	39
4.4 Updated Data Dictionary.....	40
4.5 List of Normalized Global DB Relational Schema.....	46
CHAPTER 5.....	48
SQL Statements (DDL & DML).....	48
Module 1: Authentication.....	48
Module 2: Profile.....	50
Module 3: Dashboard.....	55
Module 4: Activity.....	58
Module 5: Registration.....	63
Module 6: Feedback.....	65
Module 7: Reward.....	67
Module 8: Resume.....	70
Module 9: Personal Activity.....	71
CHAPTER 6.....	76
Summary.....	76
CHAPTER 7.....	77
Appendices.....	77
7.1 Use Case Diagram for The System.....	77
7.2 Activity Diagrams for Every Use Case Identified.....	78
7.3 Meeting Log.....	104
Log Book #1.....	104

# CHAPTER 1

## Introduction

### 1.1 Overview about the company



Figure 1.1.1: Youth Ventures Company's Logo

In this project, we are having Youth Ventures Asia, a pioneering organization dedicated to empowering passionate youth across Southeast Asia which was founded in late 2018 as our main client in developing a new system. With a vision to make these young individuals competitive and globally relevant, this organization is on a mission to create a sustainable ecosystem where the dreams and aspirations of the youth can thrive. Mr. Hanif Marzuki Mohd Saipi, the client representative for Youth Ventures Asia, is a staunch advocate for youth development and envisions a future where the organization's mission aligns perfectly with addressing the challenges of easily finding more relevant connections and job opportunities. Both our client and company can be reached at +6017-750-1827. Generally, the organization had been grappling with a host of operational challenges, such as the intricate management of multiple program cycles, the arduous compilation of data from diverse databases, and the consistent recognition of student achievements across varying cycles. Therefore, implementing a new or improved system is crucial for the company to foster a more supportive academic management platform, help students to have systematic

management and empower administrators and partners of Youth Ventures to manage more effective programs and activities.

## **1.2 Problems**

### **What does the current system do**

The current system in place, referred to as the Youth Ventures As-Is System, is characterized by manual and labor-intensive processes for managing student profiles and program data. This system relies on several inefficient practices, leading to various challenges:

#### **1. Manual Data Entry:**

Student profiles are managed through manual data entry. Information for each student needs to be input individually across different program cycles.

#### **2. Multiple Databases:**

The system relies on multiple databases for storing student information. Cross-referencing student achievements and milestones across various program cycles is challenging due to the decentralized nature of data storage.

#### **3. Recognition of students achievements:**

Difficulties in recognizing and acknowledging student achievements arise from the decentralized nature of data storage and the manual management of profiles.

#### **4. Collaboration between parties:**

The absence of a streamlined data sharing mechanism hampers efficient collaboration among clients, partners, and stakeholders.

#### **5. Operation of student data:**

The overall operational efficiency is compromised due to the manual nature of data management. The lack of a centralized platform for information further exacerbates inefficiencies.

## **Challenges Faced**

1. Inefficiencies and Errors:
  - Manual data entry leads to errors, inefficiencies, and a time-consuming process.
2. Difficulty in Tracking Accomplishments:
  - Challenges in tracking and recognizing student accomplishments due to the fragmented nature of data.
3. Limited Collaboration:
  - Lack of a streamlined data sharing mechanism hinders effective collaboration among stakeholders.
4. Operational Inefficiency:
  - Manual processes contribute to overall operational inefficiency.

The identified challenges prompted the organization to seek a transformative solution to streamline operations, enhance data management, and improve collaboration among stakeholders. The AI-IS system aims to address the manual and fragmented nature of the previous system, providing a more efficient, integrated, and collaborative solution for managing student profiles and program data.

### **1.3 Proposal**

Our proposed system, StuPort, is designed to revolutionize the operational framework of Youth Ventures Asia by addressing the current manual and fragmented system. StuPort, a web-based Student Portfolio Management System, aims to streamline data collection and administration for students, administrators, and master administrators. The key objectives include enhancing student management through streamlined processes, centralizing data for improved retrieval, fostering collaboration with a streamlined data-sharing mechanism, and significantly boosting overall operational efficiency.

StuPort comprises several modules to cater to specific user roles and functionalities. The Authentication Module establishes tailored interactions and permissions, including password and email management. The Profile Module facilitates seamless creation, viewing, editing and deletion of profiles, ensuring efficient user control. The Dashboard Module provides customized dashboards for Master Admins, Admins, Lecturers and Students, offering a comprehensive system overview. The Activity Module followed by Personal Activity Module enables the management of various activities, allowing students to add personal activities and administrators to validate and publish events. The Registration Module is crucial for user onboarding, allowing users to register for activities under the oversight of Master Admins.

The Feedback Module plays a vital role in communication, enabling students to provide feedback on activities. Master administrators create feedback forms, and the feedback details are transparently displayed on student profiles, fostering improvement. The Reward Module manages the generation and retrieval of badges to acknowledge student accomplishments, promoting recognition and motivation. Lastly, the Resume Module empowers students to edit and download their resumes, ensuring streamlined personal information and experience management.

By implementing StuPort, Youth Ventures Asia can overcome current challenges, streamline operations, and create a more supportive academic management platform. The system aligns with the organization's vision of empowering passionate youth and fostering a sustainable ecosystem for their dreams to thrive. StuPort stands as a transformative solution

poised to enhance data management, collaboration, and overall operational efficiency for Youth Ventures Asia.

## 1.4 Definitions, Acronyms and Abbreviations

Number	Acronym(s)/Abbreviation(s)	Definition
1	StuPort	Youth Ventures Student Portfolio Management System
2	Admin	Administrator
3	Master Admin	Master Administrator

Student - One of the targeted main users of the system, they are students that are using the Youth Ventures system to manage their academic experiences.

Client/ Partner/ Admin/ Lecturer - One of the targeted users of the system, they are the organizations, government agencies, private companies, universities etc. that collaborate with Youth Ventures to manage their published activities and student's personal activities.

Master Administrator - One of the main users of the system, they are staff in Youth Ventures who will manage the system and the system database.

Profile - One of the entities in the system. Profile consists of personal information including name, gender, age, date of birth etc. will be filled by students while organization information will be filled by Admin.

Registration - One of the entities in the system. Registration links will be posted by Admin to be filled by the students who want to register the activity published by the Admin.

Feedback - One of the entities in the system. Feedback link form will be created by Master Admin to be filled by students who want to give rating and feedback to the activity

participated. Meanwhile, the organizer of the activity or related party will be notified and have the access to verify the student's activity.

## **CHAPTER 2**

### **Overview of Project**

As a Year 2 data engineering student, we have a project for our courses this semester. This project is a collaboration of three Work Based Learning courses which are SECP2523, SECP2613 and SECP3723. In this project, we are required to develop a database application solution to enable the proposed system, “StuPort” will be implemented smoothly based on the client’s requirements. We have in-depth interviews with Mr. Hanif Marzuki Mohd Saipi, a client representative from Youth Ventures Asia, and collaboration with other groups to ensure a holistic understanding of the project's goals and system requirements during Industry Day 1.

Our proposed system, the Youth Ventures Student Portfolio Management System named “StuPort” embarks on a transformative journey from a manual and labor-intensive operation to a sophisticated, automated platform designed to enhance the management of student profiles and program data. The existing system was grappled with challenges such as manual data entry, decentralized databases, and operational inefficiencies. In response to these limitations, the proposed system envisions a comprehensive solution integrating a centralized database system, an achievement recognition mechanism, and a user-friendly data-sharing portal.

The mission of the to-be system is clear: to efficiently collect and manage student information, empower administrators, and create a collaborative environment for students, master administrators, and partners of Youth Ventures. By introducing modules like authentication, profile management, user registration, activity management, personal activity validation, feedback systems, resume creation, reward systems, and an interactive dashboard, the system aims to provide a seamless experience for all stakeholders.

The proposed system's scope encompasses three key user types: students, master administrators, and partners or clients of Youth Ventures. Each user type has distinct roles and permissions, contributing to a dynamic ecosystem where individuals can actively participate, contribute, and be recognized.

Technologically, the system leverages a robust stack comprising PHP, MySQL, HTML, CSS, JavaScript, and Bootstrap. This combination ensures a secure, dynamic, and user-friendly platform, accessible across different devices. GitHub is employed for version control, facilitating collaboration among developers during the system's design and development phases.

In essence, the Youth Ventures Student Portfolio Management System is poised to revolutionize the management of student profiles, streamline data processes, and foster collaboration among stakeholders. Through its innovative features and technology stack, the system aims to empower students, lecturers, administrators, and master administrators, creating a holistic and impactful environment for personal and professional development.

## **CHAPTER 3**

### **Database System Requirements (Updated Business Rule)**

#### **3.1 System Overview**

StuPort, the Student Portfolio Management System developed for Youth Ventures Asia, is a web-based solution designed to streamline the collection and administration of student information. With a focus on User Authentication and Profile Management, Dashboard Presentation, Activity Management and Registration Handling, Feedback Handling and Reward Allocation and Resume Management, StuPort aims to empower master administrators, administrators, students and lecturers. It does not handle financial transactions or replace academic platforms but rather enhances student management, centralizes data for easy retrieval and promotes better collaboration between Youth Ventures and its clients. By addressing the specific needs of master administrators, administrators and students, StuPort aims to achieve efficient student management, improved data management and enhanced collaboration, aligning with the objectives of Youth Ventures.

## **3.2 Module 1 : Authentication**

LEE YIK HONG is the person in charge of documenting this module.

### **3.2.1 Module Description**

#### **Log In:**

Roles: Master Admin, Admin, Lecturer, Student

Lecturer and Student: These users need to provide valid credentials (Username and Password) that they entered during the signup process to access the system. Upon successful login, they gain access to their respective dashboards and profiles. The system verifies the credentials against the stored data in the profile database.

Admin and Master Admin: These roles have predefined credentials provided by Youth Venture Asia. They can directly log in without going through the signup process. This is implemented to maintain control over admin access and prevent unauthorized registration as an admin.

Different Functionality for Admin and Master Admin: Once logged in, Admin and Master Admin have access to extensive data management functionalities within the system, allowing them to oversee and manage various aspects of the platform.

#### **Sign Up:**

Roles: Lecturer, Student

The Sign Up functionality is limited to Lecturers and Students. Upon reaching the signup page, users are prompted to input the following details: Username, Full Name, Email, User Role (Lecturer/Student), Password, Confirm Password, Address, Course, Institution, Telephone, and Race. The entered information is then integrated into the profile database. Users are now registered in the system with their designated roles and can access their profiles and dashboards using the provided credentials. The information provided during

signup is also displayed in the user's profile, ensuring transparency and clarity about their details within the system.

### **Log Out:**

Roles: Master Admin, Admin, Lecturer, Student

The Log Out functionality is universal for all roles in the system. Users can perform the log-out action from their respective dashboards or profiles. Upon logging out, the system terminates the user's session, ensuring security and preventing unauthorized access. Users will need to re-enter their credentials to log in again. This functionality provides a secure and controlled exit from the system for all actors, maintaining the integrity of user sessions and protecting sensitive data.

### **3.2.2 Transaction Requirement: Data Entry**

1. Enter the User's Password.
2. Enter the User's Email.
3. Enter the User's Name.
4. Enter the User's Role.
5. Enter the User's Registration Information.

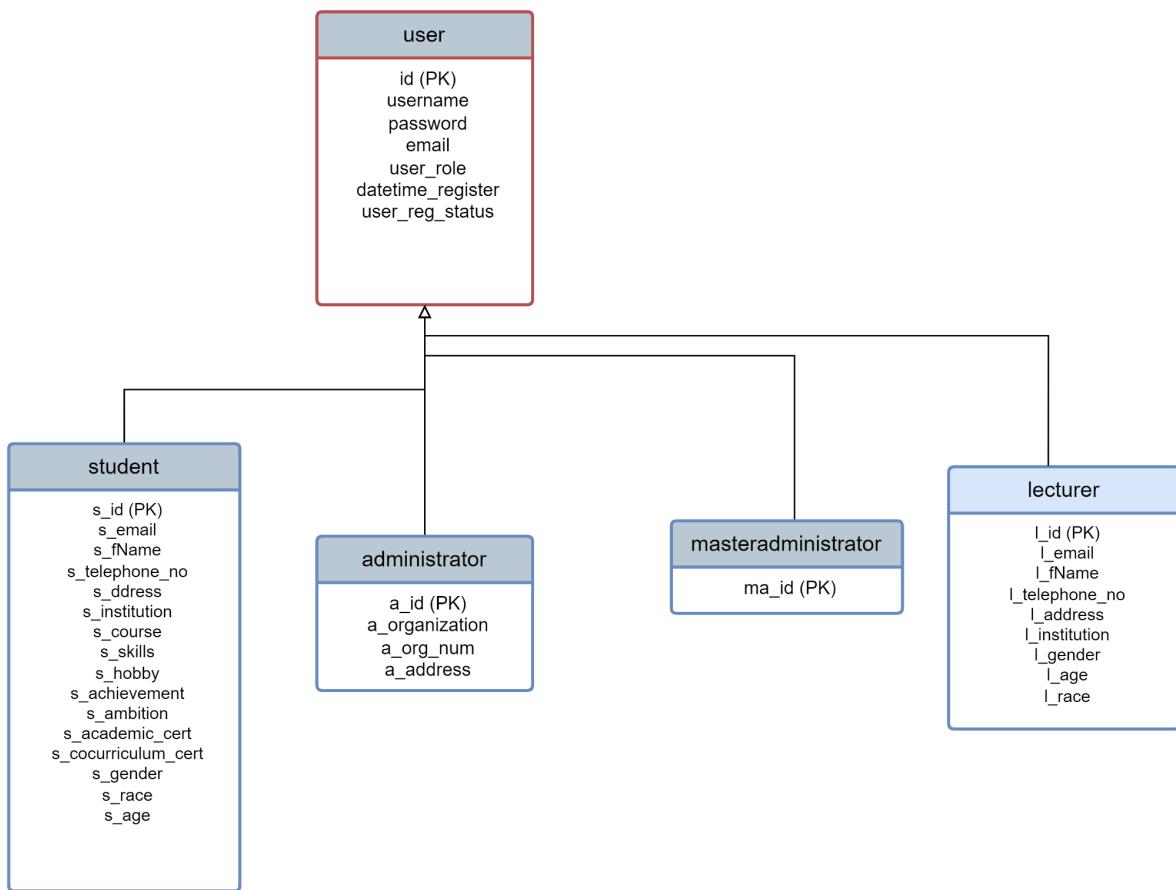
### **3.2.3 Transaction Requirement: Data Update/Deletion**

1. Update/Delete the User's Password.
2. Update/Delete the User's Email.
3. Update/Delete the User's Name.
4. Update/Delete the User's Role.
5. Update/Delete the User's Registration Information.

### **3.2.4 Transaction Requirement: Data Queries**

1. List the Users with a specific role.
2. List the User IDs, Usernames, or Emails for authentication purposes.

### 3.2.5 Local Conceptual Data Model



### **3.3 Module 2 : Profile**

KOH SU XUAN is the person in charge of documenting this module.

#### **3.3.1 Module Description**

The Profile Module allows users including Students and Lecturer to seamlessly create, view, edit and delete profiles. When the Student and the Lecturer newly sign up and first login into the system, they will be directed to create a profile which aims to input their personal details. The two types of users are then able to view their own profile and edit their own profile information. The Administrator, Student and Lecturer also have access to view other students' and lecturers' profiles by a search feature implemented in the Dashboard section to help to find specific student or lecturer profiles efficiently. When the Student and the Lecturer have decided to quit the system forever, they have the option to delete their own profile in which after the delete confirmation, their profiles and related data in the system will be permanently removed. Master Administrator not only can view the user profiles by using the search feature but is also able to edit and delete the user profile by accessing the database of the system. This module ensures a smooth experience for managing user profiles and user control.

#### **3.3.2 Transaction Requirement: Data Entry**

1. Enter the Student's or Lecturer's personal details.

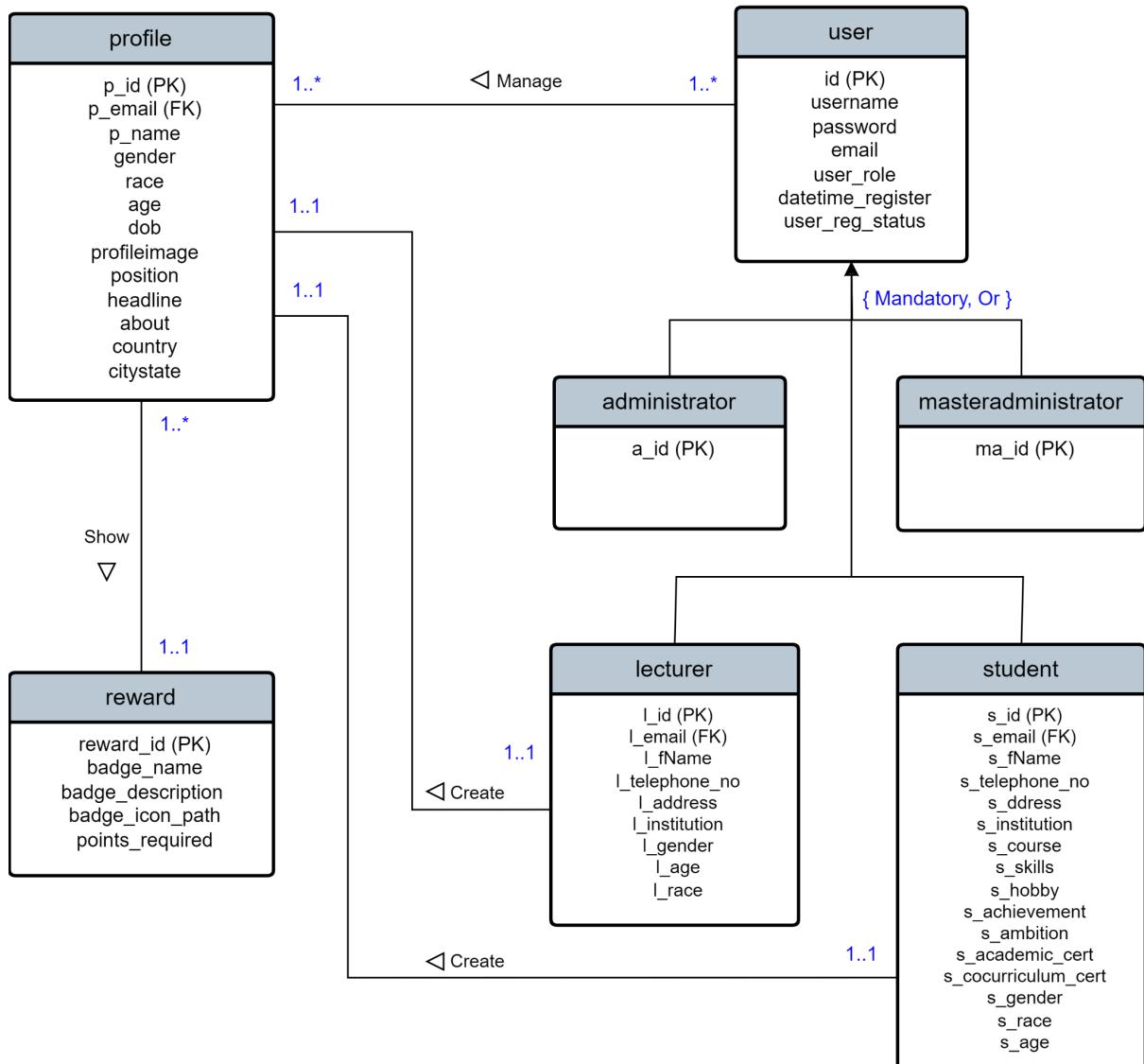
#### **3.3.3 Transaction Requirement: Data Update/Deletion**

1. Update/Delete the Student's or Lecturer's personal details.
2. Update/Delete the Student's or Lecturer's profile.

#### **3.3.4 Transaction Requirement: Data Queries**

1. List the Student's or Lecturer's personal details.
2. List the searching result that most matched with the keyword.
3. List the Student's or Lecturer's profile information.

### 3.3.5 Local Conceptual Data Model



## **3.4 Module 3 : Dashboard**

KOH LI HUI is the person in charge of documenting this module.

### **3.4.1 Module Description**

The Dashboard module serves as a centralized hub catering to Master Admins, Admins, Lecturers and Students, offering customized insights based on their roles. Upon navigating to the dashboard section, users, irrespective of their designation, prompt the system to retrieve data from connected sources and compile relevant information. For Students, the generated dashboard features convenient shortcuts to profiles and rewards, providing a comprehensive view of their details. Master Admins gain access to an abundance of information, including student details, admin insights, and overall system data, enabling them to oversee and manage the entire platform. Admins benefit from activity statistics, encompassing registration and feedback data, supporting well-informed decision-making. The Dashboard module, with its role-specific data presentation, ensures a user-centric experience, empowering each user type with information crucial to their responsibilities and objectives within the system.

This module involves all users of the system. As parents log in, they are presented with their own dashboard, the interface of which depends on their identity—whether they are a Master Admin, Admin, Lecturer or Student. The Master Admin dashboard displays the entire registered users information database, and system database, including activities statistics data, feedback, and registration data. The Master Admin has full authority over 'StuPort' and complete access to every database related to the system. In the default case, the admin dashboard displays activities statistical data, including registration and feedback data. The student dashboard shows the student's own profile and reward data, such as badges. Additionally, the dashboard includes basic features like searching, notifications, and settings, providing a comprehensive view for all users.

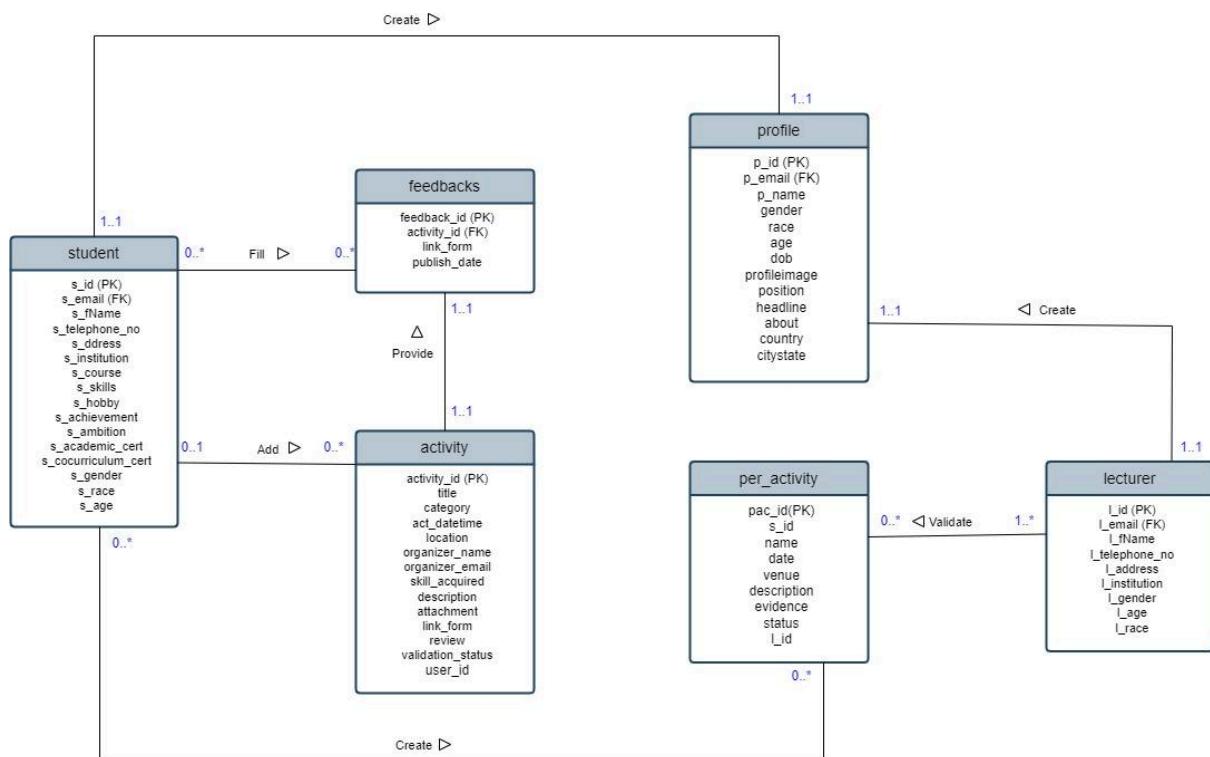
### **3.4.2 Transaction Requirements: Data Entry**

### 3.4.3 Transaction Requirements: Data Update/Deletion

### 3.4.4 Transaction Requirements: Data Queries

1. List the total number of registered students, lecturers and admins (clients or partners of Youth Ventures).
2. List the student's main details of personal activity.
3. List the six categories of activities published in the system.
4. List the profile details of all registered students and lecturers.

### 3.4.5 Local Conceptual Data Model



## **3.5 Module 4 : Activity**

LEE YIK HONG is the person in charge of documenting this module.

### **3.5.1 Module Description**

#### **Add YV Activity:**

Roles: Master Admin, Admin

This module allows Master Admin and Admin to add new activities to the system. By clicking on the "Create" button, users navigate to a form page where they can input details such as Activity Title, Description, Category (choosing from options like Competition/Scholarship, Program/Activities, Bootcamp/Workshop, Part-time, Volunteering, Internship), Date and Time, Location, Organizer Name, Skill Acquired, and Attachment. After submitting the form, the entered information is inserted into the 'activity' database table. Users are then redirected to the "Manage Activities" page, where all added activities are displayed in a table format. This page is also integrated with the feedback module. If there are inaccuracies in the activity details, users can click the "Update" button in the Action column. This action navigates them to an update form identical to the "Add Activity" form. After making edits, clicking "Submit" updates the inputs in the 'activity' database table. The user is then redirected back to the "Manage Activities" page, and the modified activity is updated in the respective row. To remove an activity entirely, users can click the "Delete" button. A confirmation message appears, and upon confirming deletion, the selected activity row is removed from the 'activity' table. Clicking "Close" returns users to the "Manage Activities" page without making any changes.

#### **View YV Activity:**

Roles: Master Admin, Admin, Student

Master Admin, Admin, and Student roles can view YV activities in a table format on the "Manage Activities" page. The table displays columns for Activity Title, Description, Category, Date and Time, Location, Organizer Name, Skill Acquired, and Attachment. Master Admin and Admin have an additional "Action" column allowing them to manage

activities, while the Student role only allows viewing without any management actions. This page provides a comprehensive overview of all activities created by Master Admin and Admin. The table format ensures easy navigation and understanding of each activity's details. For Students, this serves as an informative view-only page, allowing them to stay informed about upcoming activities without the ability to edit or delete, but they can register the activity that was created by Master Admin and Admin which because of it integrated with module Registration.

### **Edit YV Activity:**

Roles: Master Admin, Admin

The "Edit YV Activity" functionality is seamlessly integrated into the "Manage Activities" page. Users can click the "Update" button in the Action column next to the activity they wish to edit. The update action navigates users to a form page identical to the "Add Activity" form. This allows Master Admin and Admin to modify the existing details of the selected activity. After submitting the updated information, the changes are reflected in the 'activity' database table. Users are then redirected back to the "Manage Activities" page, where the modified activity details are updated in the corresponding row.

### **Delete YV Activity:**

Roles: Master Admin, Admin

Users with Master Admin and Admin roles can delete activities by clicking the "Delete" button in the Action column on the "Manage Activities" page. Upon clicking "Delete," a confirmation message appears to ensure users intend to delete the selected activity. Confirming the deletion removes the entire row from the 'activity' database table. Clicking "Close" without confirming deletion returns users to the "Manage Activities" page, preserving the existing activities. This ensures a secure and intentional deletion process.

### **3.5.2 Transaction Requirement: Data Entry**

1. Enter the activity date and time.
2. Enter the activity title.
3. Enter the activity description.
4. Enter the activity organizers/sponsors.
5. Enter the activity location.
6. Choose the type of session.

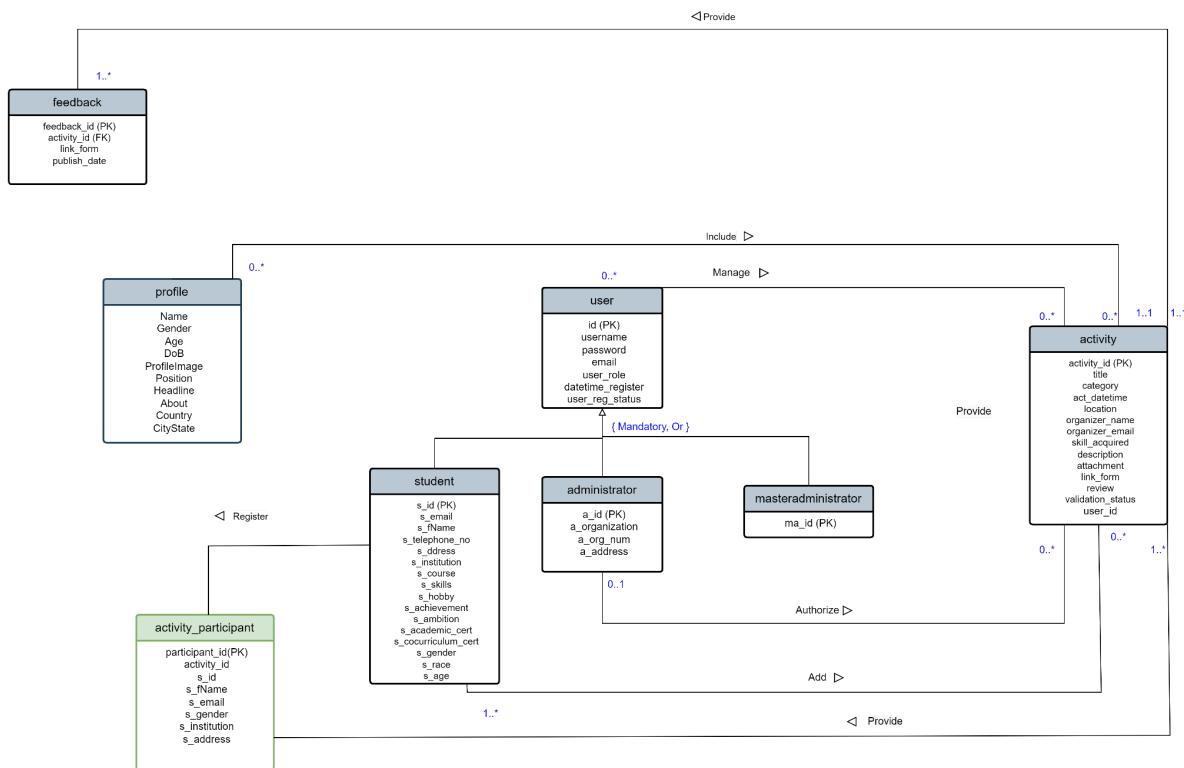
### **3.5.3 Transaction Requirement: Data Update/Deletion**

1. Update/Delete the activity date and time.
2. Update/Delete the activity title.
3. Update/Delete the activity organizers/sponsors.
4. Update/Delete the activity location.
5. Update/Delete the type of session.

### **3.5.4 Transaction Requirement: Data Queries**

1. List the published activities.
2. List the details of a specific activity.

### **3.5.5 Local Conceptual Data Model**



## **3.6 Module 5 : Registration**

LEE YIK HONG is the person in charge of documenting this module.

### **3.6.1 Module Description**

#### **Register YV Activity:**

Roles: Student

The "Register YV Activity" module allows students to register for activities created by Master Admin and Admin. On the "View Activities" page, students will find a "Register" button in the Action column for each activity. Clicking the "Register" button initiates the registration process. The button will change from red to a dull yellow color, indicating that the student has successfully registered for the activity. This color change also disables the button, preventing duplicate registrations. The registration information is stored in the 'activity\_participant' database table, where the 'activity\_id' (foreign key from the 'activity' table) is updated with the 's\_id' (Student's ID). This ensures a record of student participation in specific activities.

#### **View Registered YV Activity:**

Roles: Student

The "View Registered YV Activity" module provides students with a dedicated page to view all the activities they have successfully registered for. The layout of this page is similar to the "View Activities" page, displaying details such as Activity Title, Description, Category, Date and Time, Location, Organizer Name, Skill Acquired, and Attachment. The key difference lies in the "Action" column, where the status of the activity is indicated as 'Registered' for each registered activity. This page serves as a convenient reference for students to track their participation history, ensuring they have clear visibility into the activities they have registered for. It allows them to stay organized and engaged with their chosen activities.

The information on this page is derived from the 'activity\_participant' database table, where the 's\_id' matches the Student's ID, linking their profile to the registered activities.

### 3.6.2 Transaction Requirement: Data Entry

1. Information registered is directly concatenated to the profile.
2. Enter register YV activity

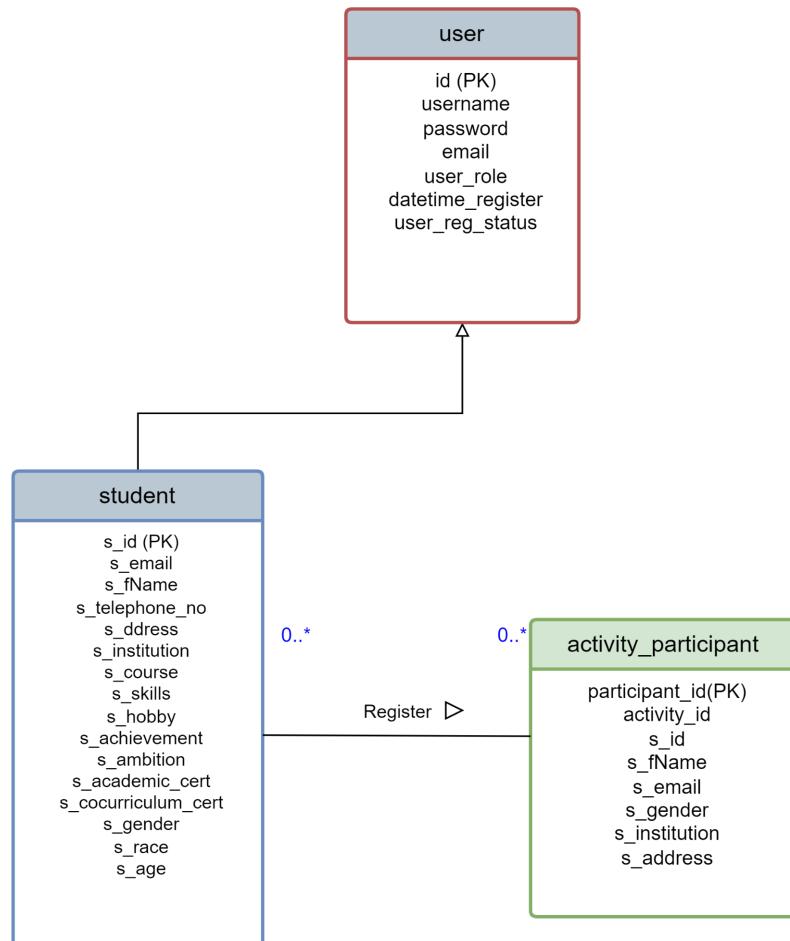
### 3.6.3 Transaction Requirement: Data Update/Deletion

-

### 3.6.4 Transaction Requirement: Data Queries

1. List of registered YV activities.

### 3.6.5 Local Conceptual Data Model



## **3.7 Module 6 : Feedback**

KOH LI HUI is the person in charge of documenting this module.

### **3.7.1 Module Description**

In the Feedback module, both Master Admins and students engage in seamless processes tailored to their roles. For Master Admins, selecting an activity from the list initiates the creation of feedback questions through a user-friendly link, culminating in the publication of the feedback form to students. On the student side, within the navigation bar, smooth navigation to the activity area reveals a comprehensive list of registered or attended activities. By selecting a specific activity, students access a feedback form through a provided link, offering valuable insights into their activity experience. Additionally, Master Admins and Admins can access a detailed overview of activity feedback. Navigating to the activity section on the dashboard, they select the feedback link for specific activities and review student feedback. This extensive Feedback module ensures a user-centric approach, facilitating constructive communication and enhancing the overall experience for both administrators and students alike.

### **3.7.2 Transaction Requirements: Data Entry**

1. Enter the feedback link to be published for the students.

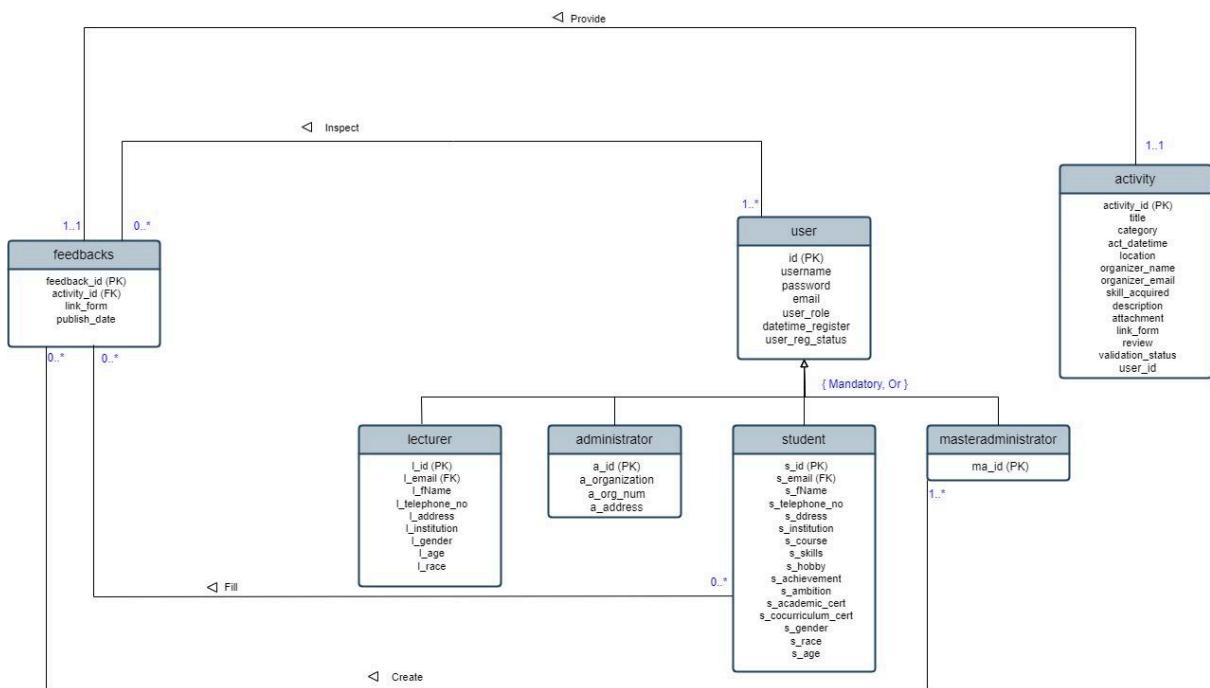
### **3.7.3 Transaction Requirements: Data Update/Deletion**

1. Update/Delete the feedback link.

### **3.7.4 Transaction Requirements: Data Queries**

1. List of the feedback link which used to fill in by the students for a certain activity.

### 3.7.5 Local Conceptual Data Model



## 3.8 Module 7 : Reward

VINESH is the person in charge of documenting this module.

### 3.8.1 Module Description

The reward management module is developed to manage the generation, alteration, and retrieval of badges in the Youth Ventures Asia system. It functions as a robust system for overseeing digital badges, granted to students to acknowledge their accomplishments in diverse events. The administrator of Youth Ventures Asia is empowered to generate badges and review the existing ones. Subsequently, these badges are assigned to students based on their specific achievements.

### 3.8.2 Transaction Requirement: Data Entry

1. Enter badge name.
2. Enter badge description.
3. Enter badge value.
4. Upload badge image.

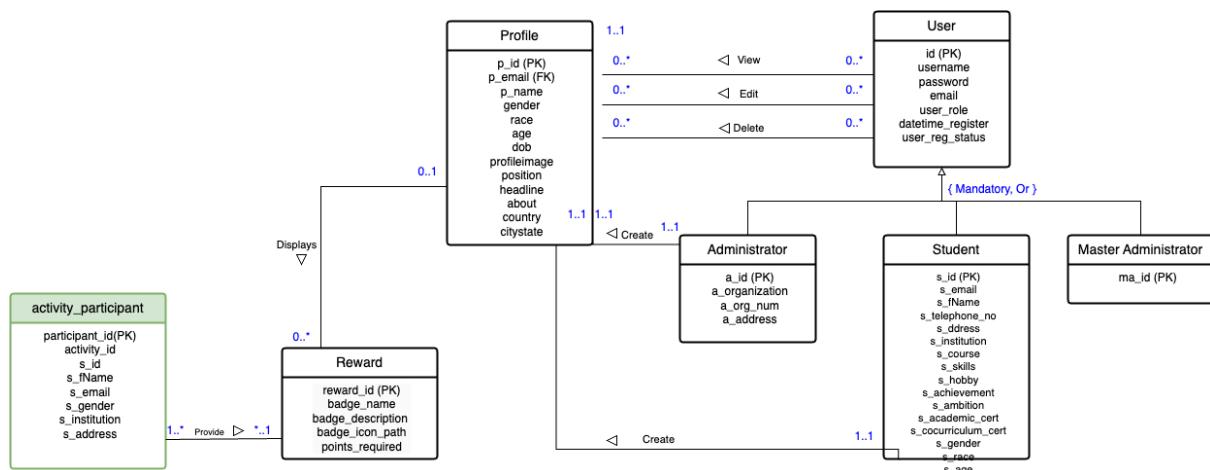
### 3.8.3 Transaction Requirement: Data Update/Deletion

1. Modify badge name.
2. Modify badge description.

### 3.8.4 Transaction Requirement: Data Queries

1. View list of available badges.

### 3.8.5 Local Conceptual Data Model



## **3.9 Module 8 : Resume**

KOH SU XUAN is the person in charge of documenting this module.

### **3.9.1 Module Description**

The Resume Module allows the student to download their master transcript within the system by including use cases of downloading master transcripts. Students have the ability to download their master transcript equipped with all personal information keyed into the system based on the choice of the student in which they will access via profile or system interface. This module ensures streamlined personal information and experience management in the form of documentation leading to effective master transcript handling within the system.

### **3.9.2 Transaction Requirement: Data Entry**

-

### **3.9.3 Transaction Requirement: Data Update/Deletion**

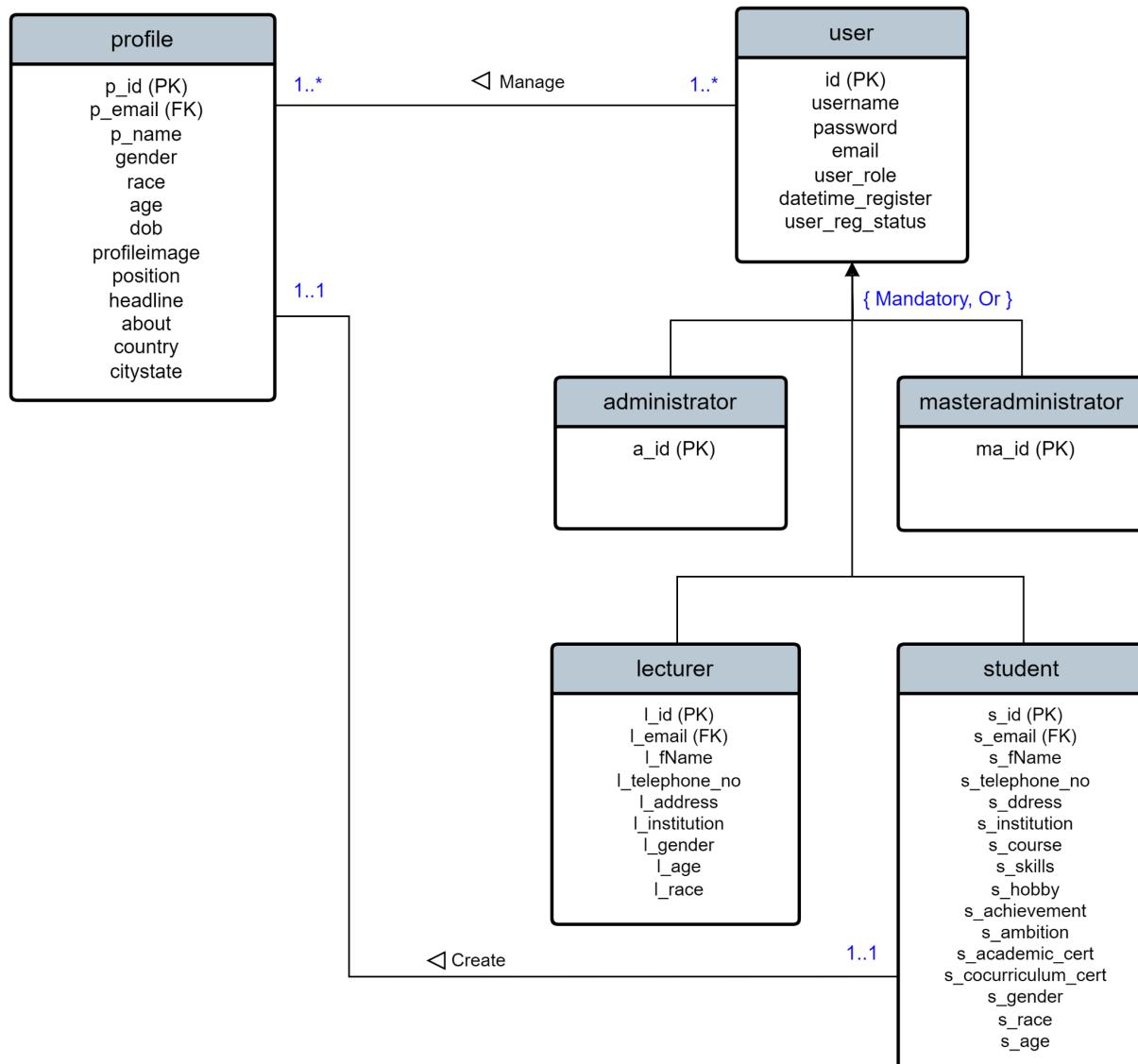
-

### **3.9.4 Transaction Requirement: Data Queries**

1. List the Student's master transcript content.

### 3.9.5 Local Conceptual Data Model

Below are the entities involved in the resume module.



## **3.10 Module 9 : Personal Activity**

LEE YIK HONG is the person in charge of documenting this module.

### **3.10.1 Module Description**

#### **Add Personal Activity:**

Roles: Student

The "Add Personal Activity" module is designed for students to add their personal activities to the system. By clicking on the "Create" button, users navigate to a form page where they can input details such as Personal Activity Name, Description, Date, Venue, and Evidence (allowing uploads of PDF, JPG, PNG, etc.). After submitting the form, the entered information is inserted into the 'per\_activity' database table. Users are then redirected to the "Manage Personal Activities" page, where all added personal activities are displayed in a table format. If there are inaccuracies in the personal activity details, students can click the "Update" button in the Action column. This action navigates them to an update form identical to the "Add Personal Activity" form. After making edits, clicking "Submit" updates the inputs in the 'per\_activity' database table. The user is then redirected back to the "Manage Personal Activities" page, and the modified personal activity is updated in the respective row. To remove a personal activity entirely, users can click the "Delete" button. A confirmation message appears, and upon confirming deletion, the selected personal activity row is removed from the 'per\_activity' table. Clicking "Close" returns users to the "Manage Personal Activities" page without making any changes.

#### **View Personal Activity:**

Roles: Student, Lecturer

Both students and lecturers can view personal activities in a table format on the "Manage Personal Activities" page. The table displays columns for Personal Activity Name, Description, Date, Venue, Evidence, and Action. For students, the "Action" column includes buttons for adding, updating, deleting, and assigning personal activities to a lecturer. For

lecturers, the "Action" column includes buttons for approving or rejecting personal activities assigned to them. They can also view the details of the assigned personal activities.

### **Edit Personal Activity:**

Roles: Student

Similar to the "Add Personal Activity" functionality, students can edit their existing personal activities by clicking the "Update" button in the Action column on the "Manage Personal Activities" page. The update action navigates students to a form page identical to the "Add Personal Activity" form. This allows them to modify the existing details of the selected personal activity. After submitting the updated information, the changes are reflected in the 'per\_activity' database table. Students are then redirected back to the "Manage Personal Activities" page, where the modified personal activity details are updated in the corresponding row.

### **Delete Personal Activity:**

Roles: Student

Students can delete personal activities by clicking the "Delete" button in the Action column on the "Manage Personal Activities" page. Upon clicking "Delete," a confirmation message appears to ensure students intend to delete the selected personal activity. Confirming the deletion removes the entire row from the 'per\_activity' database table. Clicking "Close" without confirming deletion returns students to the "Manage Personal Activities" page, preserving the existing personal activities. This ensures a secure and intentional deletion process.

### **Assign Personal Activity:**

Roles: Student

Students can assign their personal activities to a specific lecturer by clicking the "Assign" button in the Action column on the "Manage Personal Activities" page. Upon clicking "Assign," a list of available lecturers is displayed, and students can select the desired lecturer. The 'Lecturer' column in the table is updated with the selected lecturer's name. The assigned lecturer receives the details of the personal activity for validation and approval. The 'Status' column initially shows "Waiting" until the lecturer approves or rejects the assignment.

### **Validate Personal Activity:**

Roles: Lecturer

Lecturers validate and approve personal activities assigned to them. Assigned personal activities are displayed in a table format on the "Manage Personal Activities" page for lecturers. Lecturers can view details of the assigned personal activities and either click "Approve" or "Reject" based on the validity of the evidence provided. Clicking "Approve" updates the 'Status' column in the 'per\_activity' database table to "Approved." Clicking "Reject" rejects the personal activity, and the status is updated accordingly.

### **View Approved Personal Activity List:**

Roles: Student

Students can view a list of their approved personal activities on a dedicated page. The list includes Personal Activity Name, Description, Date, Venue, Evidence, and a 'Status' column showing "Approved." This page provides students with a clear overview of the personal activities that have been successfully validated and approved by the assigned lecturer.

### **3.10.2 Transaction Requirement: Data Entry**

1. Enter the personal activity date.
2. Enter the personal activity title.

3. Enter the personal activity description.
4. Enter the personal activity venue.
5. Enter the personal activity evidence.

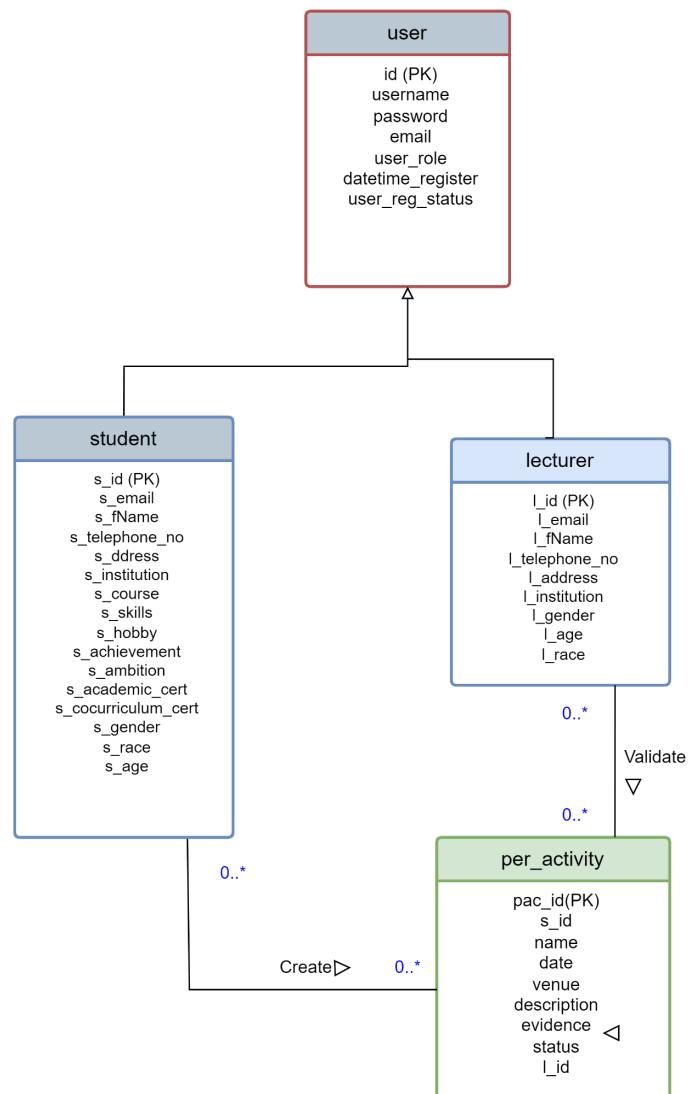
#### **3.10.3 Transaction Requirement: Data Update/Deletion**

1. Update/Delete the personal activity date.
2. Update/Delete the personal activity title.
3. Enter the personal activity description.
4. Update/Delete the personal activity venue.
5. Update/Delete the personal activity evidence.

#### **3.10.4 Transaction Requirement: Data Queries**

1. List the student's personal activities.
2. List the validated personal activities.
3. List the details of a specific personal activity.
4. List the activities requiring validation by the lecturer.

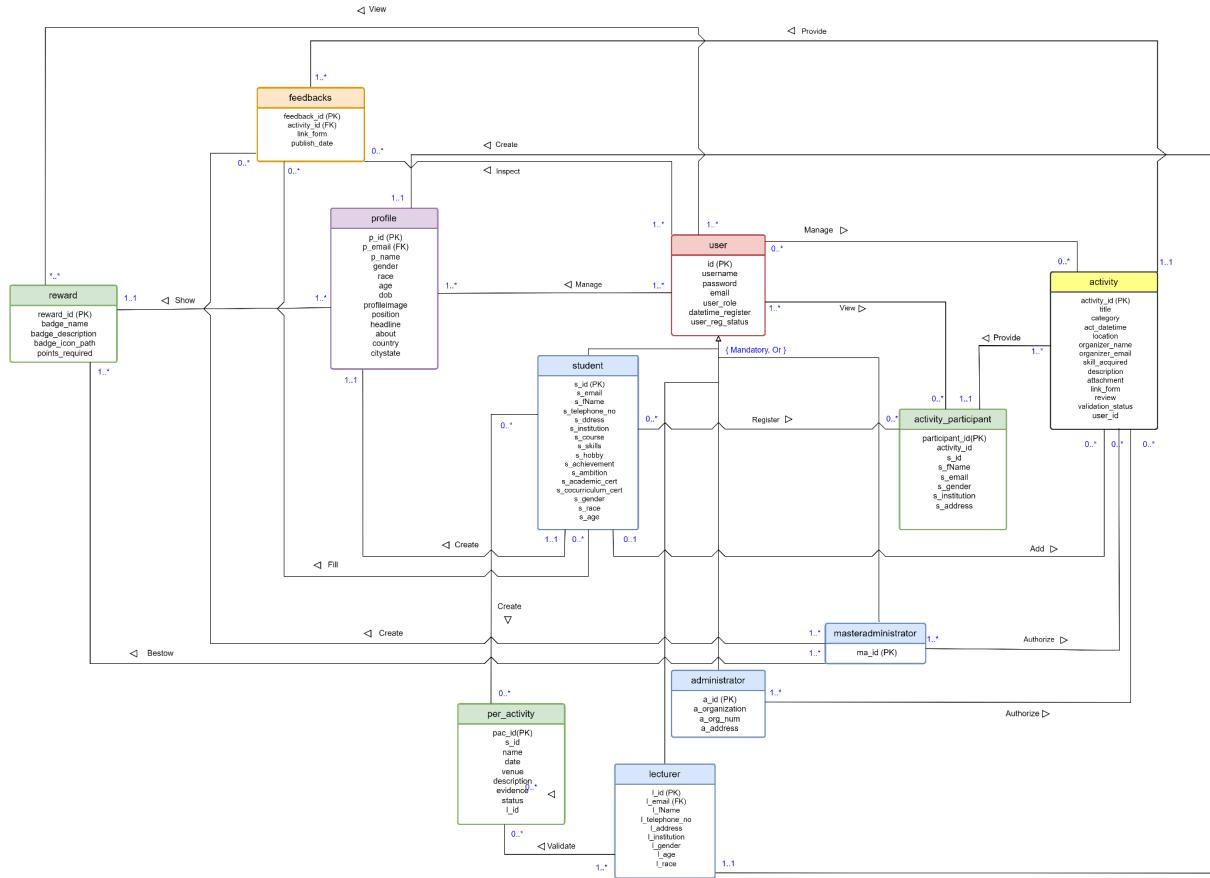
### 3.10.5 Local Conceptual Data Model



# CHAPTER 4

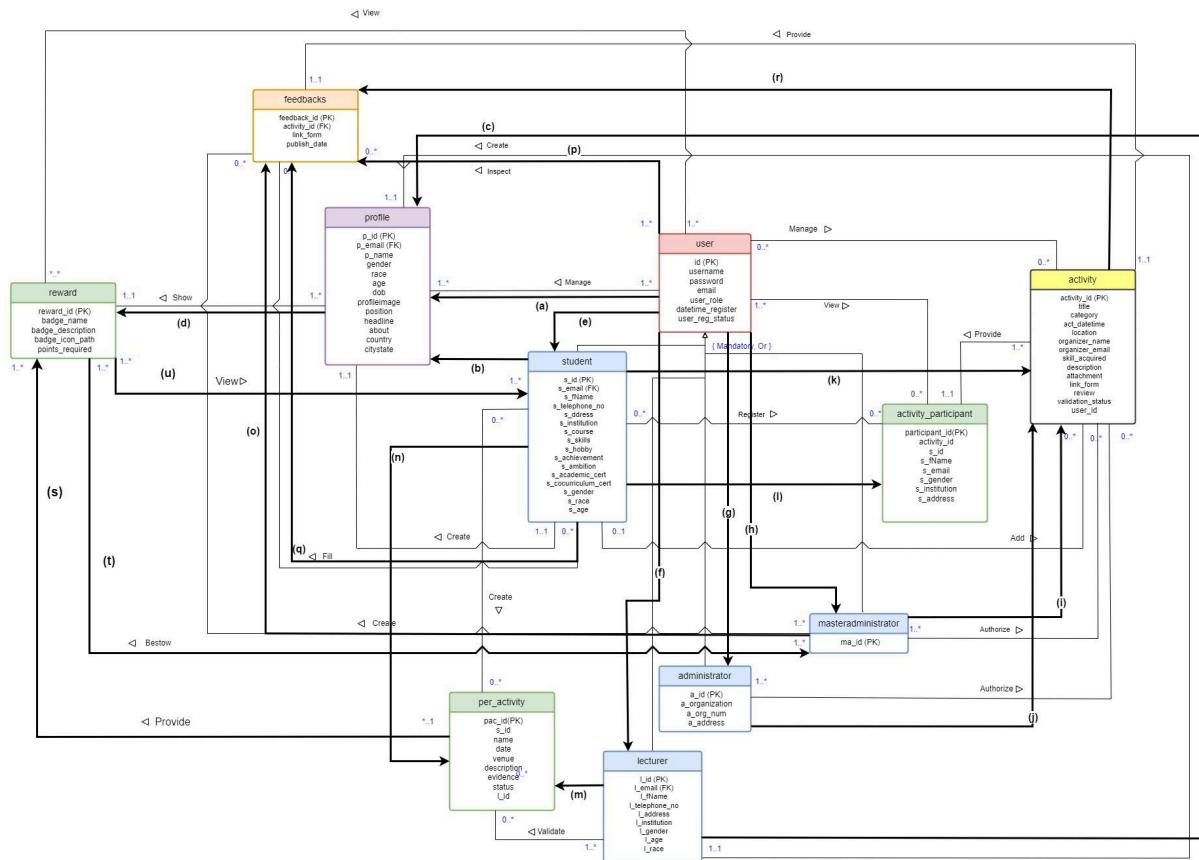
## Database Conceptual Design

### 4.1 Global Conceptual Design



\*\*Higher Definition Conceptual ERD: [Database-Updated Conceptual ERD.png](#)\*\*

## 4.2 Verified Global Conceptual Design



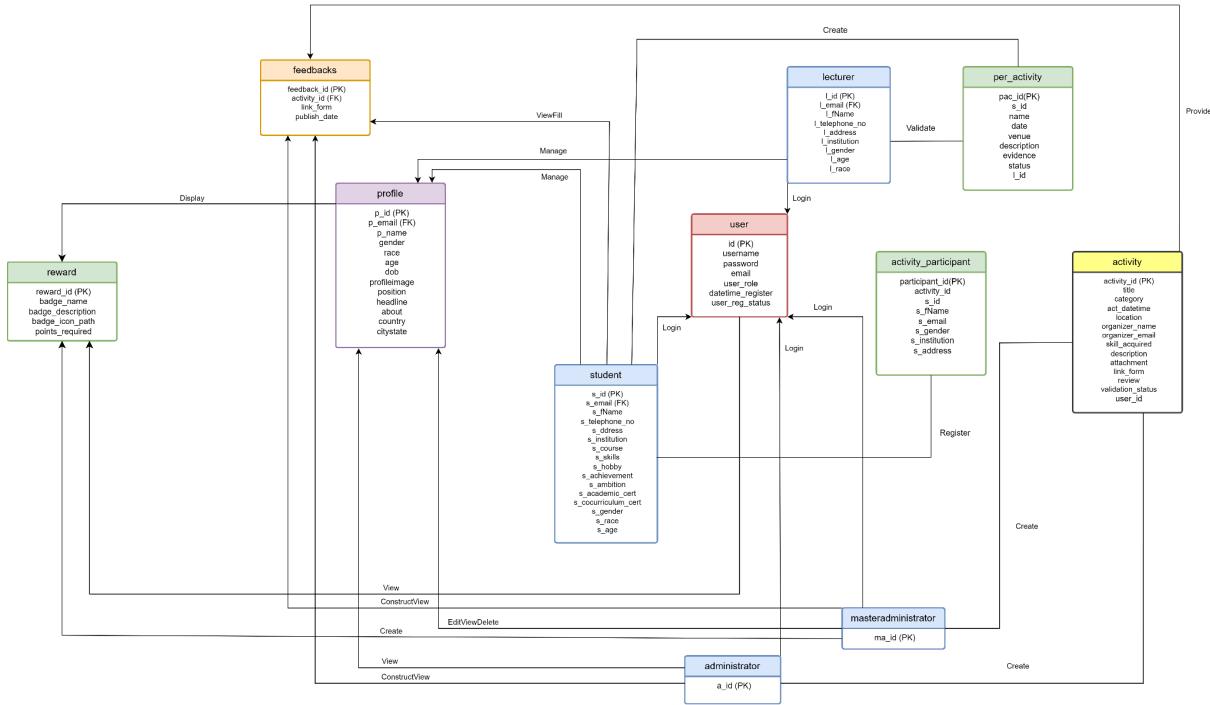
\*\*Higher Definition Verified Conceptual ERD: Database - Updated Verified ERD.jpg\*\*

Symbol	Transaction
a	3.3.3.1, 3.3.3.2, 3.3.4.1, 3.3.4.2, 3.3.4.3
b	3.3.2.1
c	3.3.2.1
d	3.3.4.1
e	3.2.2.1 , 3.2.2.2 , 3.2.2.3 , 3.2.2.4 , 3.2.2.5, 3.2.3.1 , 3.2.3.2 , 3.2.3.3 , 3.2.3.4 , 3.2.3.5, 3.2.4.1 , 3.2.4.2
f	3.2.2.1 , 3.2.2.2 , 3.2.2.3 , 3.2.2.4 , 3.2.2.5, 3.2.3.1 , 3.2.3.2 , 3.2.3.3 ,

	3.2.3.4 , 3.2.3.5, 3.2.4.1 , 3.2.4.2
g	3.2.2.1 , 3.2.2.2 , 3.2.2.3 , 3.2.2.4 , 3.2.2.5, 3.2.3.1 , 3.2.3.2 , 3.2.3.3 , 3.2.3.4 , 3.2.3.5, 3.2.4.1 , 3.2.4.2
h	3.2.2.1 , 3.2.2.2 , 3.2.2.3 , 3.2.2.4 , 3.2.2.5, 3.2.3.1 , 3.2.3.2 , 3.2.3.3 , 3.2.3.4 , 3.2.3.5, 3.2.4.1 , 3.2.4.2
i	3.5.2.1, 3.5.2.2, 3.5.2.3, 3.5.2.4, 3.5.2.5, 3.5.2.6, 3.5.3.1, 3.5.3.2, 3.5.3.3, 3.5.3.4, 3.5.3.5, 3.5.4.1, 3.5.4.2
j	3.5.2.1, 3.5.2.2, 3.5.2.3, 3.5.2.4, 3.5.2.5, 3.5.2.6, 3.5.3.1, 3.5.3.2, 3.5.3.3, 3.5.3.4, 3.5.3.5, 3.5.4.1, 3.5.4.2
k	3.5.2.1, 3.5.2.2, 3.5.2.3, 3.5.2.4, 3.5.2.5, 3.5.2.6, 3.5.4.1, 3.5.4.2
l	3.6.2.1, 3.6.2.2, 3.6.4.1
m	3.10.2.1, 3.10.2.2, 3.10.2.3, 3.10.2.4, 3.10.2.5, 3.10.2.6, 3.5.3.1, 3.5.3.2, 3.5.3.3, 3.5.3.4, 3.5.3.5, 3.5.4.1, 3.5.4.2, 3.5.4.3, 3.5.4.4
n	3.5.4.1, 3.5.4.2, 3.5.4.3, 3.5.4.4
o	3.7.2.1, 3.7.3.1
p	3.7.4.1
q	3.7.4.1
r	3.7.4.1
s	3.8.4.1
t	3.8.2.1, 3.8.2.2, 3.8.2.3 , 3.8.2.4 , 3.8.3.1 , 3.8.3.2
u	3.8.4.1

# Database Logical Design

## 4.3 Logical Data Model (Logical ERD)



\*\*Higher Definition Logical ERD: Database-Updated Logical ERD.png \*\*

## 4.4 Updated Data Dictionary

Entity Name	Attributes	Description	Data Type & Length	Nulls	Multi-valued
<b>user</b>	<b>id</b>	Unique identifier of user	int(11)	No	No
	<b>username</b>	Username of user used for login	varchar(255)	No	No
	<b>password</b>	Encrypted password of user used for login	varchar(255)	No	No
	<b>email</b>	Email of user	varchar(255)	No	No
	<b>user_role</b>	Role of user	varchar(50)	No	No
	<b>datetime_register</b>	Date Time of registration of user	datetime	No	No
	<b>user_reg_status</b>	Activation status of user	varchar(20)	No	No
<b>student</b>	<b>s_id</b>	Unique identifier of student	int(11)	No	No
	<b>s_email</b>	Email of the student	varchar(255)	No	No
	<b>s_fName</b>	Full name of the student	varchar(255)	No	No
	<b>s_telephone_no</b>	Telephone number of the student	varchar(20)	No	Yes
	<b>s_address</b>	Residential address of the student	varchar(255)	No	Yes
	<b>s_institution</b>	Institution of the student	varchar(255)	No	No

	<b>s_course</b>	Course registered by the student	varchar(255)	No	No
	<b>s_skills</b>	Skills of the student	varchar(255)	Yes	Yes
	<b>s_hobby</b>	Hobby of the student	varchar(255)	Yes	Yes
	<b>s_achievement</b>	Achievement made by the student	varchar(255)	Yes	Yes
	<b>s_ambition</b>	Ambition of the student	varchar(255)	Yes	Yes
	<b>s_academic_cert</b>	Academic certificate of the student	varchar(255)	Yes	No
	<b>s_cocurriculum_cert</b>	Co-curricular certificate of the student	varchar(255)	Yes	No
	<b>s_gender</b>	Gender of the student	varchar(255)	No	No
	<b>s_race</b>	Race of the student	int(11)	No	No
	<b>s_age</b>	Age of the student	varchar(255)	No	No
<b>lecturer</b>	<b>l_id</b>	Unique identifier of lecturer	int(11)	No	No
	<b>l_email</b>	Email of the lecturer	varchar(255)	No	No
	<b>l_fName</b>	Full name of the lecturer	varchar(255)	No	No
	<b>l_telephone_no</b>	Telephone number of the lecturer	varchar(15)	No	Yes
	<b>l_address</b>	Residential address of the lecturer	varchar(255)	No	Yes

	<b>l_institution</b>	Institution of the lecturer	varchar(255)	No	No
	<b>l_gender</b>	Gender of the lecturer	varchar(10)	No	No
	<b>l_race</b>	Race of the lecturer	varchar(255)	No	No
	<b>l_age</b>	Age of the lecturer	int(11)	No	No
<b>administrator</b>	<b>a_id</b>	Unique identifier of administrator	int(11)	No	
<b>masteradministrator</b>	<b>ma_id</b>	Unique identifier of master administrator	int(11)	No	No
<b>profile</b>	<b>p_id</b>	Unique profile information identifier of user	int(11)	No	No
	<b>p_email</b>	Email of the user as registered	varchar(20)	No	
	<b>p_name</b>	Name of the user shown in profile	varchar(50)	No	No
	<b>gender</b>	Gender of the user shown in profile	varchar(20)	No	No
	<b>race</b>	Race of the user shown in profile	varchar(20)	No	
	<b>age</b>	Age of the user shown in profile	int(11)	No	No
	<b>dob</b>	Date of Birth of the user shown in profile	date	Yes	No
	<b>profileimage</b>	Profile image of the user shown in profile	varchar(200)	Yes	No
	<b>position</b>	Position of the user shown in profile	varchar(50)	No	No

	<b>headline</b>	Headline of the user shown in profile	text	Yes	No
	<b>about</b>	About of the user shown in profile	text	Yes	No
	<b>country</b>	Country of the user in shown in profile	varchar(50)	No	No
	<b>citystate</b>	City and state of the user in shown in profile	varchar(50)	No	No
<b>activity</b>	<b>activity_id (PK)</b>	Unique serial number for each activity or event	int(11)	No	No
	<b>title</b>	Title of the activity	varchar(255)	No	No
	<b>category</b>	Type of the activity	varchar(50)	Yes	Yes
	<b>act_datetime</b>	Date and Time of the activity	datetime	Yes	Yes
	<b>location</b>	Location of the activity	varchar(255)	Yes	Yes
	<b>organizer_name</b>	Name of the organizer of the activity	varchar(255)	Yes	Yes
	<b>skill_acquired</b>	Skills gained from the activity by the student	varchar(255)	Yes	Yes
	<b>activity_desc</b>	Description of the activity	text	Yes	No
	<b>attachment</b>	Attachment of the activity	varchar(255)	Yes	Yes
	<b>validation_status</b>	The validation status for the activity attended by student whether valid or invalid	varchar(20)	Yes	Yes
	<b>link_form</b>	feedback form link	varchar(200)	Yes	Yes

	<b>review</b>	feedback review	text	Yes	Yes
	<b>user_id</b>	user's id	int(11)	Yes	No
<b>activity_participant</b>	<b>participant_id</b>	Unique serial number for each participant	int(11)	No	No
	<b>activity_id</b>	Unique serial number for each activity	int(11)	Yes	No
	<b>s_id</b>	students id	int(11)	Yes	No
	<b>s_email</b>	students email	varchar(255)	Yes	Yes
	<b>s_fName</b>	students full name	varchar(255)	Yes	Yes
	<b>s_gender</b>	students gender	varchar(10)	Yes	Yes
	<b>s_institution</b>	students institution of study	varchar(10)	Yes	Yes
	<b>s_address</b>	students address	varchar(255)	Yes	Yes
<b>reward</b>	<b>reward_id</b>	Rewards details correspond to unique serial number	int(11)	No	No
	<b>badge_name</b>	Name of the badges that provided in the system	varchar(255)	No	Yes
	<b>badge_description</b>	Description of the badges	text	No	No
	<b>badge_icon_path</b>	Image path	varchar(255)	No	No
	<b>points_required</b>	The number of activities required to acquire different type of badges	int(11)	No	No

feedbacks	<b>feedback_id</b>	Unique serial numbers have been assigned to each feedback link uploaded or published for an activity.	int(11)	No	No
	<b>activity_id</b>	Feedback details correspond to unique serial number for each activity or event	int(11)	No	No
	<b>link_form</b>	Link of the feedback form given after the activity has been organized	varchar(150)	No	No
	<b>publish_date</b>	Timestamp of publication of feedback link for a specific activity	datetime	No	Yes
per_activity	<b>pac_id</b>	Unique serial numbers personal activity.	int(11)	No	No
	<b>s_id</b>	Student id	int(11)	Yes	No
	<b>name</b>	Personal activity name	varchar(255)	No	Yes
	<b>date</b>	Personal activity date	date	Yes	Yes
	<b>venue</b>	Personal activity venue	varchar(255)	Yes	Yes
	<b>description</b>	Personal activity description	text	Yes	Yes
	<b>evidence</b>	Personal activity evidence	varchar(255)	Yes	Yes
	<b>status</b>	Personal activity validation status	varchar(50)	Yes	Yes
	<b>l_id</b>	Lecturer's id	int(11)	Yes	No

## 4.5 List of Normalized Global DB Relational Schema

1. user (**id**, username, password, email, user\_role, datetime\_register, user\_reg\_status)  
Primary Key: id  
Alternate Key: email
2. masteradministrator (**ma\_id**)  
Primary Key: ma\_id
3. administrator (**a\_id**)  
Primary Key: a\_id
4. student (**s\_id**, s\_email, s\_fName, s\_telephone\_no, s\_address, s\_institution, s\_course, s\_skills, s\_hobby, s\_achievement, s\_ambition, s\_academic\_cert, s\_cocurriculum\_cert, s\_gender, s\_race, s\_age)  
Primary Key: s\_id  
Foreign Key: s\_email
5. lecturer (**l\_id**, l\_email, l\_fName, l\_telephone\_no, l\_address, l\_institution, l\_gender, l\_age, l\_race)  
Primary Key: l\_id  
Foreign Key: l\_email
6. activity (**activity\_id**, user\_id, title, activity\_desc, act\_datetime, category, location, organizer\_name, skill\_acquired, attachment, link\_form, review, validation\_status)  
Primary Key: activity\_id
7. activity\_participant (**participant\_id**, activity\_id, s\_id, s\_fName, s\_email, s\_gender, s\_institution, s\_address)  
Primary Key: participant\_id

8. per\_activity (pac\_id, s\_id, name, date, venue, description, status, evidence, l\_id)

Primary Key: pac\_id

9. feedbacks (feedback\_id, activity\_id, link\_form, publish\_date)

Primary Key: feedback\_id

Foreign Key: activity\_id

10. profile (p\_id, p\_email, p\_name, gender, race, age, dob, profileimage, position, headline, about, country, citystate)

Primary Key: p\_id

Foreign Key: p\_email

11. reward (reward\_id, badge\_name , badge\_description , badge\_icon\_path , points\_required)

Primary Key: reward\_id

Foreign Key : points\_required

# CHAPTER 5

## SQL Statements (DDL & DML)

### Module 1: Authentication

The image shows a registration form titled "YOUTH Registration Here !". The form is divided into two columns. The left column contains fields for Username, Email, Full Name, User Role (set to "Student"), Gender (set to "Male"), Age, Password, Confirm Password, Address, Course, Institution, Telephone, Race, and E-mail. The right column contains fields for Email, Full Name, User Role (set to "Student"), Gender (set to "Male"), Age, Password, Confirm Password, Address, Course, Institution, Telephone, Race, and E-mail. At the bottom of the form is a "Register" button.

**Figure 5.1.1 Register Student account**

```
$this->db->query('INSERT INTO student (s_email, s_fName, s_telephone_no, s_address, s_institution, s_course, s_skills, s_hobby, s_achievement, s_ambition, s_academic_cert, s_cocurriculum_cert, s_age, s_gender, s_race) VALUES(:s_email, :s_fName, :s_telephone_no, :s_address, :s_institution, :s_course, :s_skills, :s_hobby, :s_achievement, :s_ambition, :s_academic_cert, :s_cocurriculum_cert, :s_age , :s_gender, :s_race)');
```

The registration form for a Lecturer account on the YOUTH platform is displayed on a yellow background. The form fields include:

- Username: yeudi@gmail.com
- Email: yeudi@gmail.com
- Full Name: Full name \*
- User Role: Student
- Gender: Male
- Age: Age \*
- Password: Password \*
- Confirm Password: Confirm Password \*
- Address: Address \*
- Course: Course \*
- Institution: Institution \*
- Telephone: Telephone \*
- Race: Race \*

At the bottom of the form is a "Register" button.

**Figure 5.1.2 Register Lecturer account**

```
$this->db->query('INSERT INTO lecturer (l_email, l_fName, l_telephone_no, l_address,
l_institution, l_age, l_gender, l_race) VALUES(:l_email, :l_fName, :l_telephone_no,
:l_address, :l_institution, :l_age, :l_gender, :l_race)');
```

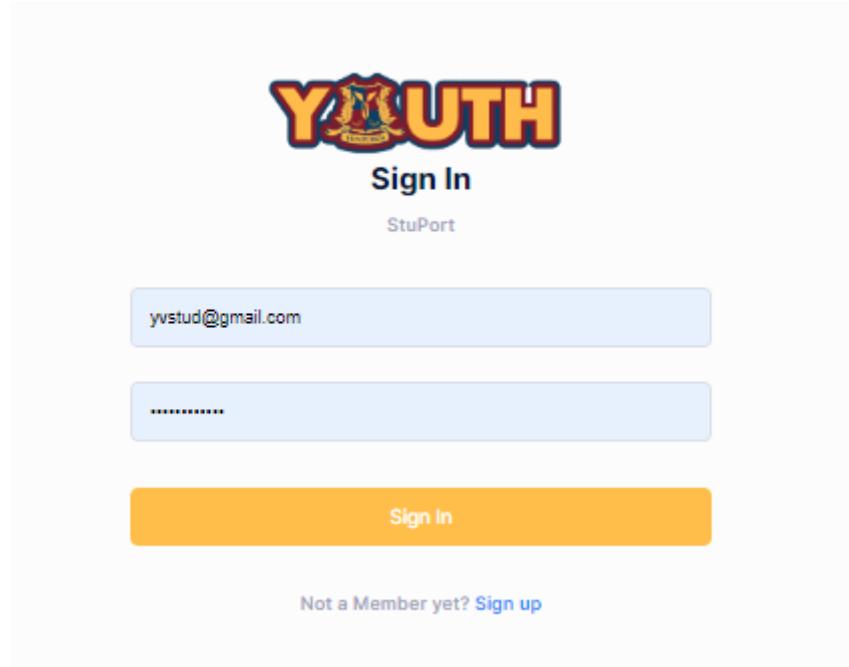
The registration form for all user accounts on the YOUTH platform is displayed on a yellow background. The form fields are identical to Figure 5.1.2, including:

- Username: yeudi@gmail.com
- Email: yeudi@gmail.com
- Full Name: Full name \*
- User Role: Student
- Gender: Male
- Age: Age \*
- Password: Password \*
- Confirm Password: Confirm Password \*
- Address: Address \*
- Course: Course \*
- Institution: Institution \*
- Telephone: Telephone \*
- Race: Race \*

At the bottom of the form is a "Register" button.

**Figure 5.1.3 Register all user account**

```
$this->db->query('INSERT INTO user (username, password, email, user_role,
datetime_register, user_reg_status) VALUES(:username, :password, :email, :user_role,
:datetime_register, :user_reg_status)');
```



**Figure 5.1.4 Log in all user account**

```
$this->db->query('SELECT * FROM user WHERE email = :email');
```

## Module 2: Profile

Profile Details	
Name	Muhammad Ali
Email	yvlect@gmail.com
Gender	Male
Race	Indian
Age	36

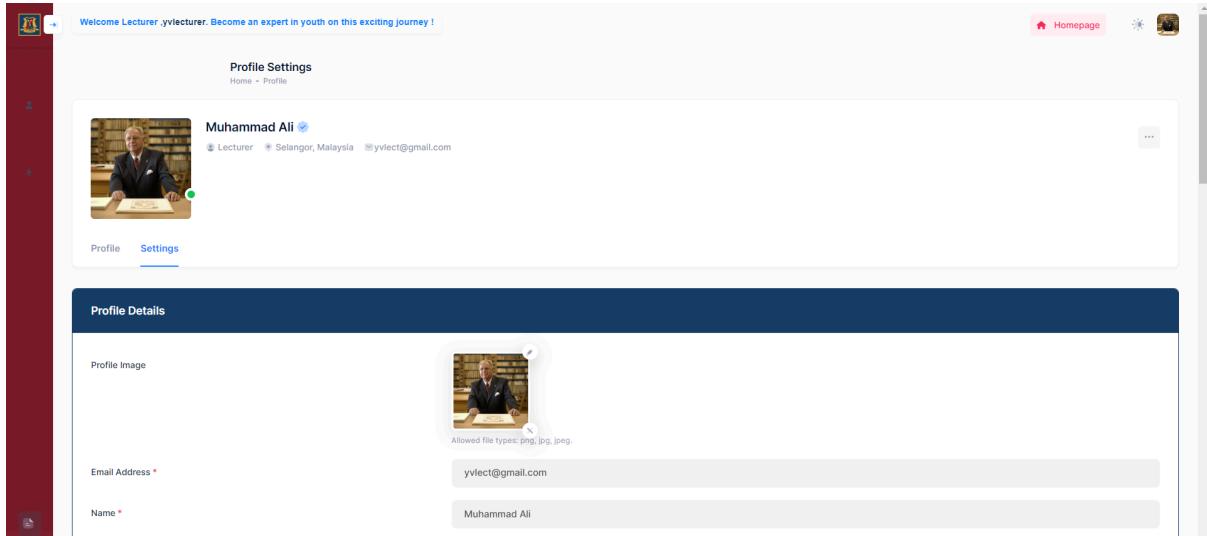
**Figure 5.2.1 Viewing Lecturer Profile**

```
$this->db->query("SELECT * FROM profile AS p JOIN user AS u ON p_email = u.email  
JOIN lecturer AS l ON p_email = l_email WHERE u.email = :email");
```

The screenshot shows the YOUTH platform's profile overview page. On the left is a dark red sidebar with navigation links: PROFILE (Profile Settings), PERSONAL ACTIVITY (Personal Activity Options), ACTIVITY (Activity Options), and REWARD (Reward Options). The main content area has a white header with the text "Welcome Student ,yvstudent. Become an expert in youth on this exciting journey !", a "Homepage" button, and a user profile icon. Below the header is a "Profile Overview" section with a sub-header "Home - Profile". It features a profile picture of a woman named Lee Yik Hong, her status as a "Student" from "Johor, Malaysia" with email "yvstud@gmail.com", and a "..." button. Below this is a tab bar with "Profile" (selected) and "Settings". The main content area has a yellow header "Profile Details" with an "Edit Profile" button. The details listed are Name: Lee Yik Hong, Email: yvstud@gmail.com, Gender: Male, Race: Chinese, and Age: 23.

**Figure 5.2.2 Viewing Student Profile**

```
$this->db->query("SELECT * FROM profile AS p JOIN user AS u ON p_email = u.email  
JOIN student AS s ON p_email = s_email WHERE u.email = :email");
```



**Figure 5.2.3 Editing Lecturer Profile**

### **With Profile Image**

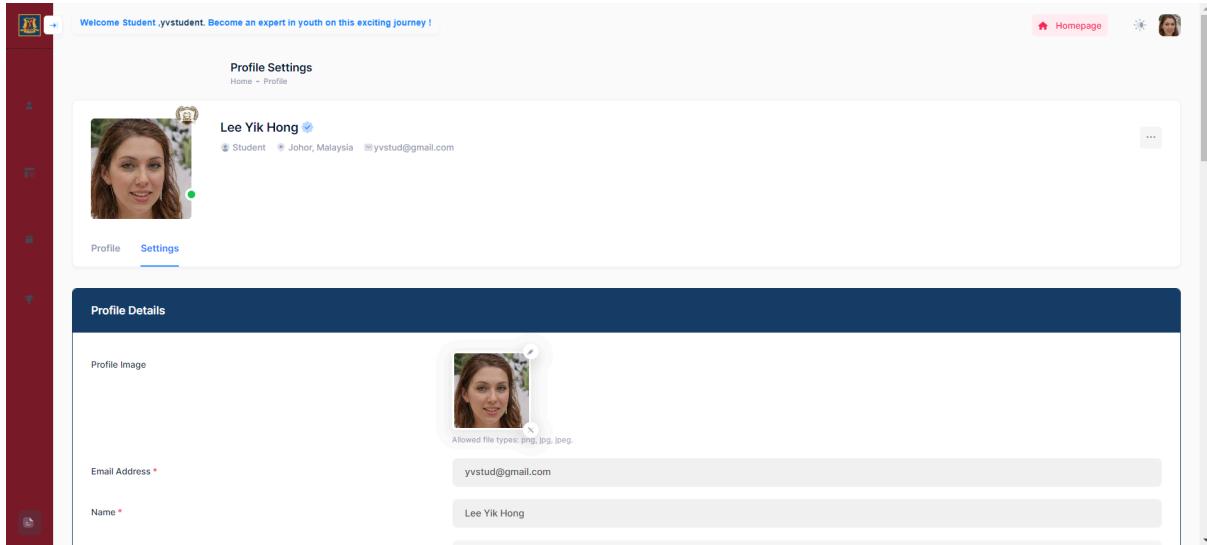
```
$this->db->query("UPDATE profile SET p_email = :email, p_name = :p_name, dob = :dob,
profileimage = :profileimage, position = :position, headline = :headline, about = :about,
country = :country, citystate = :citystate WHERE p_email = :email;
```

```
UPDATE lecturer SET l_email = :email, l_fName = :l_fName, l_telephone_no =
:l_telephone_no, l_address = :l_address, l_institution = :l_institution, l_gender = :l_gender,
l_age = :l_age, l_race = :l_race WHERE l_email = :email");
```

### **Without Profile Image**

```
this->db->query("UPDATE profile SET p_email = :email, p_name = :p_name, dob = :dob,
position = :position, headline = :headline, about = :about, country = :country, citystate =
:citystate WHERE p_email = :email;
```

```
UPDATE lecturer SET l_email = :email, l_fName = :l_fName, l_telephone_no =
:l_telephone_no, l_address = :l_address, l_institution = :l_institution, l_gender = :l_gender,
l_age = :l_age, l_race = :l_race WHERE l_email = :email");
```



**Figure 5.2.4 Editing Student Profile**

### **With Profile Image**

```
$this->db->query("UPDATE profile SET p_email = :email, p_name = :p_name, dob = :dob,
profileimage = :profileimage, position = :position, headline = :headline, about = :about,
country = :country, citystate = :citystate WHERE p_email = :email;
```

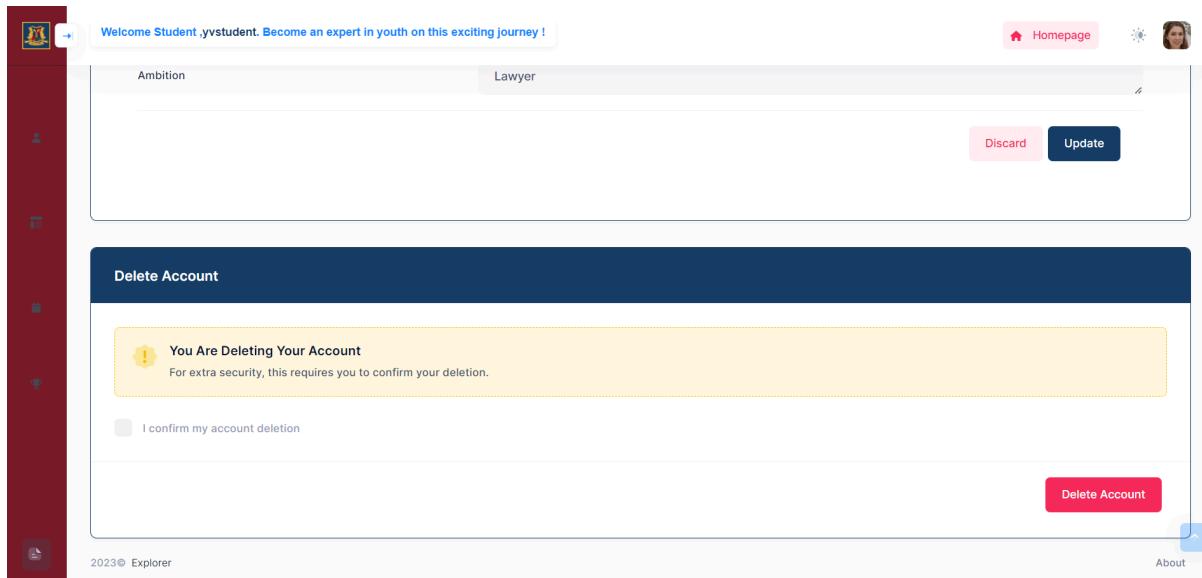
```
UPDATE student SET s_email = :email, s_fName = :s_fName, s_telephone_no =
:s_telephone_no, s_address = :s_address, s_institution = :s_institution, s_course = :s_course,
s_skills = :s_skills, s_hobby = :s_hobby, s_achievement = :s_achievement, s_ambition =
:s_ambition, s_academic_cert = :s_academic_cert, s_cocurriculum_cert =
:s_cocurriculum_cert, s_race = :s_race, s_age = :s_age, s_gender = :s_gender WHERE
s_email = :email;");
```

### **Without Profile Image**

```
$this->db->query("UPDATE profile SET p_email = :email, p_name = :p_name, dob = :dob,
position = :position, headline = :headline, about = :about, country = :country, citystate =
:citystate WHERE p_email = :email;
```

```
UPDATE student SET s_email = :email, s_fName = :s_fName, s_telephone_no =
:s_telephone_no, s_address = :s_address, s_institution = :s_institution, s_course = :s_course,
s_skills = :s_skills, s_hobby = :s_hobby, s_achievement = :s_achievement, s_ambition =
```

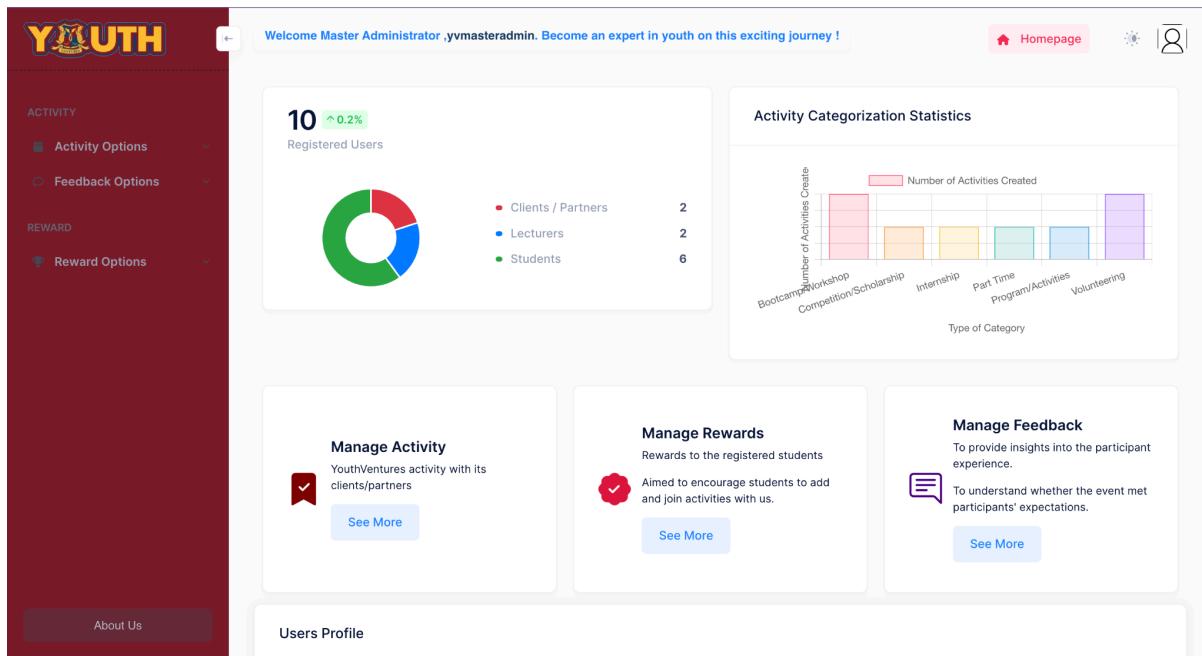
```
:s_ambition, s_academic_cert = :s_academic_cert, s_cocurriculum_cert =
:s_cocurriculum_cert, s_race = :s_race, s_age = :s_age, s_gender = :s_gender WHERE
s_email = :email;");
```



**Figure 5.2.5 Deleting Student and Lecturer Profile**

```
$this->db->query("DELETE FROM profile WHERE p_email = :email; DELETE FROM
user WHERE email = :email; DELETE FROM student WHERE s_email = :email; DELETE
FROM lecturer WHERE l_email = :email;");
```

## Module 3: Dashboard



**Figure 5.3.1 Viewing Master Administrator Dashboard**

```
$userSql = "SELECT COUNT(*) as userCount FROM user";  
  

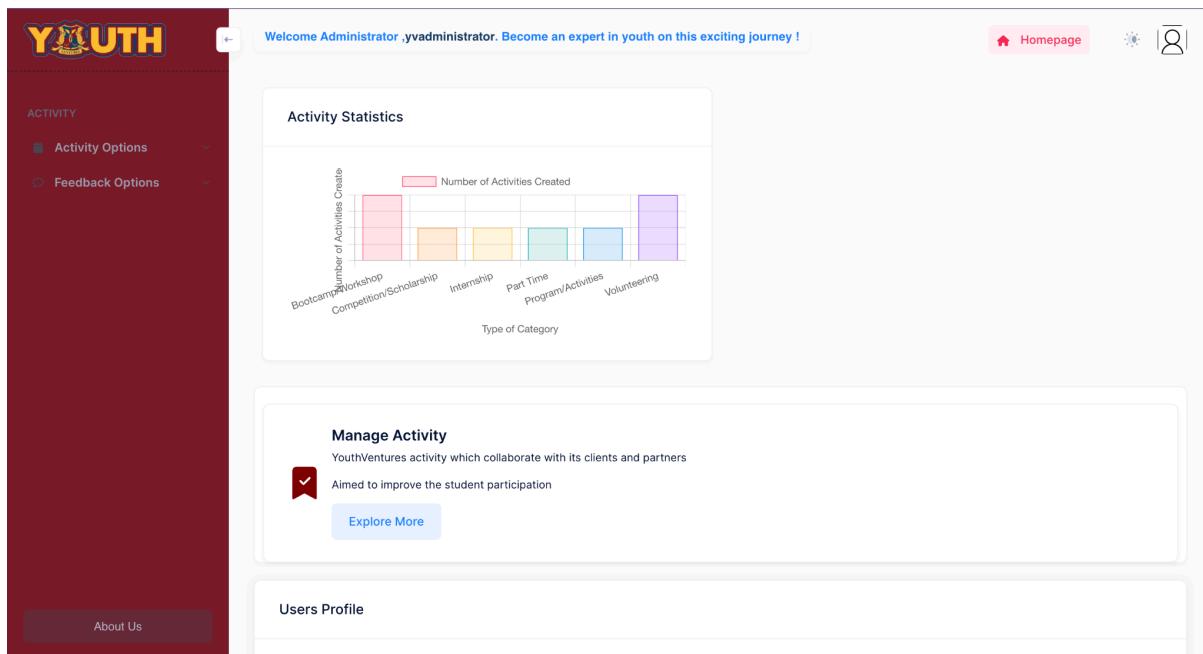
$lecturerSql = "SELECT COUNT(*) as lecturerCount FROM lecturer";  
  

$studentSql = "SELECT COUNT(*) as studentCount FROM student";  
  

$clientSql = "SELECT COUNT(*) as clientCount FROM administrator";  
  

$stmt = $conn->prepare('SELECT category, COUNT(*) as num_activities FROM activity GROUP BY category');  
  

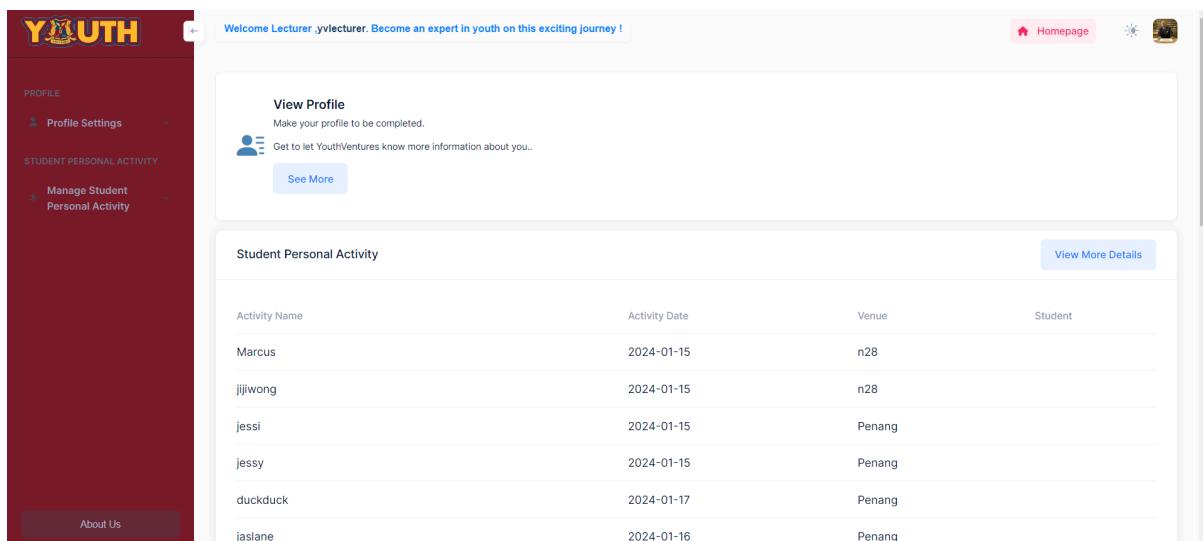
$queryCombined = "SELECT * FROM profile AS p LEFT JOIN user AS u ON p.p_email = u.email LEFT JOIN student AS s ON p.p_email = s.s_email LEFT JOIN lecturer AS l ON p.p_email = l.l_email;";
```



**Figure 5.3.2 Viewing Master Administrator Dashboard**

```
$stmt = $conn->prepare('SELECT category, COUNT(*) as num_activities FROM activity GROUP BY category');
```

```
$queryCombined = "SELECT * FROM profile AS p LEFT JOIN user AS u ON p.p_email = u.email LEFT JOIN student AS s ON p.p_email = s.s_email LEFT JOIN lecturer AS l ON p.p_email = l.l_email;";
```



**Figure 5.3.3 Viewing Lecturer Dashboard**

```
$query = "SELECT name, date, venue FROM per_activity";
```

```
$queryCombined = "SELECT * FROM profile AS p LEFT JOIN user AS u ON p.p_email = u.email LEFT JOIN student AS s ON p.p_email = s.s_email LEFT JOIN lecturer AS l ON p.p_email = l.l_email;";
```

Activity Name	Activity Date	Venue
Marcus	2024-01-15	n28
jijiwong	2024-01-15	n28
jessi	2024-01-15	Penang
jessy	2024-01-15	Penang
duckduck	2024-01-17	Penang

**Figure 5.3.4 Viewing Student Dashboard**

```
$query = "SELECT name, date, venue FROM per_activity";
```

```
$queryCombined = "SELECT * FROM profile AS p LEFT JOIN user AS u ON p.p_email = u.email LEFT JOIN student AS s ON p.p_email = s.s_email LEFT JOIN lecturer AS l ON p.p_email = l.l_email;";
```

## Module 4: Activity

Manage Activities										<a href="#">Create</a>
Activity Code No.	Title	Category	Content	Date	Location	Organizer Name	Skill Acquired	Attachment	Action	Feedback given
21	Kid Helper	Volunteering	Give some funs for the young children.	January 6 01:01	Klang	Child Garden	Critical thinking		<a href="#">Update</a> <a href="#">Delete</a>	<a href="https://airtable.com/appoZ0TyhtweTd3og/shrNNblQFMBl835Rb">https://airtable.com/appoZ0TyhtweTd3og/shrNNblQFMBl835Rb</a>
22	Community Cleanup	Volunteering	Participate in local environmental initiatives by joining community cleanups to keep public spaces tidy.	January 10 04:01	Taman Angkasa 5, Kluang, Johor.	Jawatankuasa Taman Angkasa	Cleaning		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
23	Art and Creativity	Bootcamp/Workshop	Including painting, sculpting, or other creative endeavors.	February 10 08:02	Sasala Hall	Arting Sdn. Bhd	Drawing, Design		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
24	Math Tutor	Part Time	Must have STPM and SPM certificates	January 27 02:01	STDC	Lim Hoon King	Teaching		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
25	Journalism Internship	Internship	Write articles and conduct interviews	January 19 12:01	NONE	WordPublish	Writing		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
26	Cultural Exchange Program	Program/Activities	Experience different cultures, traditions, and lifestyles.	January 31 10:01	SMK AJAK BAHRU	Terengganu	Learn		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
27	Football Competition	Competition/Scholarship	YV Football Tournament	January 17 04:01	Skudai, Johor Bahru	Youth Venture	Football		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>

**Figure 5.4.1 Admin & master admin manage activities**

```
$this->db->query('SELECT * FROM activity');
```

**Create Activity**

[Manage Activities](#)

**Activity Title \***

**Description**

**Category \***

**Date and Time \***

**Location**

**Organizer Name**

**Skill Acquired**

**Attachment**  
  
 Allowed file types: png, jpg, jpeg.

**Submit**

**Figure 5.4.2 Admin & master admin create activities**

```
$this->db->query('INSERT INTO activity (user_id, title, activity_desc, act_datetime,
category, location, organizer_name, skill_acquired, attachment) VALUES (:user_id, :title,
:activity_desc, :act_datetime, :category, :location, :organizer_name, :skill_acquired,
:attachment)');
```

Welcome Administrator .yvadministor. Become an expert in youth on this exciting journey !

[Homepage](#) [Logout](#)

**Manage Activities** [Create](#)

Activity No. ^	Title	Category	Content	Date	Location	Organizer Name	Skill Acquired	Attachment	Action	Feedback given
21	Kid Helper	Volunteering	Give some funs for the young children.	January 6 01:01	Klang	Child Garden	Critical thinking		<a href="#">Update</a> <a href="#">Delete</a>	<a href="https://airtable.com/appZ0TyhtweTd3og/shrNNbLOFMBn83SRb">https://airtable.com/appZ0TyhtweTd3og/shrNNbLOFMBn83SRb</a>
22	Community Cleanup	Volunteering	Participate in local environmental initiatives by joining community cleanups to keep public spaces tidy.	January 10 04:01	Taman Angkasa 5, Kluang, Johor.	Jewatankusa Taman Angkasa	Cleaning		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
23	Art and Creativity	Bootcamp/Workshop	Including painting, sculpting, or other creative endeavors.	February 10 08:02	Sasala Hall	Arting Sdn. Bhd	Drawing, Design		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
24	Math Tutor	Part Time	Must have STPM and SPM certificates	January 27 02:01	STDC	Lim Hoon King	Teaching		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
25	Journalism Internship	Internship	Write articles and conduct interviews	January 19 12:01	NONE	WordPublish	Writing		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
26	Cultural Exchange Program	Program/Activities	Experience different cultures, traditions, and lifestyles.	January 31 10:01	SMK AJAK BAHRU	Terengganu	Learn		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>
27	Football Competition	Competition/Scholarship	YV Football Tournament	January 17 04:01	Skudai, Johor Bahru	Youth Venture	Football		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>

**Figure 5.4.3 Admin & master admin find activity by Id**

```
$this->db->query('SELECT * FROM activity WHERE activity_id = :activity_id');
```

**Update Activity**

[Manage Activities](#)

---

**Activity Title \***  
Football Competition

**Description**  
\* YV Football Tournament

**Date and Time \***  
17/01/2024 04:33 AM

**Location**  
Skudai, Johor Bahru

**Organizer Name**  
Youth Venture

**Skill Acquired**  
Football

**Attachment**


  
Allowed file types: png, jpg, jpeg.

**Submit**

---

**Footer**

**Figure 5.4.4 Admin & master admin update activity**

```
$this->db->query('UPDATE activity SET title = :title, activity_desc = :activity_desc,
location = :location, organizer_name = :organizer_name, skill_acquired = :skill_acquired
WHERE activity_id = :activity_id; UPDATE feedbacks SET title = :title WHERE
activity_id = :activity_id');
```

Manage Activities											<a href="#">Create</a>
Activity Code No.	Title	Category	Content	Date	Location	Organizer Name	Skill Acquired	Attachment	Action	Feedback given	
21	Kid Helper	Volunteering	Give some funs for the young children.	January 6 01:01	Klang	Child Garden	Critical thinking		<a href="#">Update</a> <a href="#">Delete</a>	<a href="https://airtable.com/appZ0TyhtweTd3oq/shrNNbLOfMBn835Rb">https://airtable.com/appZ0TyhtweTd3oq/shrNNbLOfMBn835Rb</a>	
23	Art and Creativity	Bootcamp/Workshop	Including painting, sculpting, or other creative endeavors.	February 10 08:02	Sasala Hall	Arting Sdn. Bhd	Drawing, Design		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>	
24	Math Tutor	Part Time	Must have STPM and SPM certificates	January 27 02:01	STDC	Lim Hoon King	Teaching		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>	
25	Journalism Internship	Internship	Write articles and conduct interviews	January 19 12:01	NONE	WordPublish	Writing		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>	
26	Cultural Exchange Program	Program/Activities	Experience different cultures, traditions, and lifestyles.	January 31 10:01	SMK AJAK BAHRU	Terengganu	Learn		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>	
27	Football Competition	Competition/Scholarship	YY Football Tournament	January 17 04:01	Skudai, Johor Bahru	Youth Venture	Football		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>	
28	Handcraft Workshop	Bootcamp/Workshop	Handcraft skill	January 15 08:01	PSZ	Youth Venture Asia	Handcrafting		<a href="#">Update</a> <a href="#">Delete</a>	<a href="#">Add Feedback</a>	

10 ▾ Showing 1 to 7 of 7 records

< 1 >

**Figure 5.4.5 Admin & master admin delete activity**

```
$this->db->query('DELETE FROM activity WHERE activity_id = :activity_id');
```

## Module 5: Registration

Activity									
Title ^	Category	Content	Date	Location	Organizer Name	Skill Acquired	Attachment	Action	Feedback need to filled after event
Art and Creativity	Bootcamp/Workshop	Including painting, sculpting, or other creative endeavors.	February 10 08:02	Sasala Hall	Artling Sdn. Bhd	Drawing, Design		<button>Register</button>	Feedback is not given yet.
Cultural Exchange Program	Program/Activities	Experience different cultures, traditions, and lifestyles.	January 31 10:01	SMK AJAK BAHRU	Terengganu	Learn		<button>Register</button>	Feedback is not given yet.
Football Competition	Competition/Scholarship	YV Football Tournament	January 17 04:01	Skudai, Johor Bahru	Youth Venture	Football		<button>Register</button>	Feedback is not given yet.
Handcraft Workshop	Bootcamp/Workshop	Handcraft skill	January 15 08:01	PSZ	Youth Venture Asia	Handcrafting		<button>Register</button>	Feedback is not given yet.
Journalism Internship	Internship	Write articles and conduct interviews	January 19 12:01	NONE	WordPublish	Writing		<button>Register</button>	Feedback is not given yet.
Kid Helper	Volunteering	Give some funs for the young children.	January 6 01:01	Klang	Child Garden	Critical thinking		<button>Register</button>	<a href="https://airtable.com/appZ0TyhtweTd3oq/shNNbLOfMBn83SRb">https://airtable.com/appZ0TyhtweTd3oq/shNNbLOfMBn83SRb</a>
Math Tutor	Part Time	Must have STPM and SPM certificates	January 27 02:01	STDC	Lim Hoon King	Teaching		<button>Register</button>	Feedback is not given yet.

Showing 1 to 7 of 7 records

**Figure 5.5.1 Student register activity**

```
$this->db->query('SELECT * FROM user WHERE id = :user_id');
```

```
$this->db->query('SELECT * FROM student WHERE s_email = :email');
```

```
$this->db->query('SELECT COALESCE(MAX(participant_id), 0) AS max_participant_id
```

```
$this->db->query('INSERT INTO activity_participant ( activity_id, s_id, s_fName, s_email, s_gender, s_institution, s_address) VALUES ( :activity_id, :s_id, :s_fName, :s_email, :s_gender, :s_institution, :s_address)');
```

Activity Registered						
No. ^	Category	Activity's Name	Activity Date	Venue	Description	Action
1	Bootcamp/Workshop	Art and Creativity	February 10 08:02	Sasala Hall	Including painting, sculpting, or other creative endeavors.	<button>Registered</button>
2	Part Time	Math Tutor	January 27 02:01	STDC	Must have STPM and SPM certificates	<button>Registered</button>
3	Program/Activities	Cultural Exchange Program	January 31 10:01	SMK AJAK BAHRU	Experience different cultures, traditions, and lifestyles.	<button>Registered</button>
4	Competition/Scholarship	Football Competition	January 17 04:01	Skudai, Johor Bahru	YV Football Tournament	<button>Registered</button>
5	Internship	Journalism Internship	January 19 12:01	NONE	Write articles and conduct interviews	<button>Registered</button>
6	Volunteering	Kid Helper	January 6 01:01	Klang	Give some funs for the young children.	<button>Registered</button>
7	Bootcamp/Workshop	Handcraft Workshop	January 15 08:01	PSZ	Handcraft skill	<button>Registered</button>

Showing 1 to 7 of 7 records

**Figure 5.5.2 Student registered activity**

```

$this->db->query('SELECT * FROM user WHERE id = :user_id');

$this->db->query('SELECT * FROM student WHERE s_email = :email');

$this->db->query('SELECT * FROM activity_participant WHERE s_id = :s_id');

$this->db->query('SELECT * FROM activity WHERE activity_id = :activity_id');

```

Activity									
Title ^	Category	Content	Date	Location	Organizer Name	Skill Acquired	Attachment	Action	Feedback need to filled after event
Art and Creativity	Bootcamp/Workshop	Including painting, sculpting, or other creative endeavors.	February 10 08:02	Sasala Hall	Artling Sdn. Bhd	Drawing, Design		<button>Register</button>	Feedback is not given yet.
Cultural Exchange Program	Program/Activities	Experience different cultures, traditions, and lifestyles.	January 31 10:01	SMK AJAK BAHRU	Terengganu	Learn		<button>Register</button>	Feedback is not given yet.
Football Competition	Competition/Scholarship	YV Football Tournament	January 17 04:01	Skudai, Johor Bahru	Youth Venture	Football		<button>Register</button>	Feedback is not given yet.
Handcraft Workshop	Bootcamp/Workshop	Handcraft skill	January 15 08:01	PSZ	Youth Venture Asia	Handcrafting		<button>Register</button>	Feedback is not given yet.
Journalism Internship	Internship	Write articles and conduct interviews	January 19 12:01	NONE	WordPublish	Writing		<button>Register</button>	Feedback is not given yet.
Kid Helper	Volunteering	Give some funs for the young children.	January 6 01:01	Klang	Child Garden	Critical thinking		<button>Register</button>	<a href="https://airtable.com/app20TyhtweTd3oq/shNNbLOFBn83SRb">https://airtable.com/app20TyhtweTd3oq/shNNbLOFBn83SRb</a>
Math Tutor	Part Time	Must have STPM and SPM certificates	January 27 02:01	STDC	Lim Hoon King	Teaching		<button>Register</button>	Feedback is not given yet.

Showing 1 to 7 of 7 records

**Figure 5.5.2 Check if student registered activity**

```

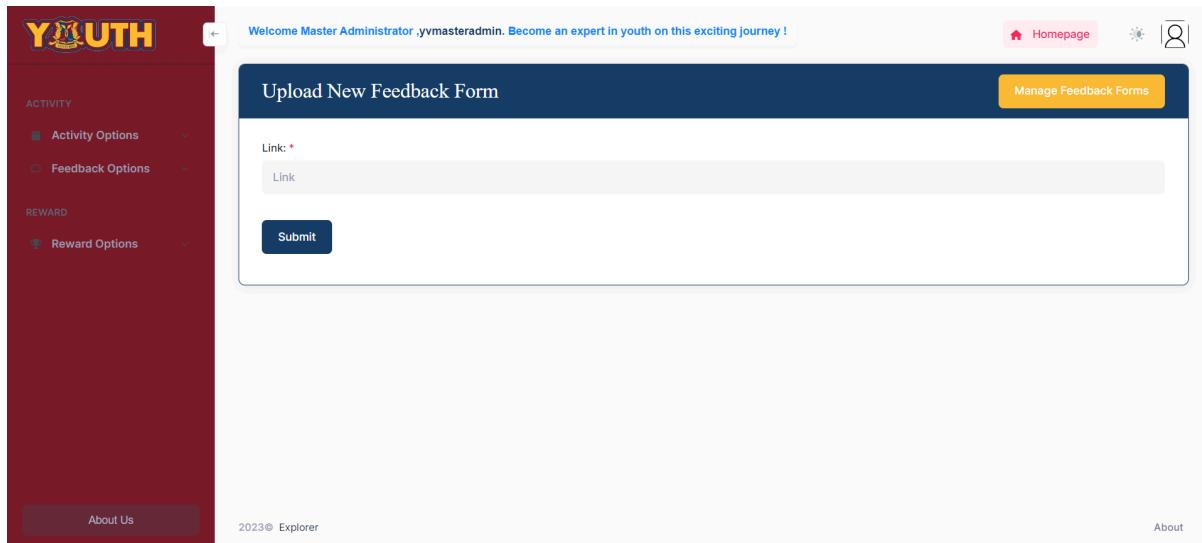
$this->db->query('SELECT * FROM user WHERE id = :user_id');

$this->db->query('SELECT * FROM student WHERE s_email = :email');

$this->db->query('SELECT * FROM activity_participant WHERE s_id = :s_id AND
activity_id = :activity_id');

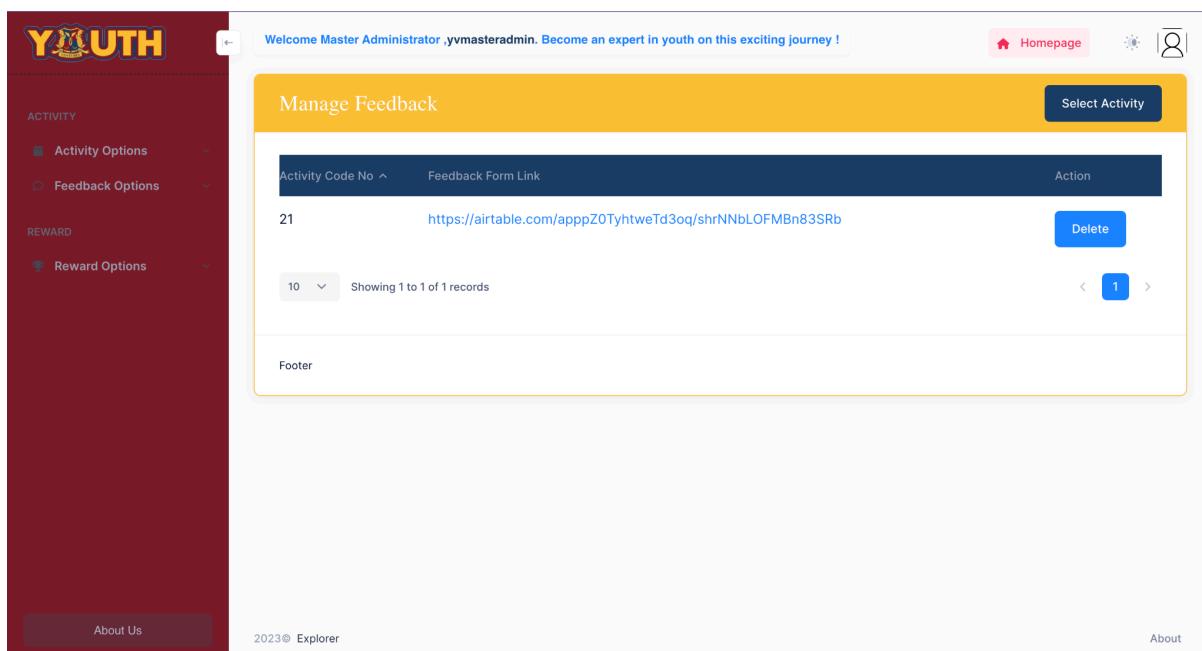
```

## Module 6: Feedback



**Figure 5.6.1 Master Admin upload feedback**

```
$this->db->query('INSERT INTO feedbacks (link_form, activity_id) VALUES (:link_form, :activity_id); UPDATE activity SET link_form = :link_form WHERE activity_id = :activity_id');
```

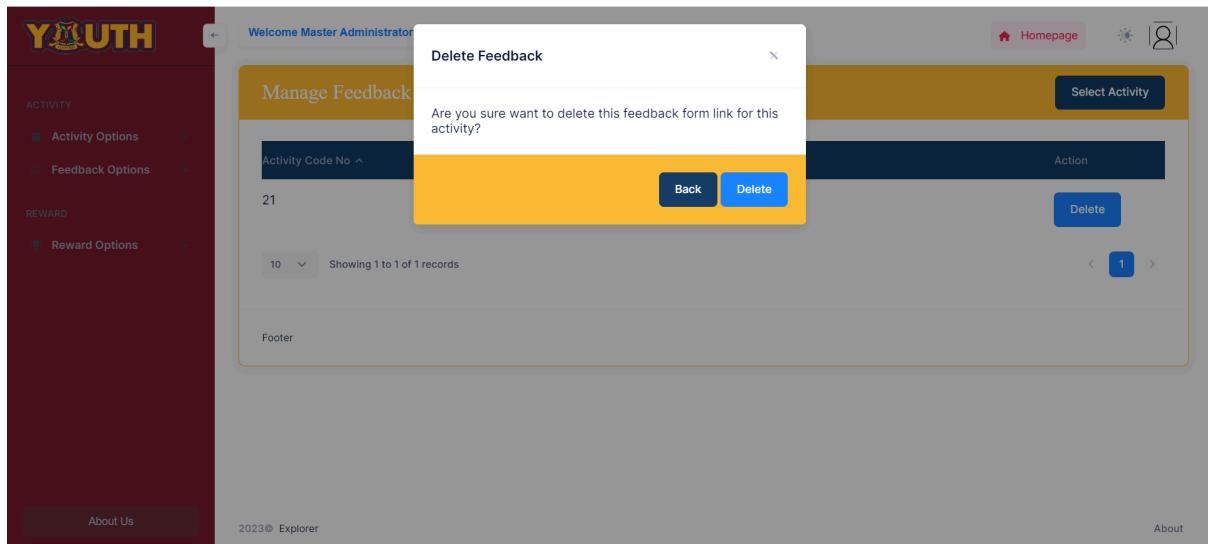


**Figure 5.6.2 Master Admin and Admin manage feedback**

```
$this->db->query('SELECT * FROM feedbacks');
```

```
$this->db->query('SELECT * FROM feedbacks WHERE feedback_id = :feedback_id');
```

```
$this->db->query('SELECT * FROM activity WHERE activity_id = :activity_id');
```



**Figure 5.6.3 Master Admin and Admin delete feedback**

```
$this->db->query('DELETE FROM feedbacks WHERE feedback_id = :feedback_id');
UPDATE activity SET link_form = :link_form WHERE activity_id = :activity_id');
```

## Module 7: Reward

Welcome Master Administrator .yvmasteradmin. Become an expert in youth on this exciting journey !

[Homepage](#)

[Create Rewards](#)

Badge Icon	Badge Name	Badge Description	Activities Joined	Action
	Bronze	The activities joined by students is less than 10	10	<a href="#">Update</a> <a href="#">Delete</a>
	gold	The activities joined by students is less than 30	30	<a href="#">Update</a> <a href="#">Delete</a>
	Diamond	The activities joined by students is less than 50	50	<a href="#">Update</a> <a href="#">Delete</a>

**Figure 5.7.1 Master Admin and manage rewards**

```
$this->db->query('SELECT * FROM reward');
```

Welcome Master Administrator .yvmasteradmin. Become an expert in youth on this exciting journey !

[Homepage](#)

[Manage Rewards](#)

Badge Name *	gold
Badge Description	above 50
Badge icon	<a href="#">Remove</a>
Activity Joined *	50

Allowed file types: png, jpg, jpeg.

[Submit](#)

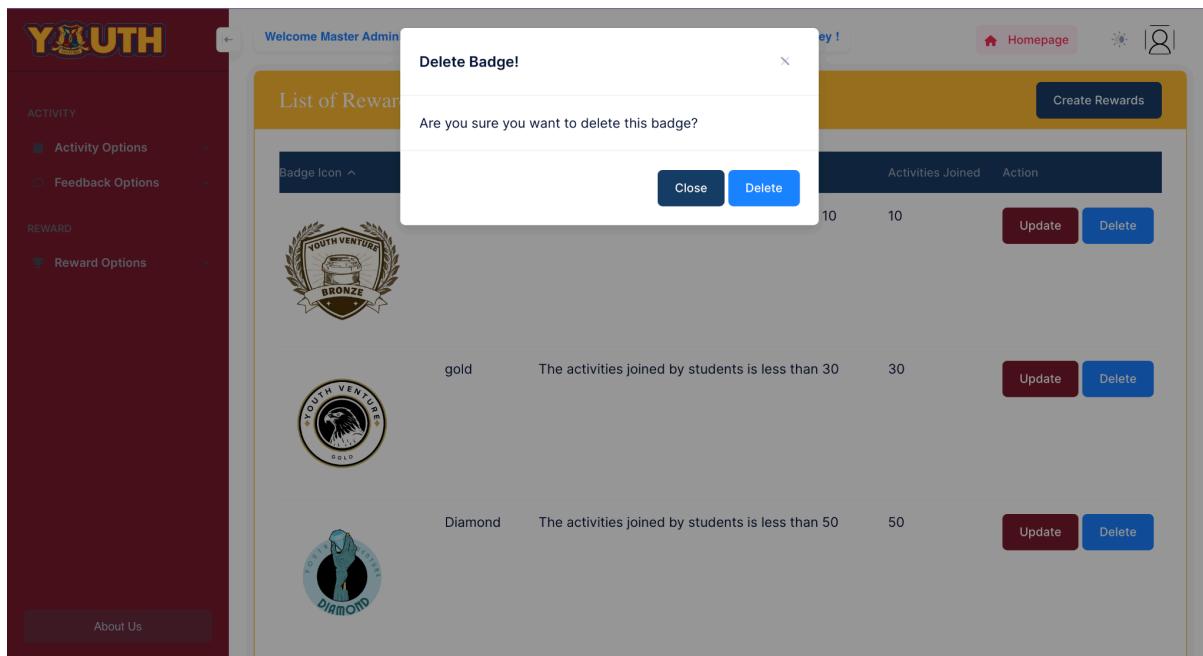
**Figure 5.7.2 Master Admin and create rewards**

```
$this->db->query('INSERT INTO reward (badge_name, badge_description,
badge_icon_path, points_required) VALUES (:badge_name, :badge_description,
:badge_icon_path, :points_required)');
```

The screenshot shows a web-based application interface for managing rewards. On the left, there's a dark sidebar with a logo and navigation links for 'ACTIVITY' (Activity Options, Feedback Options) and 'REWARD' (Reward Options). The main content area has a header 'Welcome Master Administrator, yvmasteradmin. Become an expert in youth on this exciting journey !' with links for 'Homepage', 'Logout', and a user icon. Below the header is a 'Update Reward' form. The form fields are: 'Badge Name' (input: Bronze), 'Badge Description' (input: The activities joined by students is less than 10), 'New Icon (if any)' (input: Choose file - No file chosen), and 'Activities Joined' (input: 10). At the bottom of the form is a 'Submit' button.

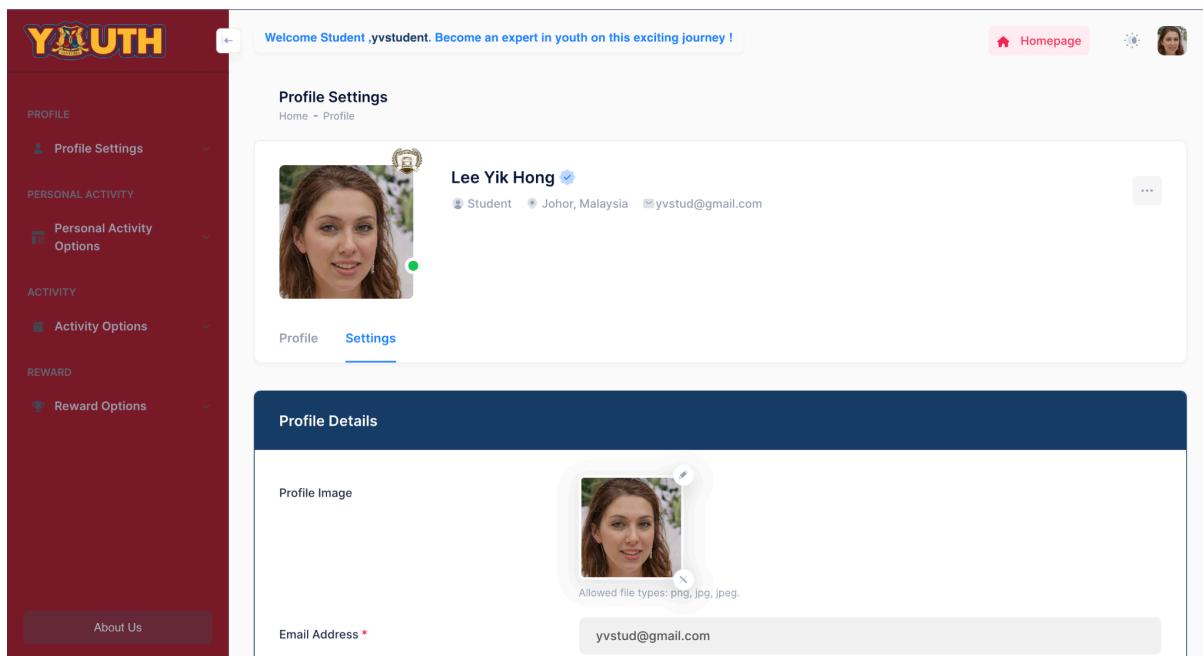
**Figure 5.7.3 Master Admin and update rewards**

```
$this->db->query('SELECT * FROM reward WHERE reward_id = :reward_id');
$this->db->query('UPDATE reward SET badge_name = :badge_name, badge_description =
:badge_description, badge_icon_path = :badge_icon_path, points_required = :points_required
WHERE reward_id = :reward_id');
```



**Figure 5.7.4 Master Admin and delete rewards**

```
$this->db->query('DELETE FROM reward WHERE reward_id = :id');
```



**Figure 5.7.5 Students and view rewards**

```
$this->db->query('SELECT * FROM reward WHERE badge_name = :badge_name');
```

## Module 8: Resume

**Resume**



**Lee Yik Hong**  
**Student**  
Work hard.  
[yystud@gmail.com](mailto:yystud@gmail.com)  
[012345678910](tel:012345678910)  
Johor, Malaysia

**About Me**  
Student in UTM

**Personal Details**  
Date of Birth: 0000-00-00  
Age : 23  
Gender: Male  
Race : Chinese  
Address : 104,Jalan Jasa 88, Taman Mutiara Rini  
Institution : UTM  
Course : Data Engineering  
Hobby : Gardening  
Ambition : Lawyer

**Achievement**  
Best Poster in SayNoToPlastic week.

**Skills & Proficiency**  
Coding C++

**Print**

This resume is auto-generated by Youth Ventures Asia. Designed by Explorer for StuPort

**Figure 5.8.1 Student downloads Master Transcript**

```
$this->db->query("SELECT * FROM profile AS p JOIN user AS u ON p_email = u.email  
JOIN student AS s ON p_email = s_email WHERE u.email = :email");
```

## Module 9: Personal Activity

Manage Personal Activity							Create
Personal Activity's Name ~	Date	Venue	Description	Evidence	Lecturer	Action	
dd	February 15 12:02	Penang	competition	<a href="#">View</a>	Muhammad Ali	<a href="#">Update</a>	<a href="#">Assign</a>
Ping Pong Competition	February 7 12:02	Penang	competition	<a href="#">View</a>		<a href="#">Update</a>	<a href="#">Assign</a>
						<a href="#">Delete</a>	<a href="#">Delete</a>

**Figure 5.9.1 Manage student personal activities**

```
$this->db->query('SELECT * FROM per_activity ORDER BY date ASC');
```

```
$this->db->query('SELECT * FROM per_activity WHERE pac_id = :pac_id');
```

### Create Personal Activity

[Manage Personal Activity](#)

Name of Personal Activity \*

Date \*

Venue \*

Description

Evidence \*

Choose File No file chosen

**Submit**

The screenshot shows a web-based application for creating personal activities. At the top, there's a dark blue header bar with the title 'Create Personal Activity' on the left and a yellow 'Manage Personal Activity' button on the right. Below the header, the main form area has a light gray background. It contains five input fields: 'Name of Personal Activity' (with a placeholder 'Name of Activity'), 'Date' (with a placeholder 'dd/mm/yyyy' and a small calendar icon), 'Venue' (with a placeholder 'Venue of Activity'), 'Description' (with a placeholder '\*' and a red asterisk indicating it's required), and 'Evidence' (with a 'Choose File' button and a message 'No file chosen'). At the bottom of the form is a dark blue 'Submit' button.

**Figure 5.9.2 Create student personal activities**

```
$this->db->query('INSERT INTO per_activity (s_id, name, date, venue, description, status, evidence) VALUES (:s_id, :name, :date, :venue, :description, "Waiting", :evidence)');
```

## Update Activity

Name \*

Venue of Activity \*

Date \*

 CALENDAR

Description

Evidence \*

Current Evidence: uploads/Edit Menu Sequence Diagram.png

**Submit**

**Figure 5.9.3 Update student personal activities**

```
$this->db->query('UPDATE per_activity SET name = :name, venue = :venue, date = :date ,
`description` = :description WHERE pac_id = :pac_id');
```

Manage Personal Activity						Create
Personal Activity's Name ^	Date	Venue	Description	Evidence	Lecturer	Action
dd	February 15 12:02	Penang	competition	<a href="#">View</a>	Muhammad Ali	<a href="#">Update</a> <a href="#">Assign</a> <a href="#">Delete</a>
Showing 1 to 1 of 1 records						

**Figure 5.9.3 Delete student personal activities**

```
$this->db->query('DELETE FROM per_activity WHERE pac_id = :pac_id');
```

Approved Personal Activity					
Personal Activity's Name	Date	Venue	Description	Evidence	Action
Badminton Competition	January 15 12:01	Seremban	competition	uploads/Module 4 The Network Layer (Data Plane) (7 Dec 2022)(Student).pdf	<button>Approved</button>
Capybara	January 17 12:01	Penang	competition	uploads/Module 7 Wireless and Mobile Networks v5 (13 Jan 2023)(Student).pdf	<button>Approved</button>
duckduck	January 17 12:01	Penang	competition	Module 4 The Network Layer (Data Plane)(7 Dec 2022)(Student).pdf	<button>Approved</button>
jaslane	January 16 12:01	Penang	competition	uploads/Module 6 The Link Layer v4 (7 Dec 2022)(Student).pdf	<button>Approved</button>
jessi	January 15 12:01	Penang	competition	Module 6 The Link Layer v4 (7 Dec 2022)(Student).pdf	<button>Approved</button>
jessy	January 15 12:01	Penang	competition	Module 4 The Network Layer (Data Plane)(7 Dec 2022)(Student).pdf	<button>Approved</button>
jjiwong	January 15 12:01	n28	competition	Module 2 Application Layer v7 (missing pages).pdf	<button>Approved</button>
Marcus	January 15 12:01	n28	competition	Module 4 The Network Layer (Data Plane)(7 Dec 2022)(Student).pdf	<button>Approved</button>
Ping Pong Competition	January 16 12:01	Seremban	competition	uploads/Module 2 Application Layer v7 (missing pages).pdf	<button>Approved</button>

Showing 1 to 9 of 9 records

**Figure 5.9.4 Approved student personal activities**

```
$this->db->query('SELECT * FROM per_activity WHERE s_id = :s_id AND status = "Approved"');
```

Assign this personal activity to:						
Lecturer:						
Muhammad Ali [1]						
<input checked="" type="checkbox"/> Muhammad Ali [1]						
<input type="checkbox"/> Dr Aryati [2]						
<input type="button" value="Submit"/>						

Manage Personal Activity						
Personal Activity's Name	Date	Venue	Description	Evidence	Lecturer	Action
dd	February 15 12:02	Penang	competition	<a href="#">View</a>	Dr Aryati	<button>Update</button> <button>Assign</button> <button>Delete</button>

Showing 1 to 1 of 1 records

**Figure 5.9.4 Assigned student personal activities to selected lecturer**

```
$this->db->query('UPDATE per_activity SET l_id = :l_id WHERE pac_id = :pac_id');
```

```
$this->db->query('SELECT * FROM lecturer');
```

```
$this->db->query('SELECT l_fName FROM lecturer WHERE l_id = :l_id');
```

The figure consists of two screenshots of a web application titled "Manage Personal Activity".

**Screenshot 1:** This screenshot shows a table of student personal activities. The columns are: Personal Activity's Name, Date, Venue, Description, Evidence, Student, and Action. One row is visible:

Personal Activity's Name	Date	Venue	Description	Evidence	Student	Action
dd	February 15 12:02	Penang	competition	<a href="#">View</a>	Lee Yik Hong	<a href="#">Approve</a> <a href="#">Reject</a>

Below the table, there is a pagination control showing "Showing 1 to 1 of 1 records" and a page number "1".

**Screenshot 2:** This screenshot shows the same interface but with no data. It displays the message "No data available in table" and "Showing no records".

**Figure 5.9.5 Approve and approved student personal activities by selected lecturer**

```
$this->db->query('SELECT * FROM per_activity WHERE l_id = :l_id AND status = "Waiting"');
```

```
$this->db->query('SELECT * FROM per_activity WHERE s_id = :s_id AND status = "Waiting" ORDER BY date ASC');
```

```
$this->db->query('UPDATE per_activity SET status = "Approved" WHERE pac_id = :pac_id');
```

```
$this->db->query('SELECT s_fName FROM student WHERE s_id = :s_id');
```

## CHAPTER 6

### Summary

In summary, our project, StuPort, aims to transform the manual and inefficient process of managing student portfolios into an automated system. Through extensive collaboration with the client, Mr. Hanif Marzuki Mohd Saipi from Youth Ventures Asia, and other groups, we've identified key challenges and requirements essential for the success of the system.

The proposed system addresses these challenges by introducing modules such as authentication, profile management, activity tracking, feedback systems, resume creation, reward assignment and interactive dashboards. These modules cater to four primary user types: students, master administrators, administrators and lecturers, each with distinct roles and permissions. Technologically, the system leverages a robust stack comprising PHP, MySQL, HTML, CSS, JavaScript and Bootstrap. GitHub facilitates version control, ensuring seamless collaboration during the system's development.

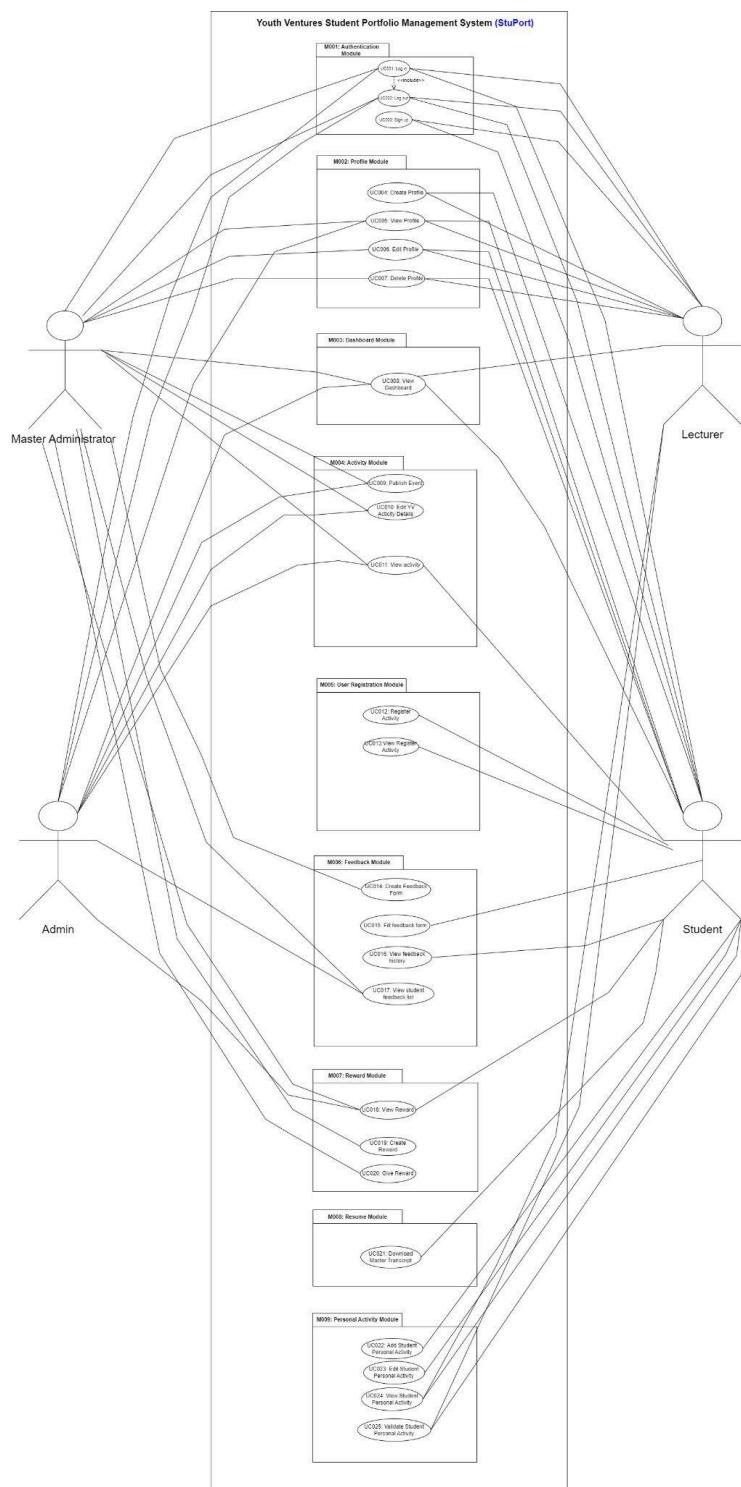
For the Database of the system, we have carefully planned the logical structure of the database. We transformed the conceptual ERD into a logical ERD, derived relational schema and performed normalization up to BCNF where applicable. The final logical ERD, relational database schemas, data dictionary and interface design have been developed to align with the system's transaction requirements. Our SQL statements, encompassing Data Definition Language (DDL) and Data Manipulation Language (DML), are customized to efficiently interact with the database and fulfill the system's functionalities.

In conclusion, the Youth Ventures Student Portfolio Management System, StuPort, is set to revolutionize student profile management, streamline data processes and foster collaboration among Youth Ventures Asia and their clients through its innovative features, user-friendly interface and robust technology stack.

# CHAPTER 7

## Appendices

### 7.1 Use Case Diagram for The System



\*\*Higher Definition Overview Use Case Diagram: Updated Overview UseCase.jpg\*\*

## 7.2 Activity Diagrams for Every Use Case Identified

### Module 001: Authentication Module

UC001: Use Case <Log In>

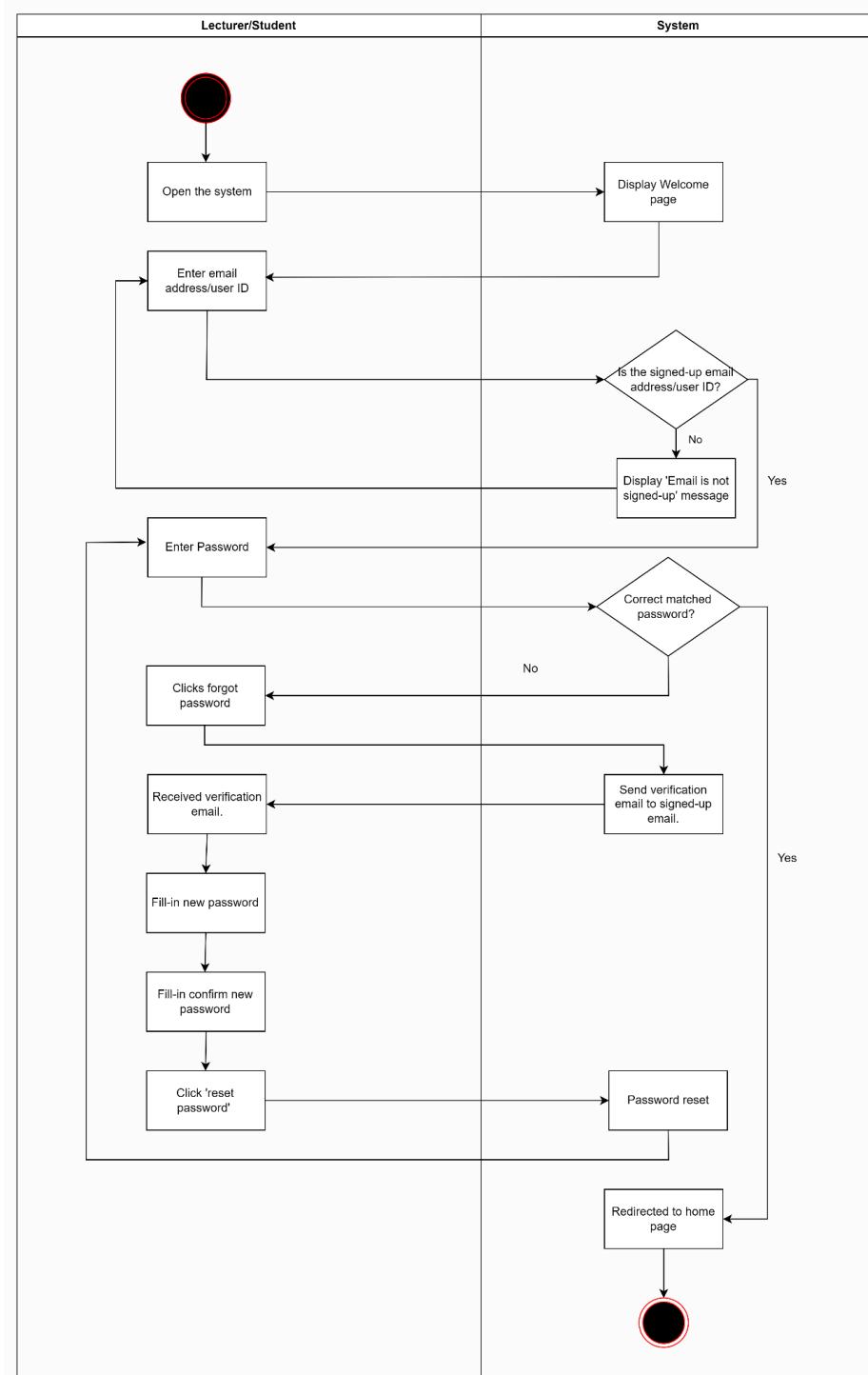


Figure 7.2.1.1: Activity Diagram of Log In

## UC002: Use Case <Log Out>

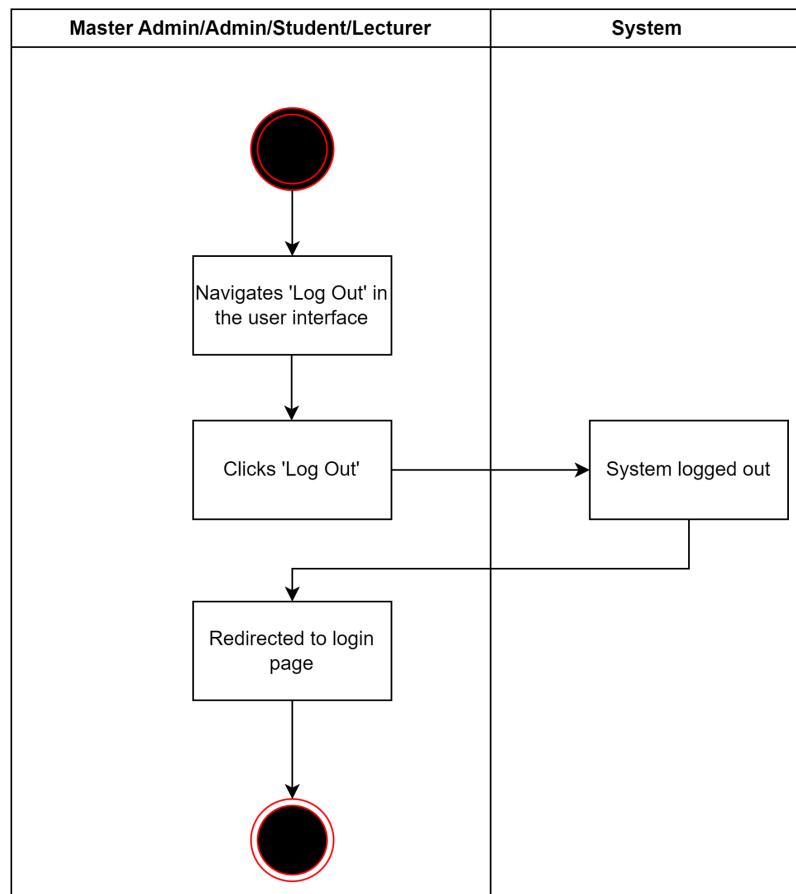
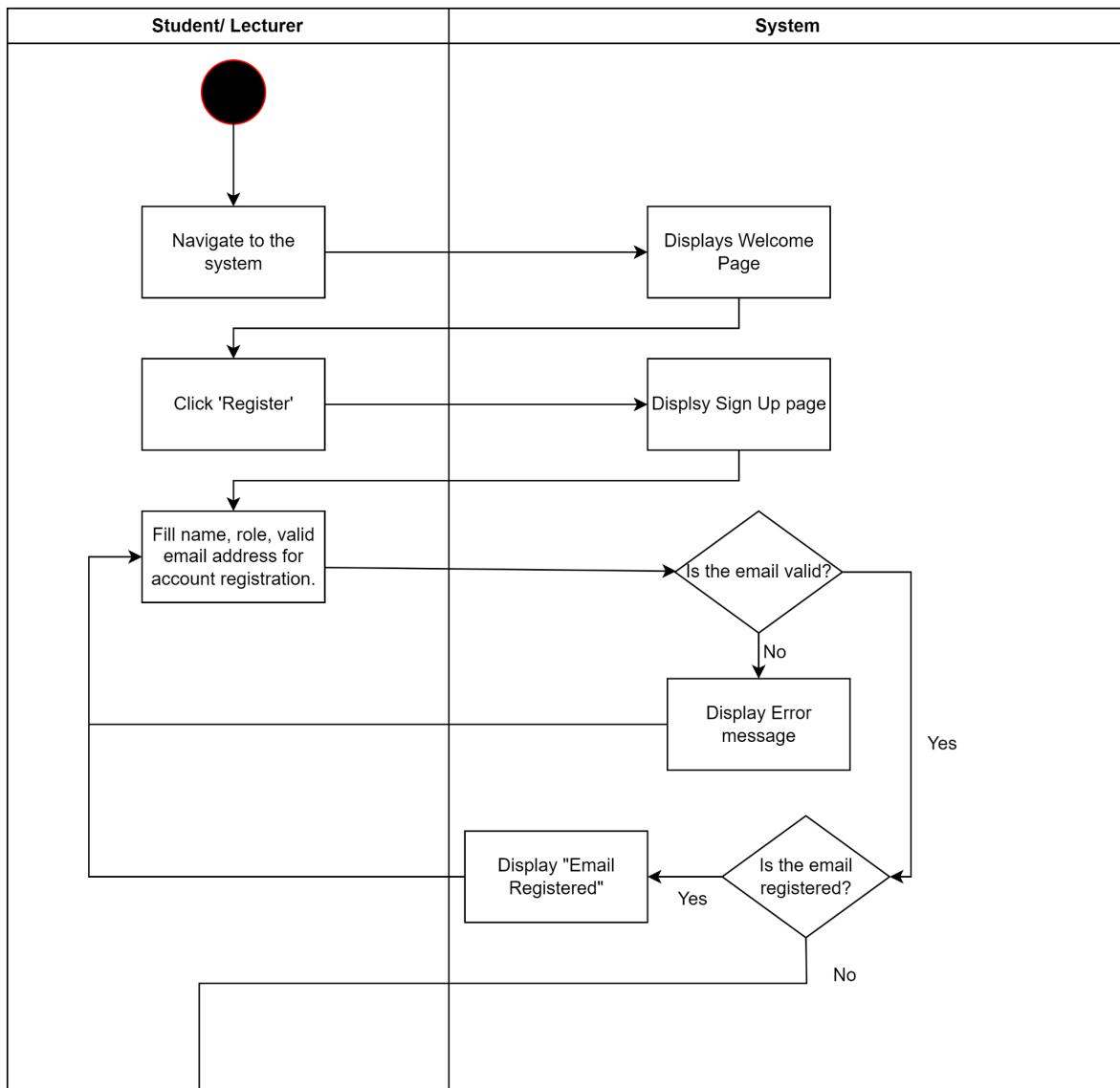


Figure 7.2.1.2: Activity Diagram of Log Out

### UC003: Use Case <Sign Up>



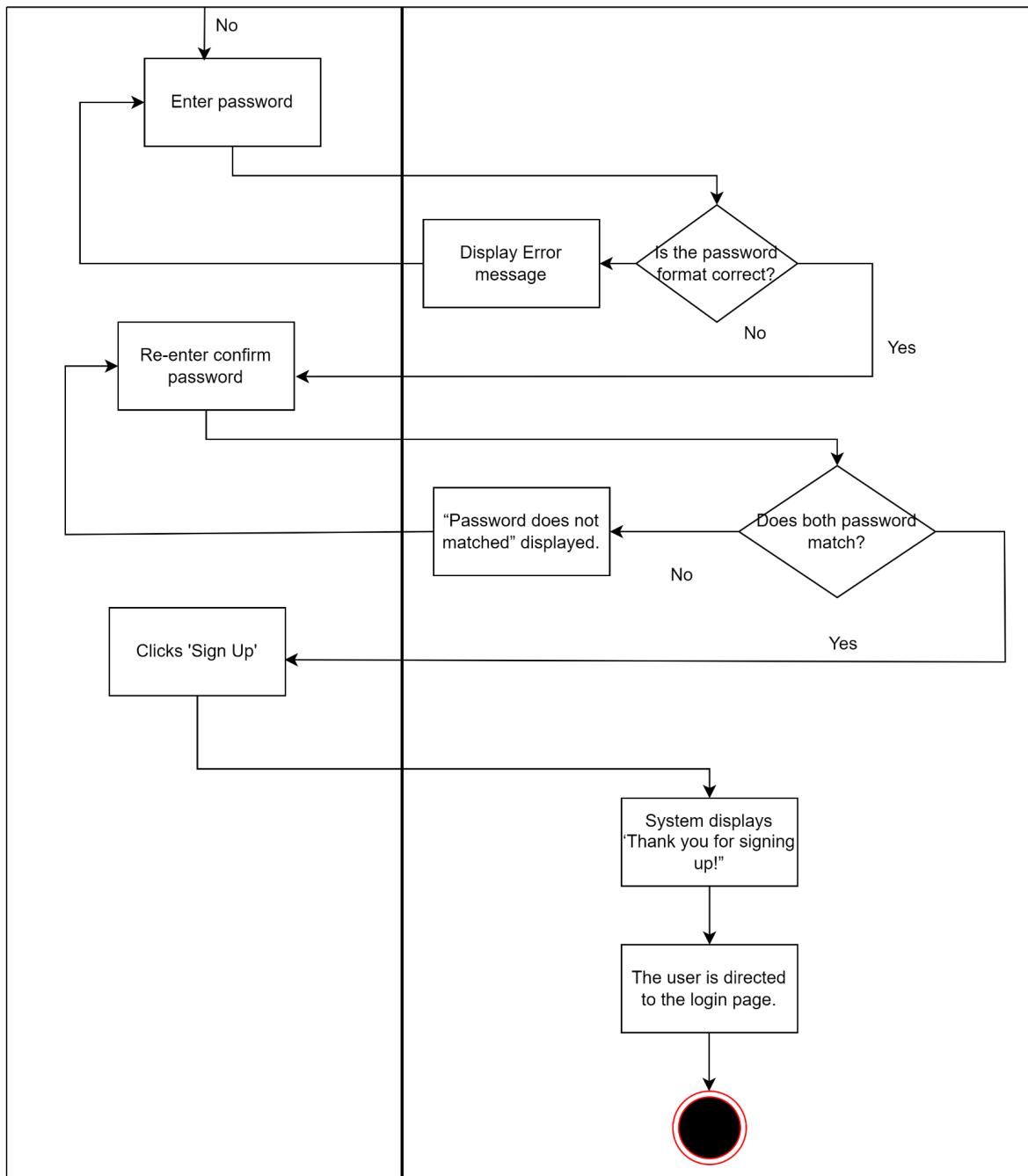
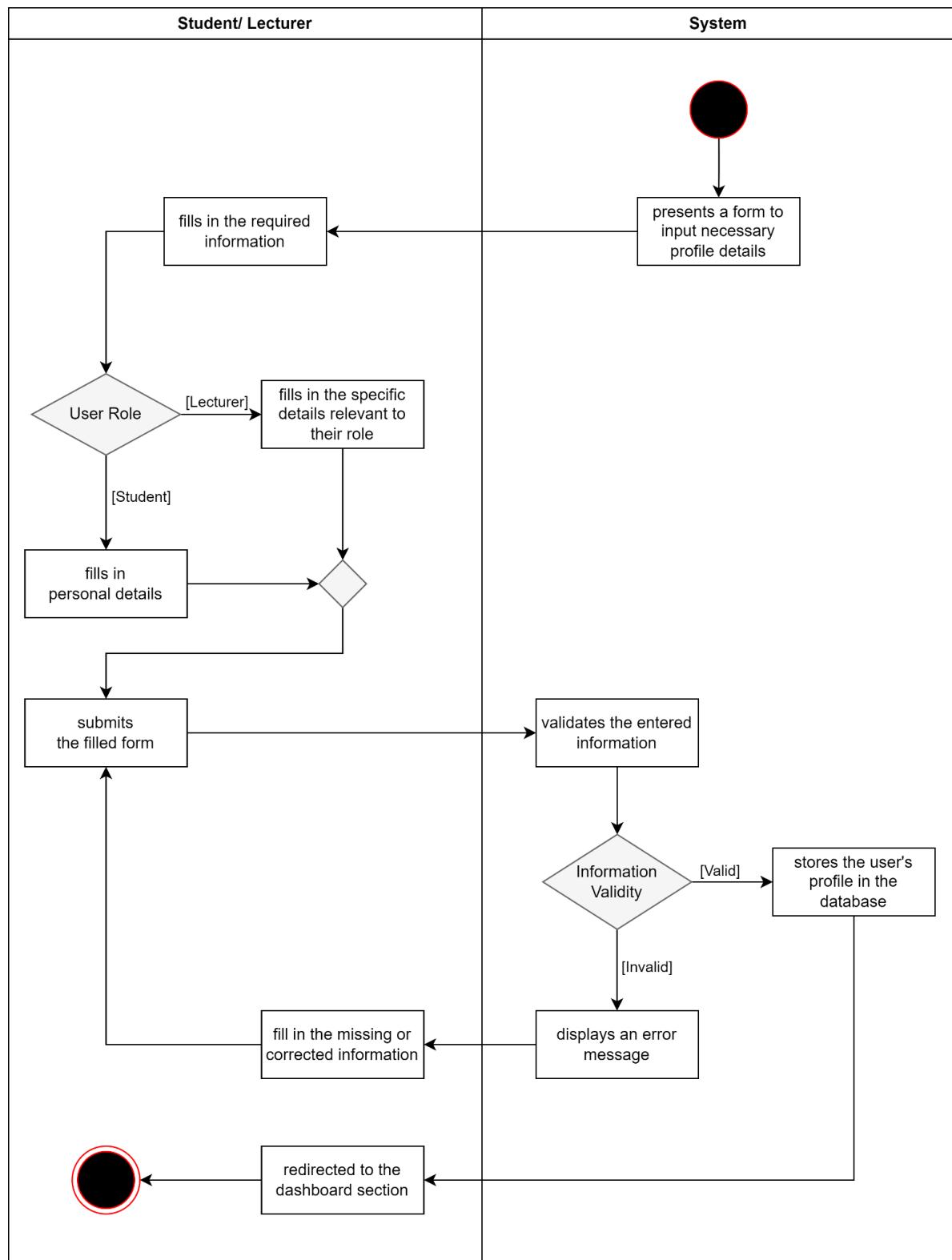


Figure 7.2.1.3: Activity Diagram of Sign Up

## **Module 002: Profile Module**

UC004: Use Case <Create Profile>



*Figure 7.2.2.1: Activity Diagram of Create Profile*

### UC005: Use Case <View Profile>

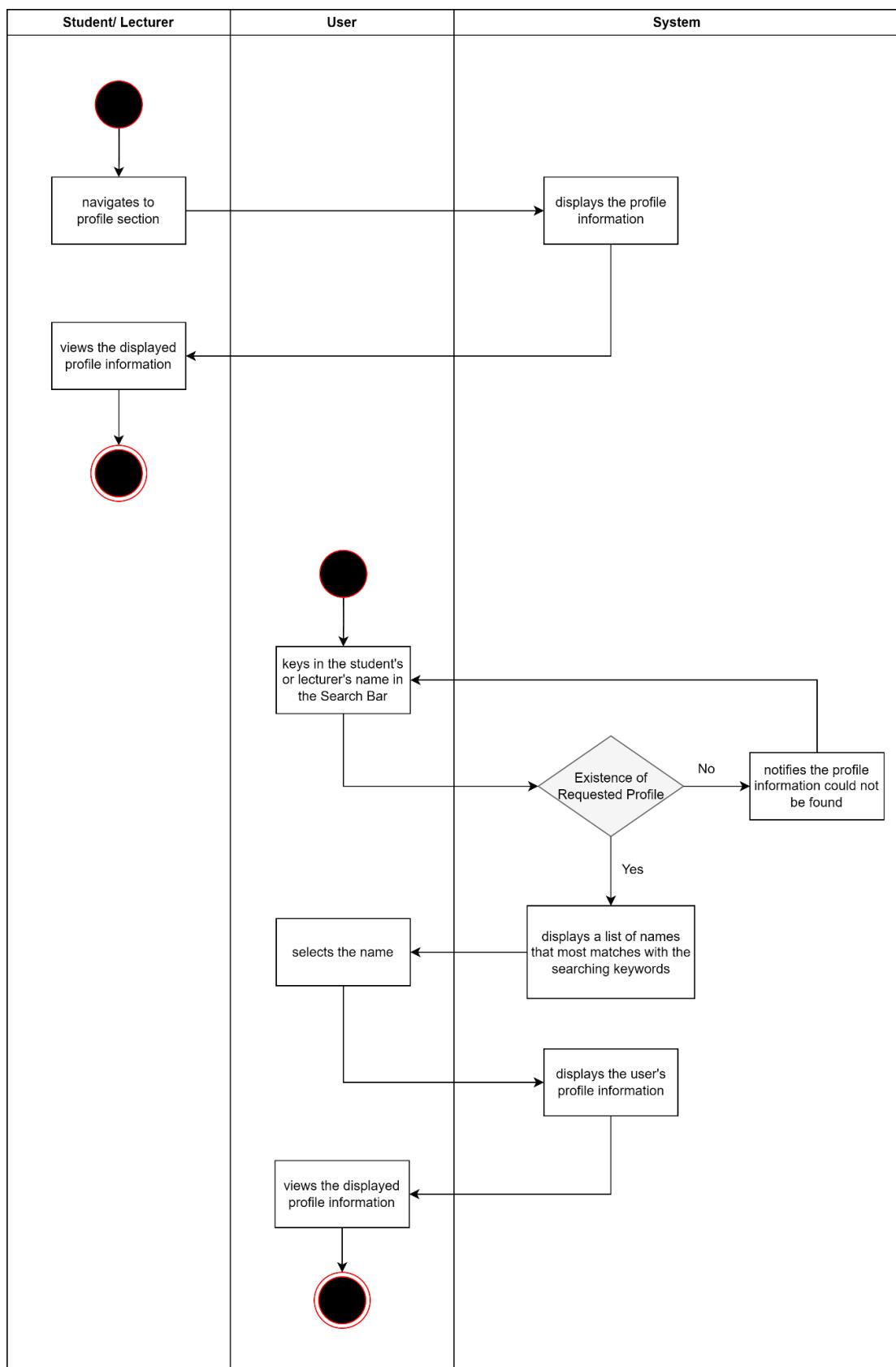


Figure 7.2.2.2: Activity Diagram of View Profile

## UC006: Use Case <Edit Profile>

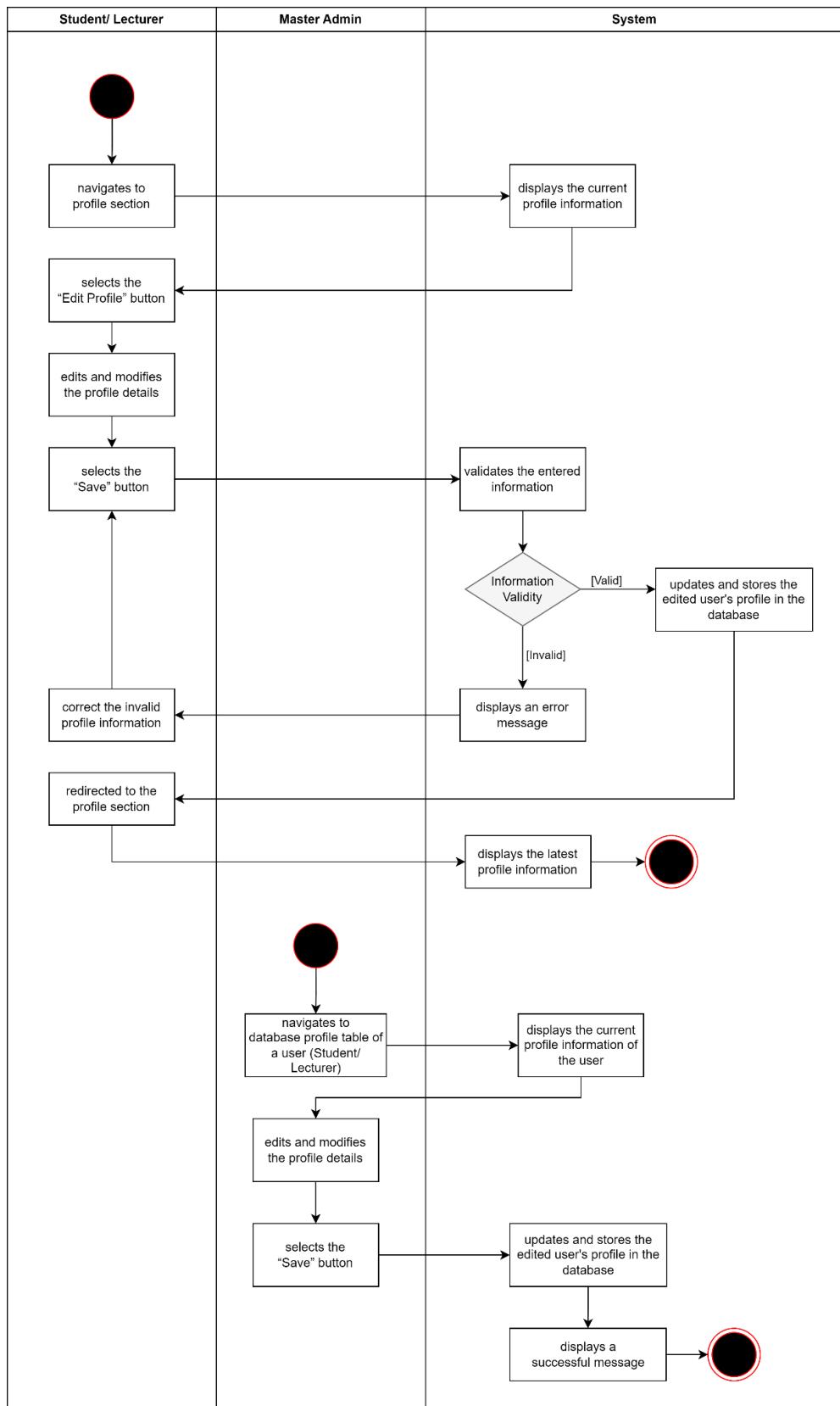


Figure 7.2.2.3: Activity Diagram of Edit Profile

## UC007: Use Case <Delete Profile>

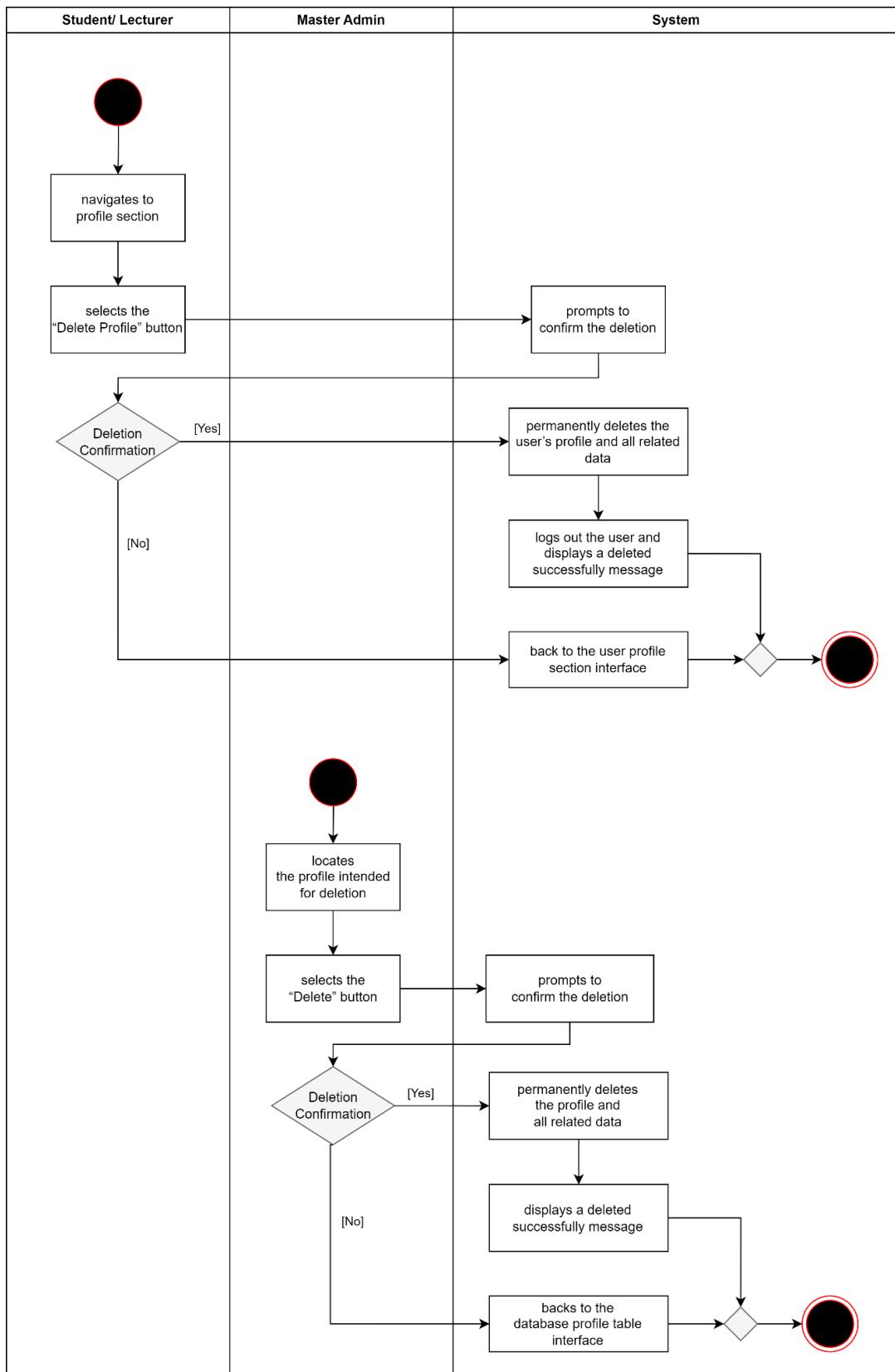
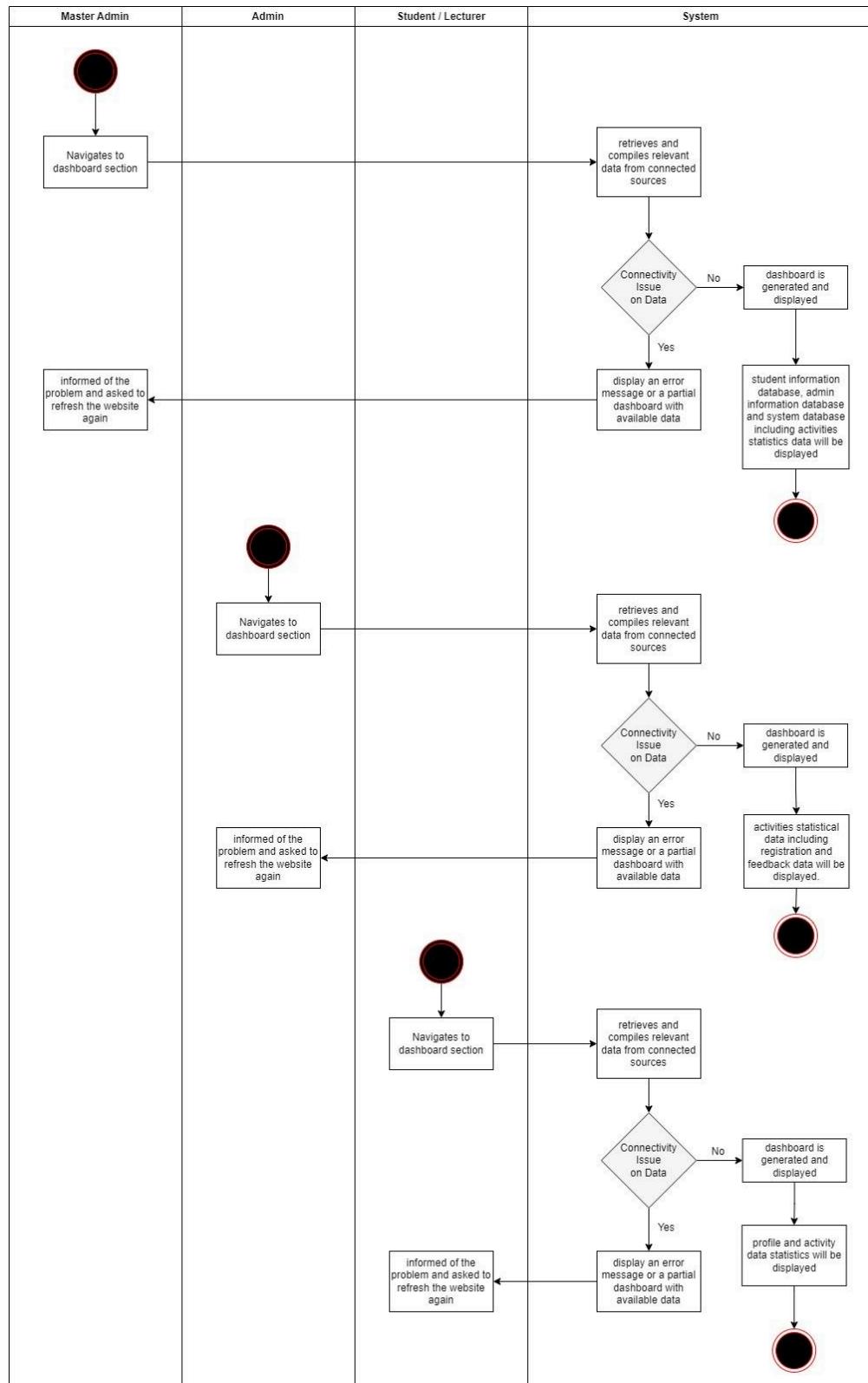


Figure 7.2.2.4: Activity Diagram of Delete Profile

## **Module 003: Dashboard Module**

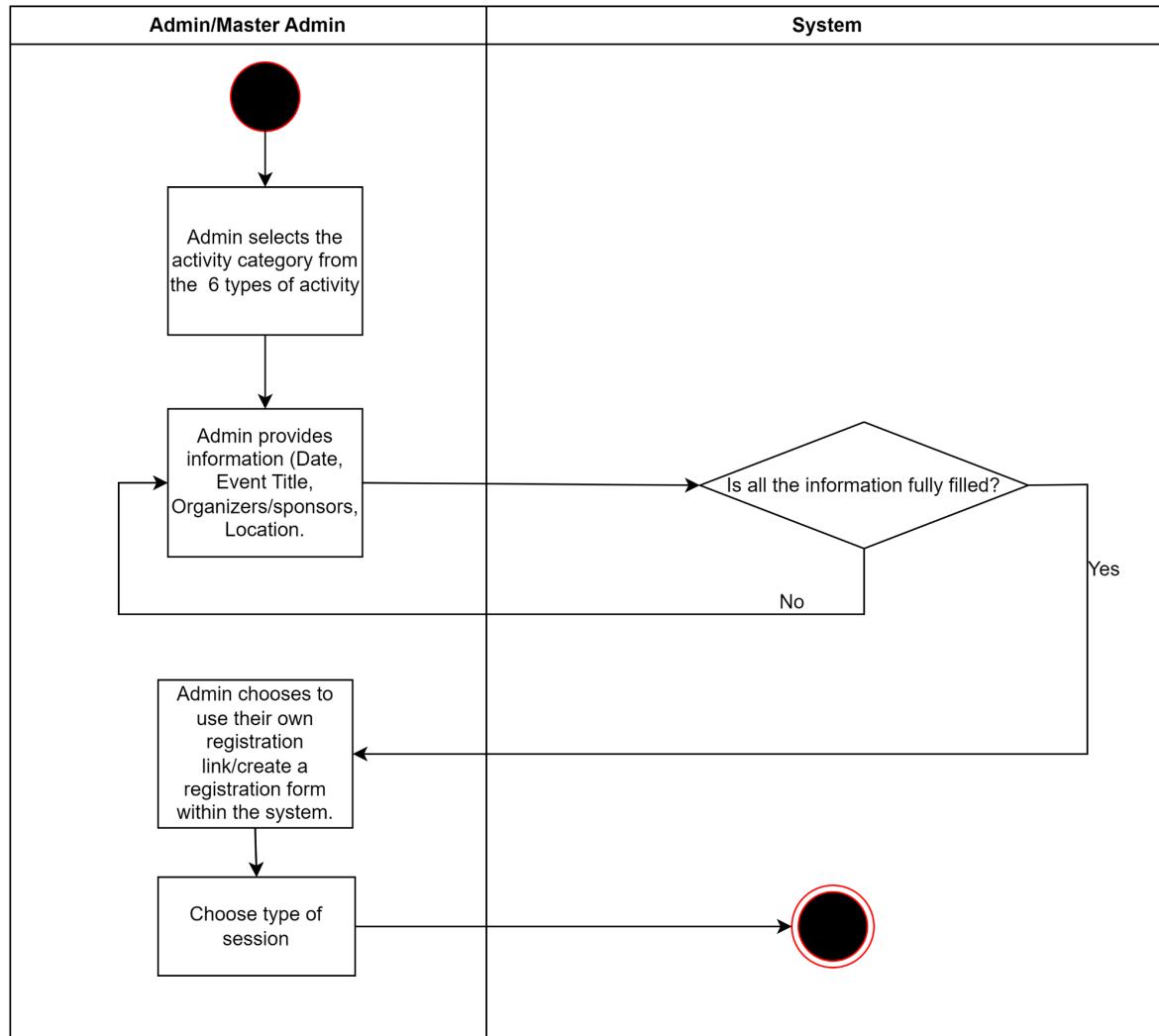
UC008: Use Case <View Dashboard>



*Figure 7.2.3.1: Activity Diagram of View Dashboard*

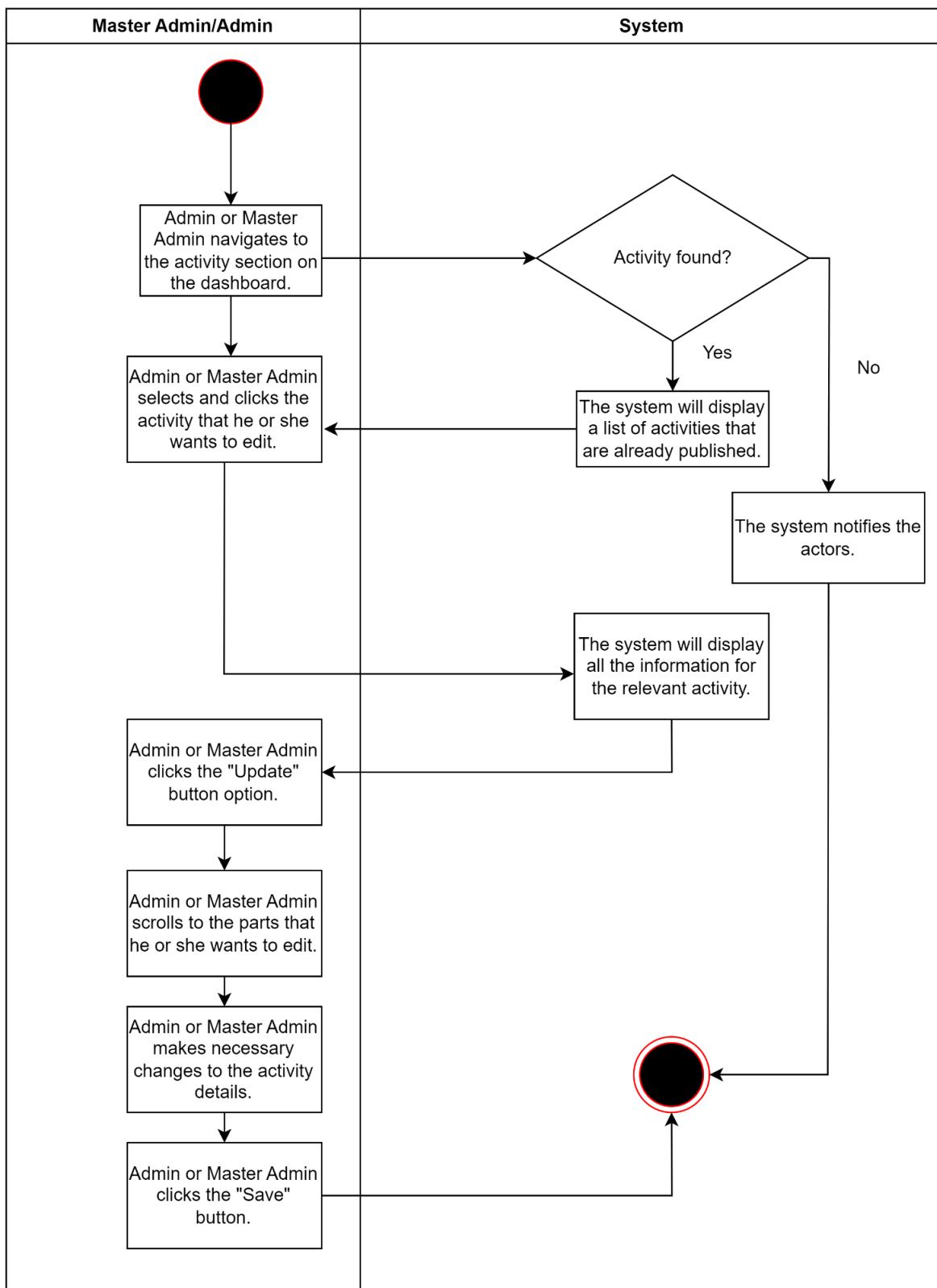
## **Module 004: Activity Module**

UC009: Use Case <Publish Event>



*Figure 7.2.4.1: Activity Diagram of Publish Event*

UC010: Use Case <Edit YV Activity Details>



*Figure 7.2.4.2: Activity Diagram of Edit YV Activity Details*

## UC011: Use Case <View Activity>

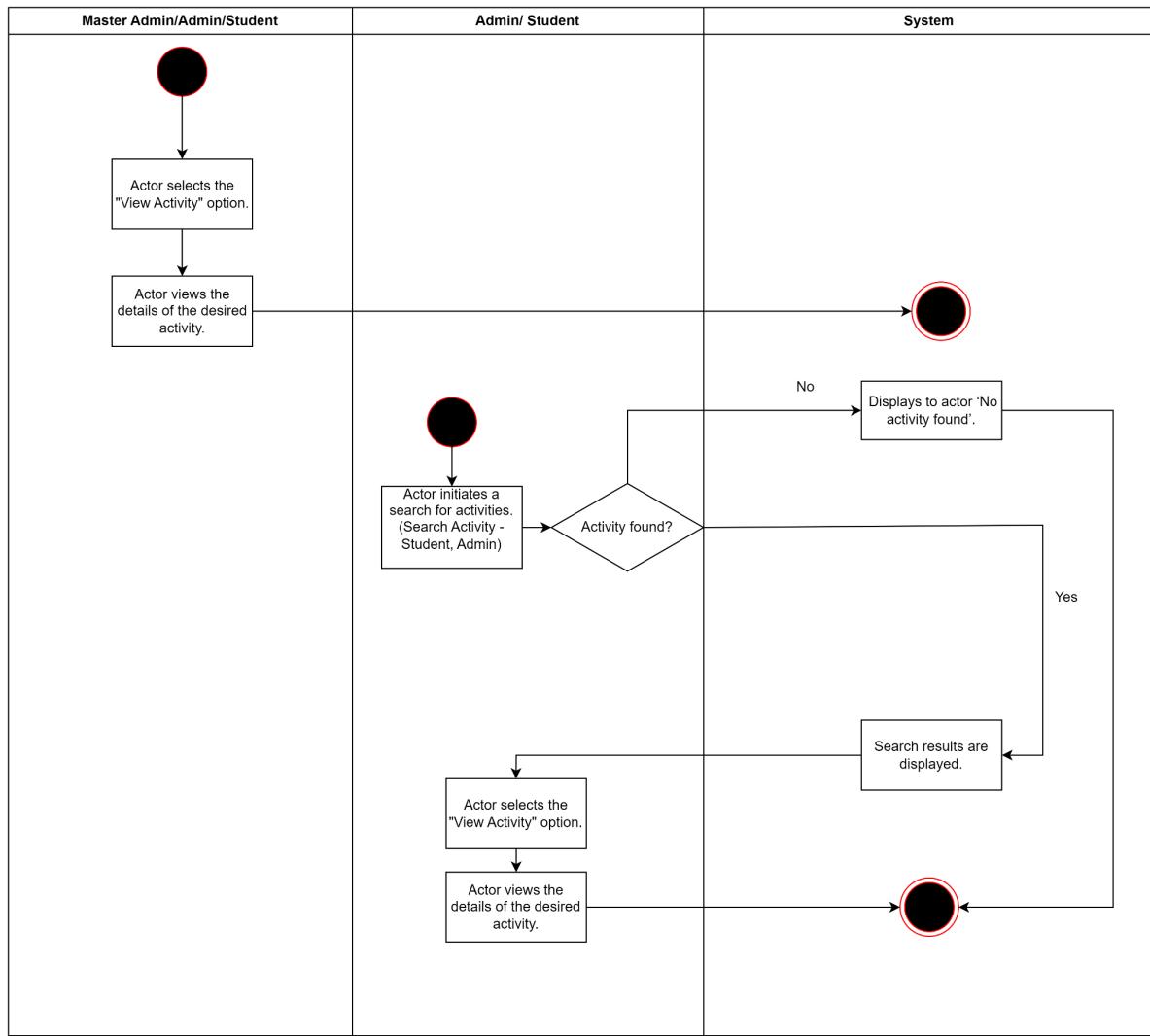
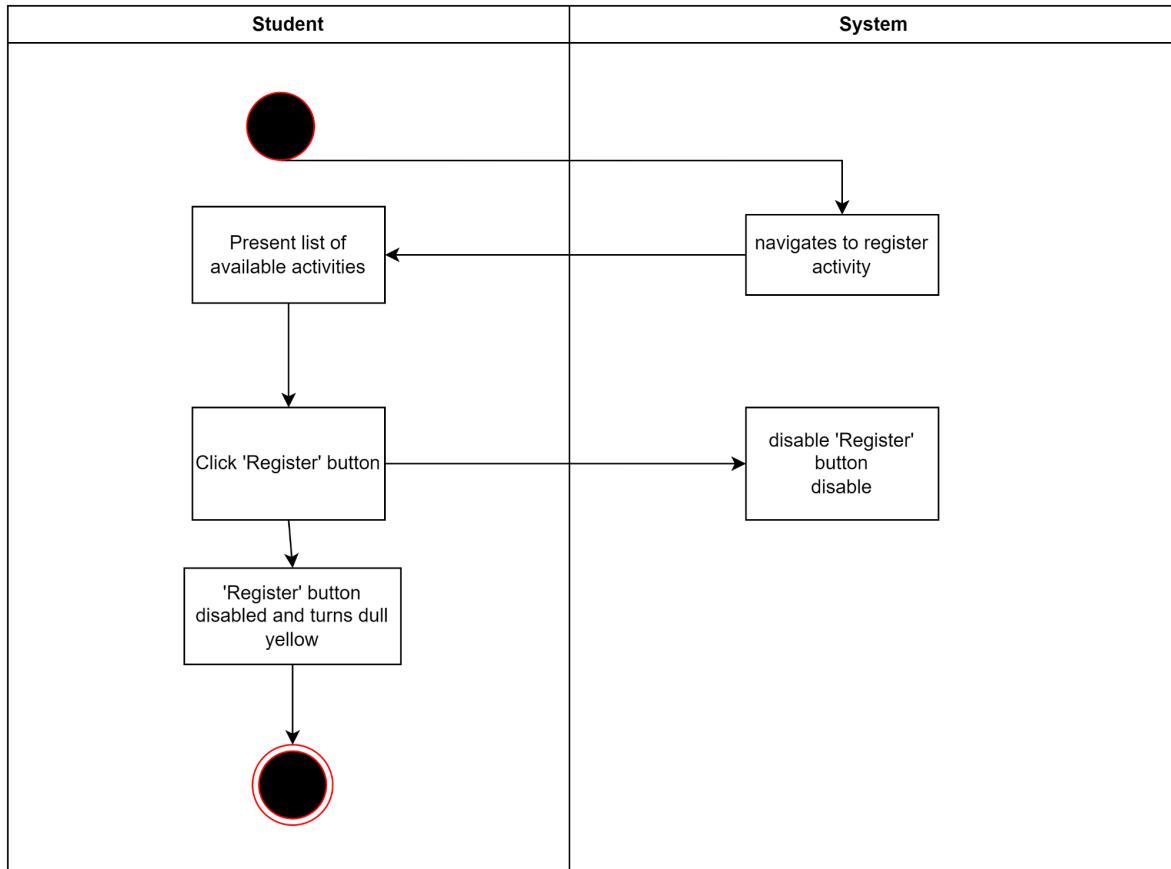


Figure 7.2.4.3: Activity Diagram of View Activity

## **Module 005: Registration Module**

UC012: Use Case <Register Activity>



*Figure 7.2.5.3: Activity Diagram of Register Activity*

UC013: Use Case <View Register Activity>

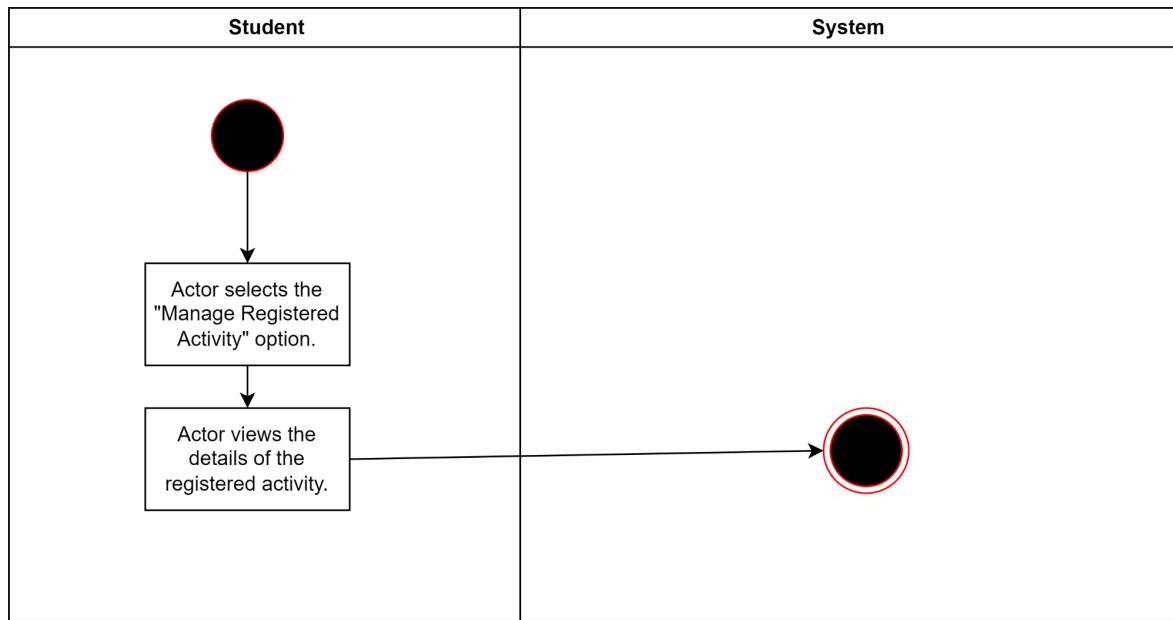
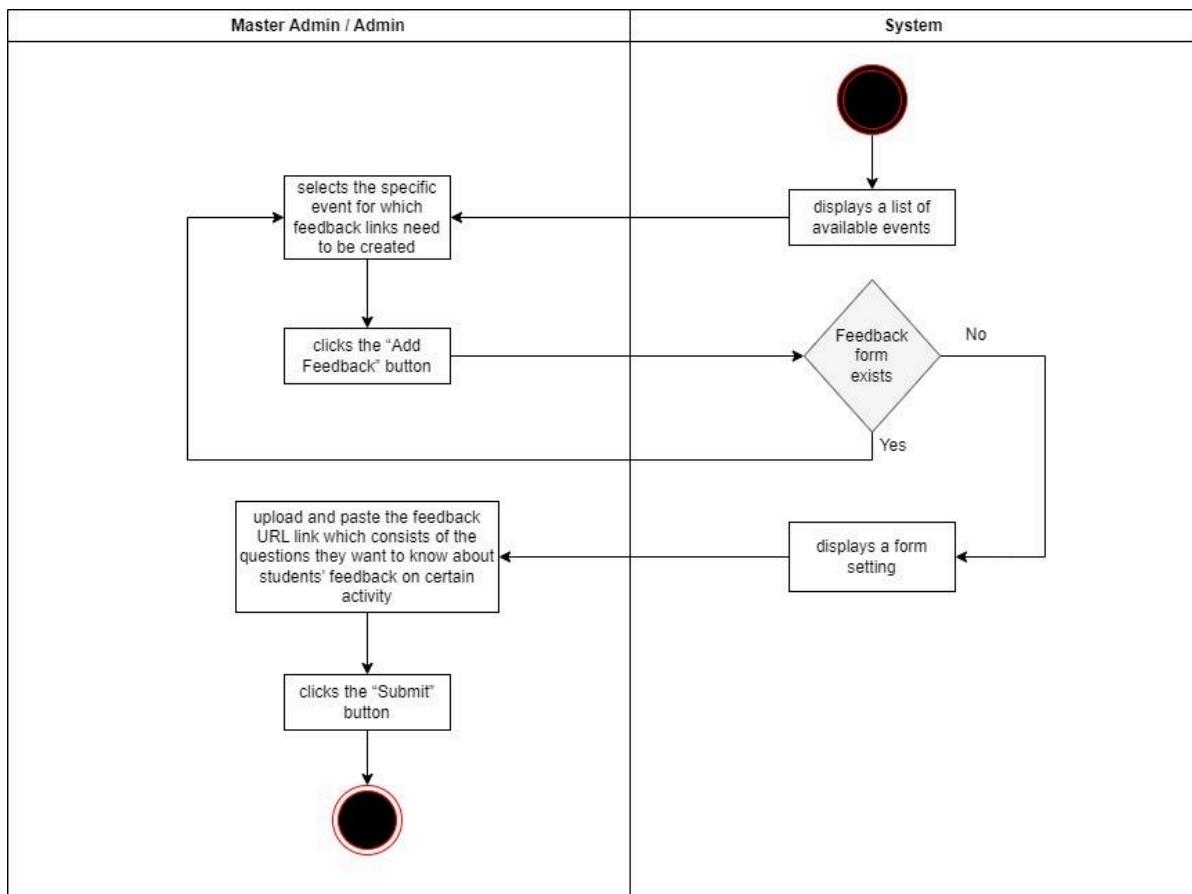


Figure 7.2.5.4: Activity Diagram of View Registered Activity

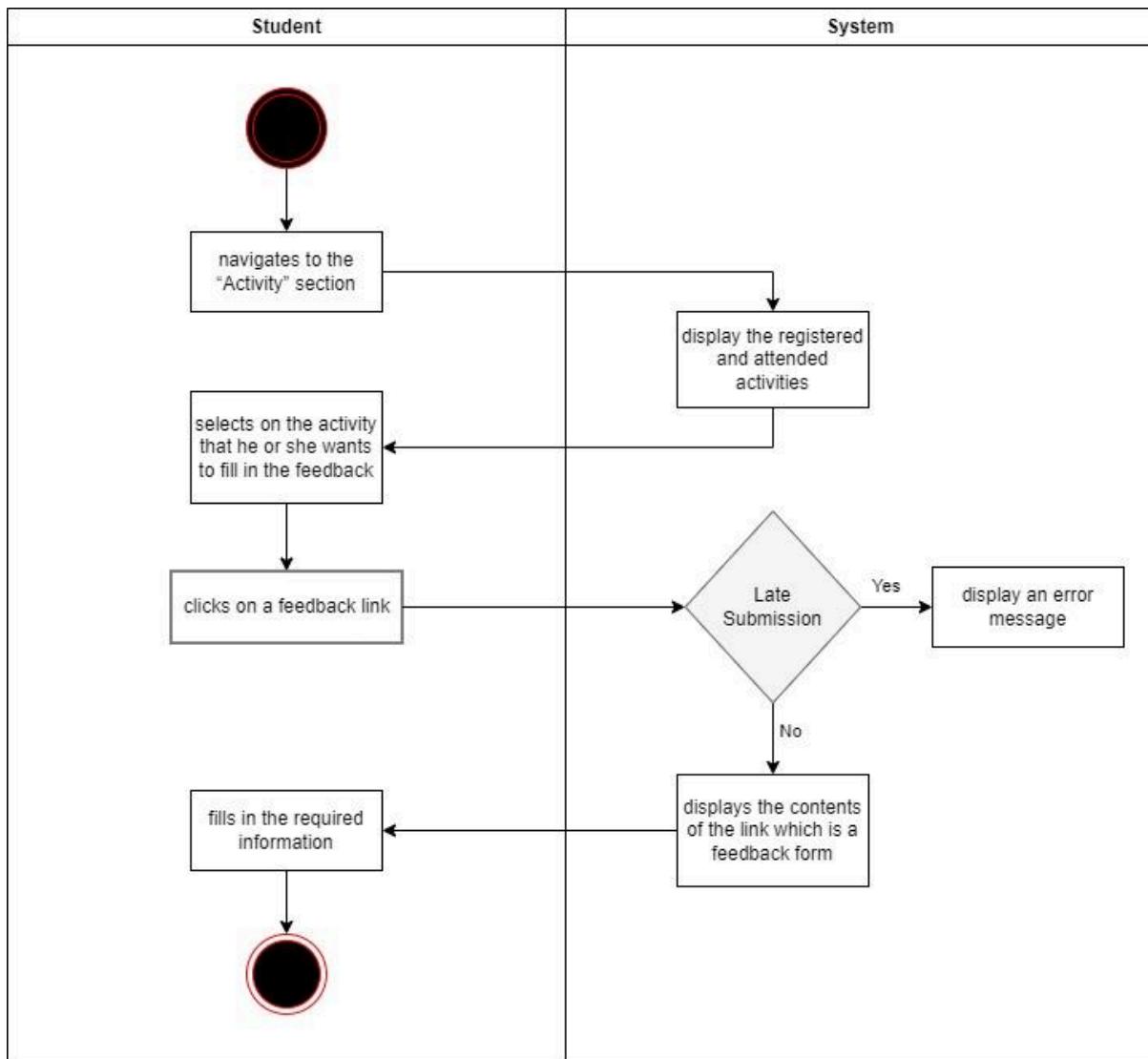
## **Module 006: Feedback Module**

UC014: Use Case <Create Feedback Form>



*Figure 7.2.6.1: Activity Diagram of Create Feedback Form*

UC015: Use Case <Fill Feedback Form>



*Figure 7.2.6.2: Activity Diagram of Fill Feedback Form*

UC016: Use Case <View Feedback History>

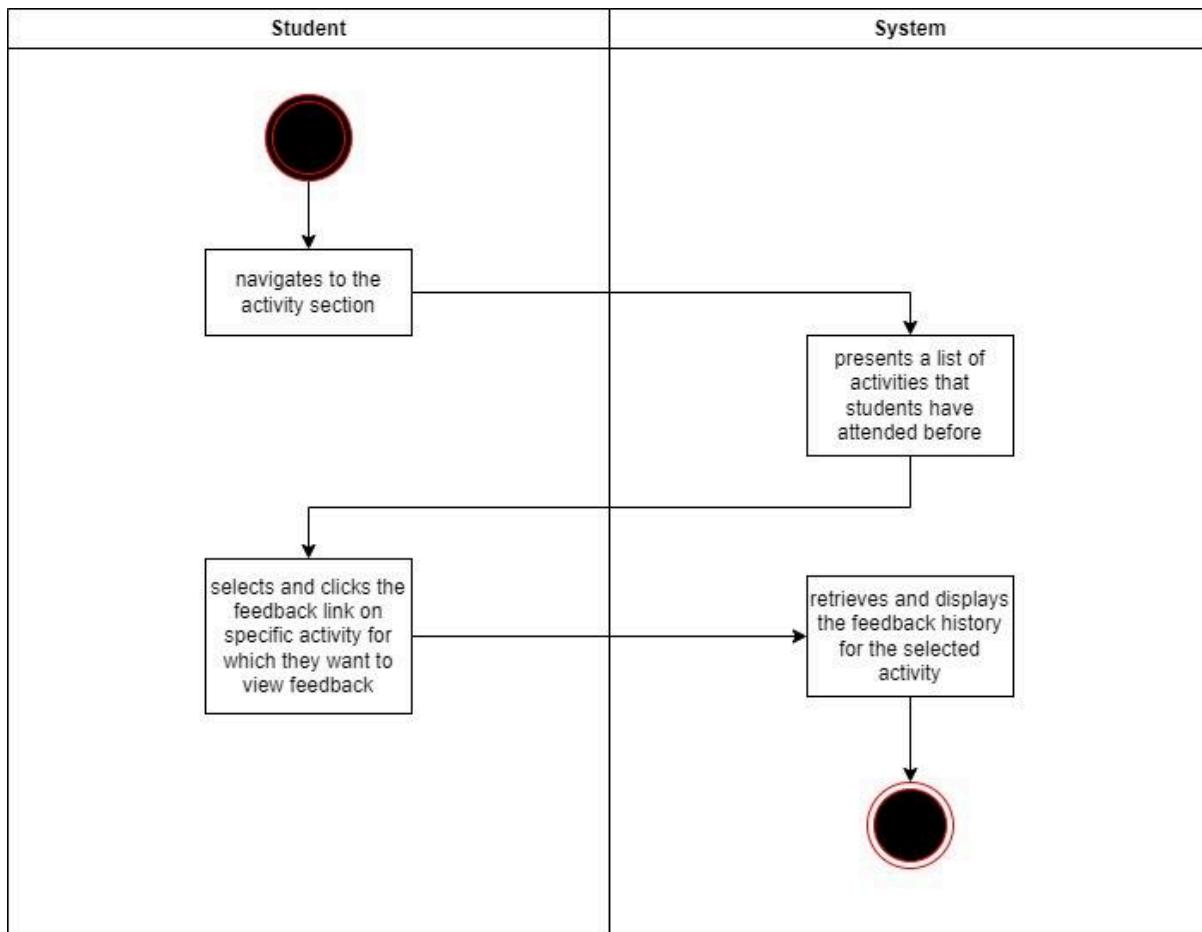
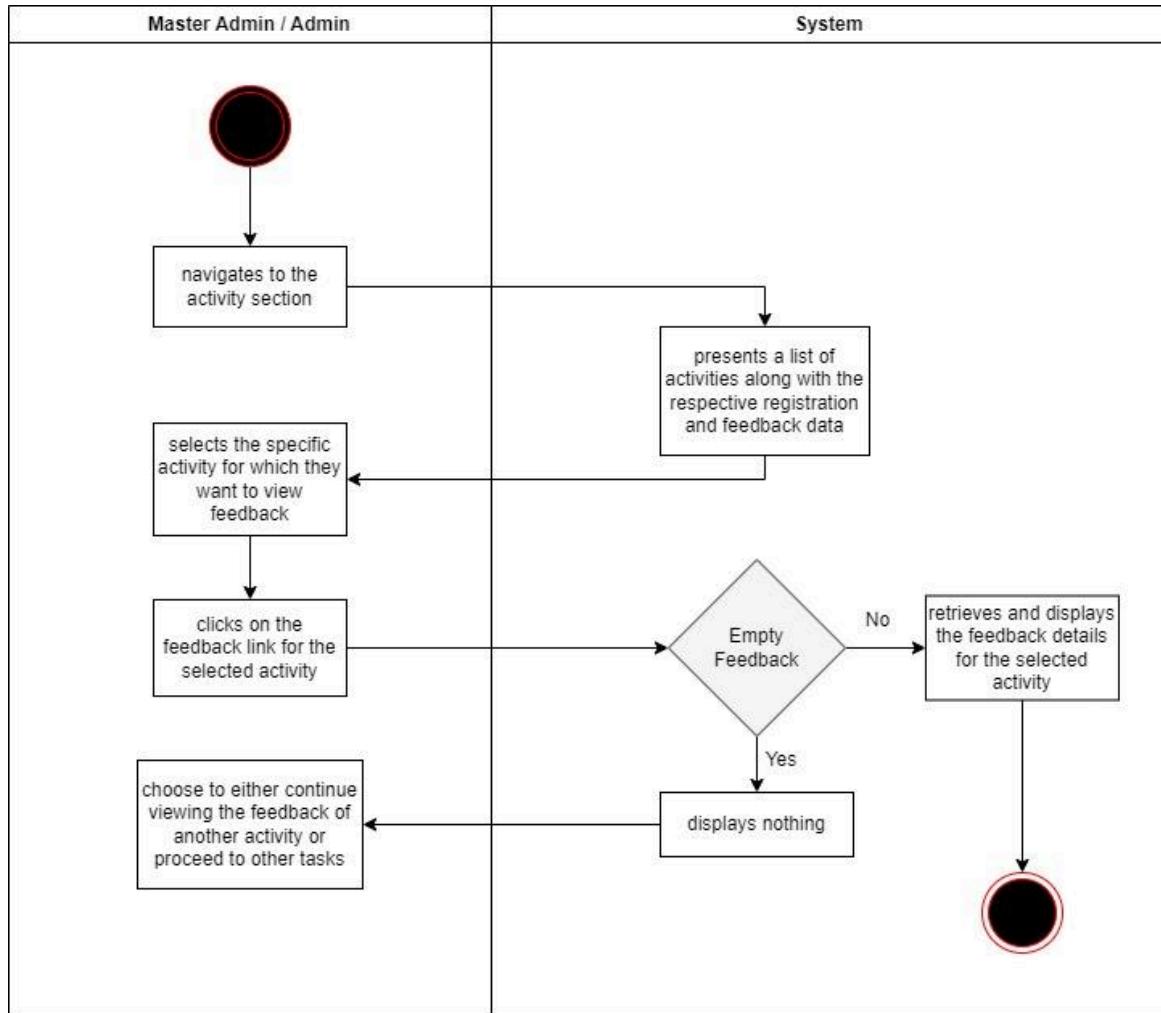


Figure 7.2.6.3: Activity Diagram of View Feedback History

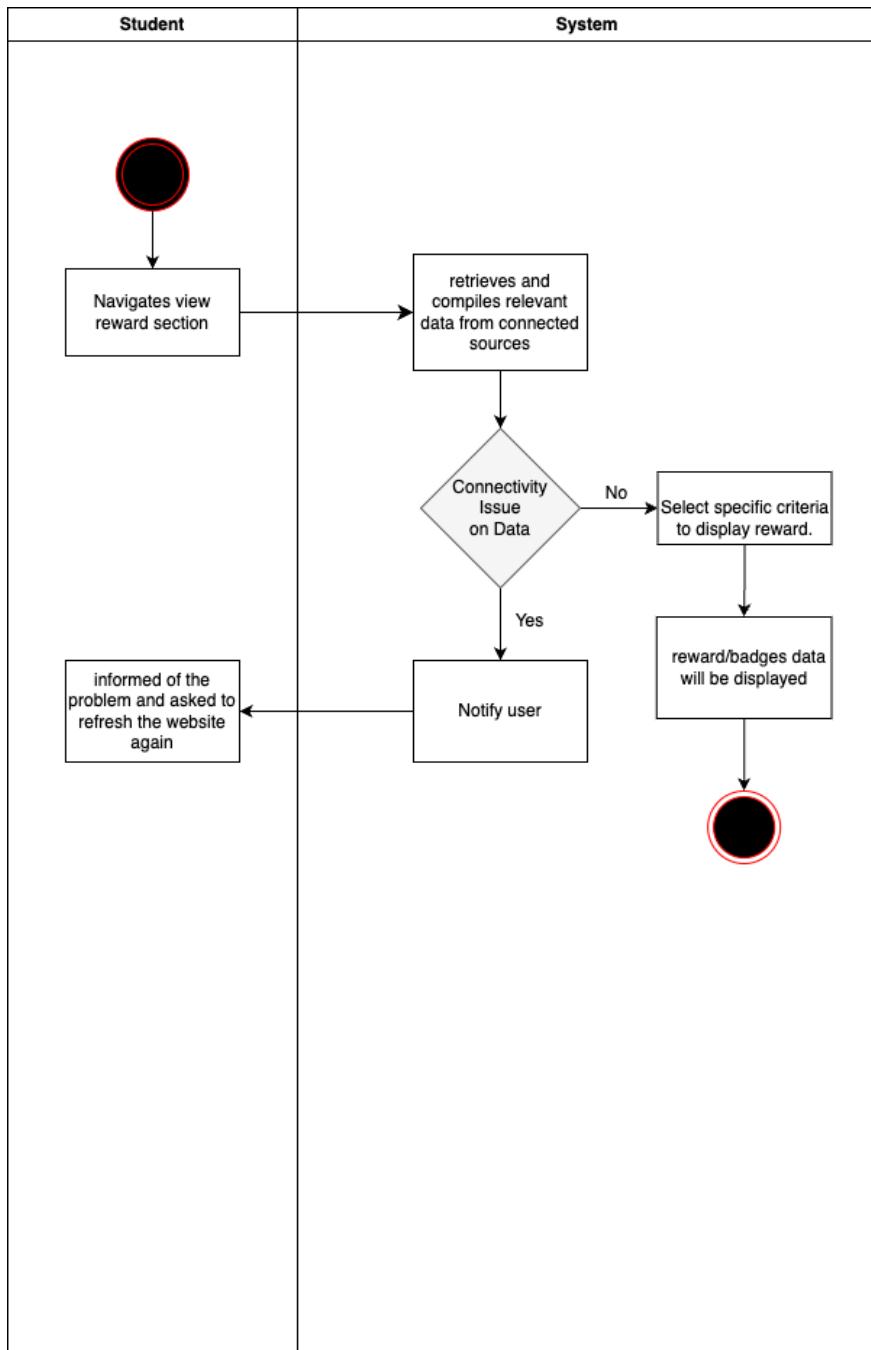
UC017: Use Case <View Student Feedback List>



*Figure 7.2.6.4: Activity Diagram of View Student Feedback List*

## **Module 007: Reward Module**

UC018: Use Case <View Reward>



*Figure 7.2.7.1: Activity Diagram of View Reward*

### UC019: Use Case <Create Reward>

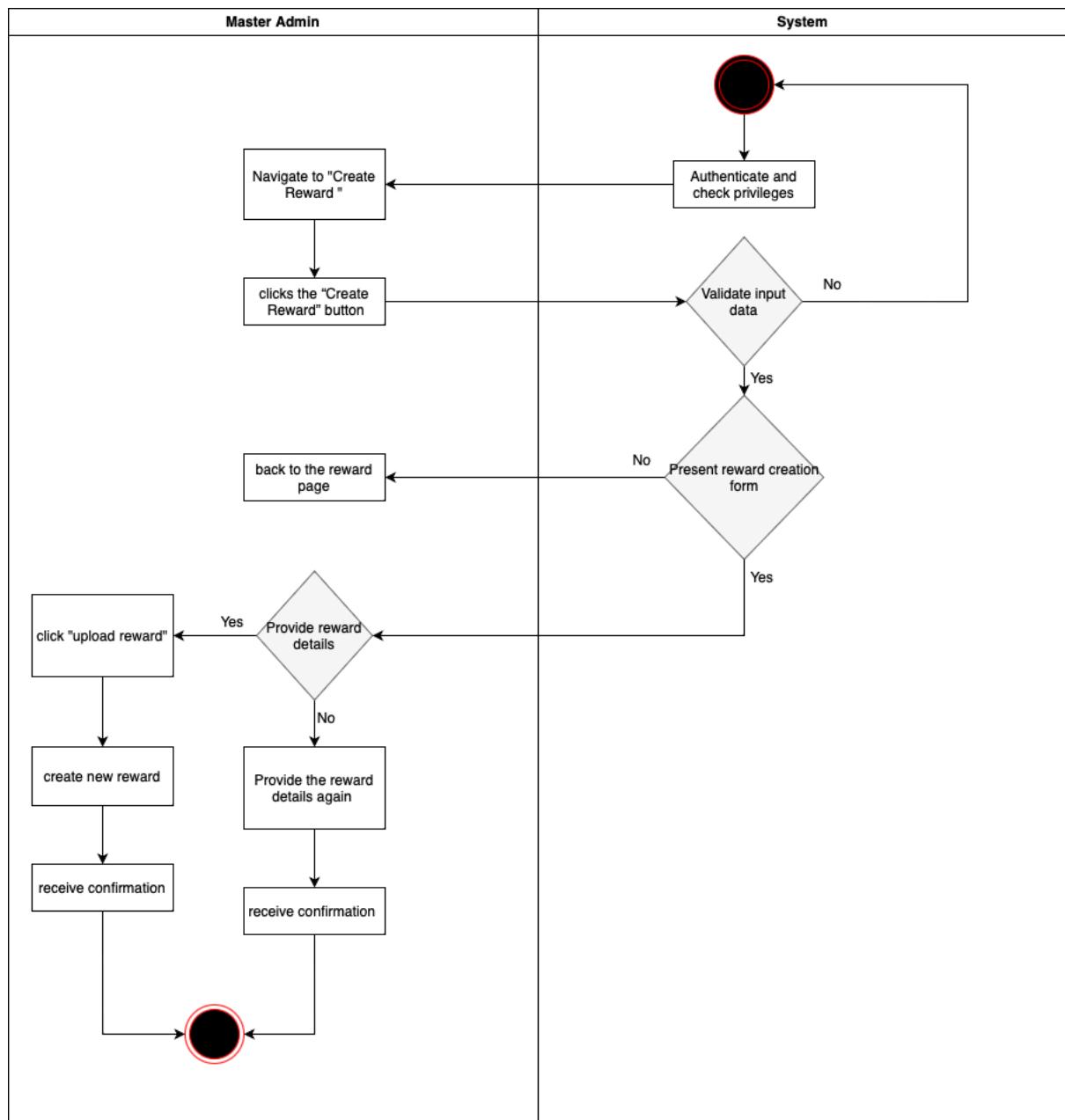


Figure 7.2.7.2: Activity Diagram of Create Reward

## UC020: Use Case <Give Reward>

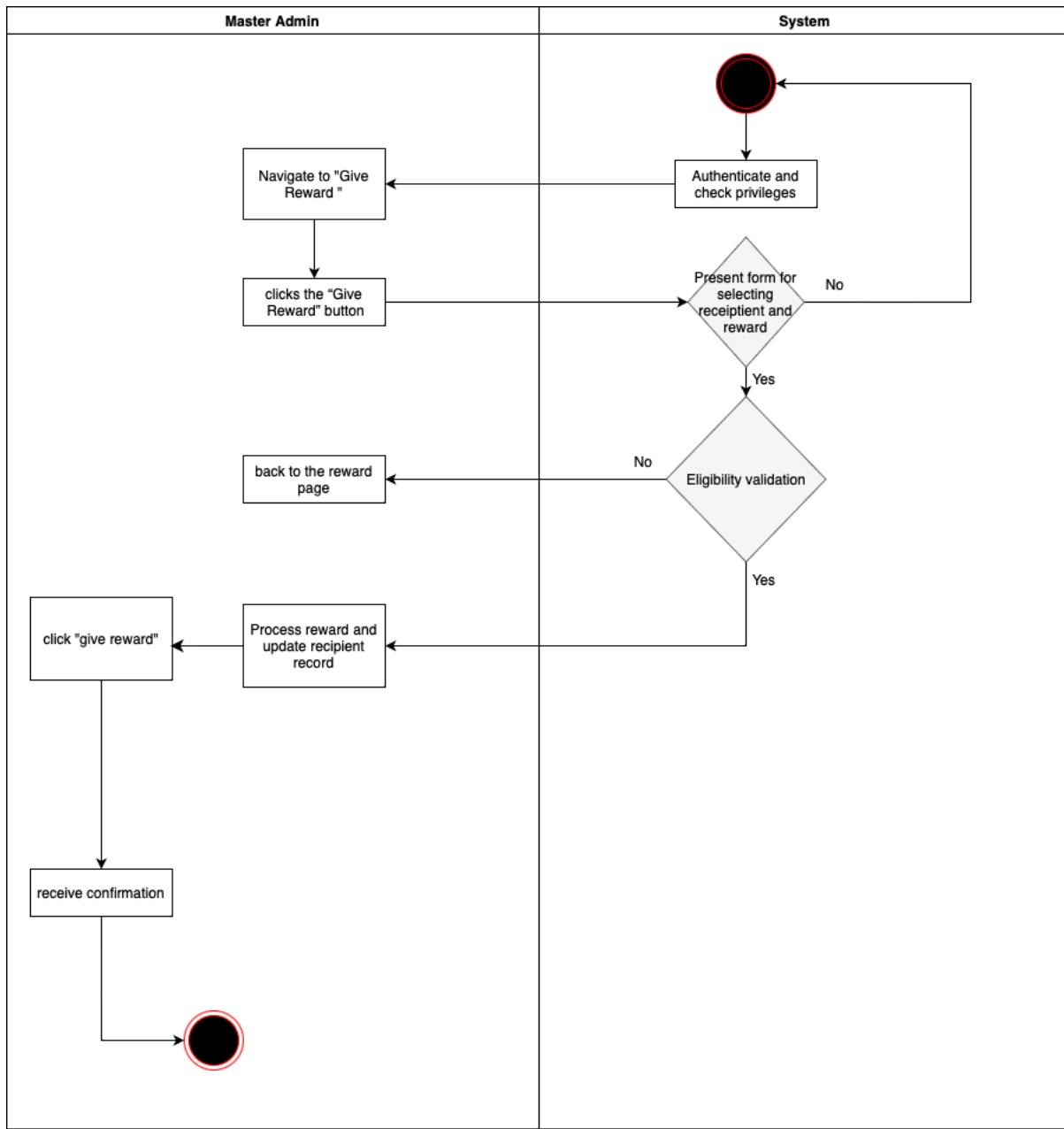
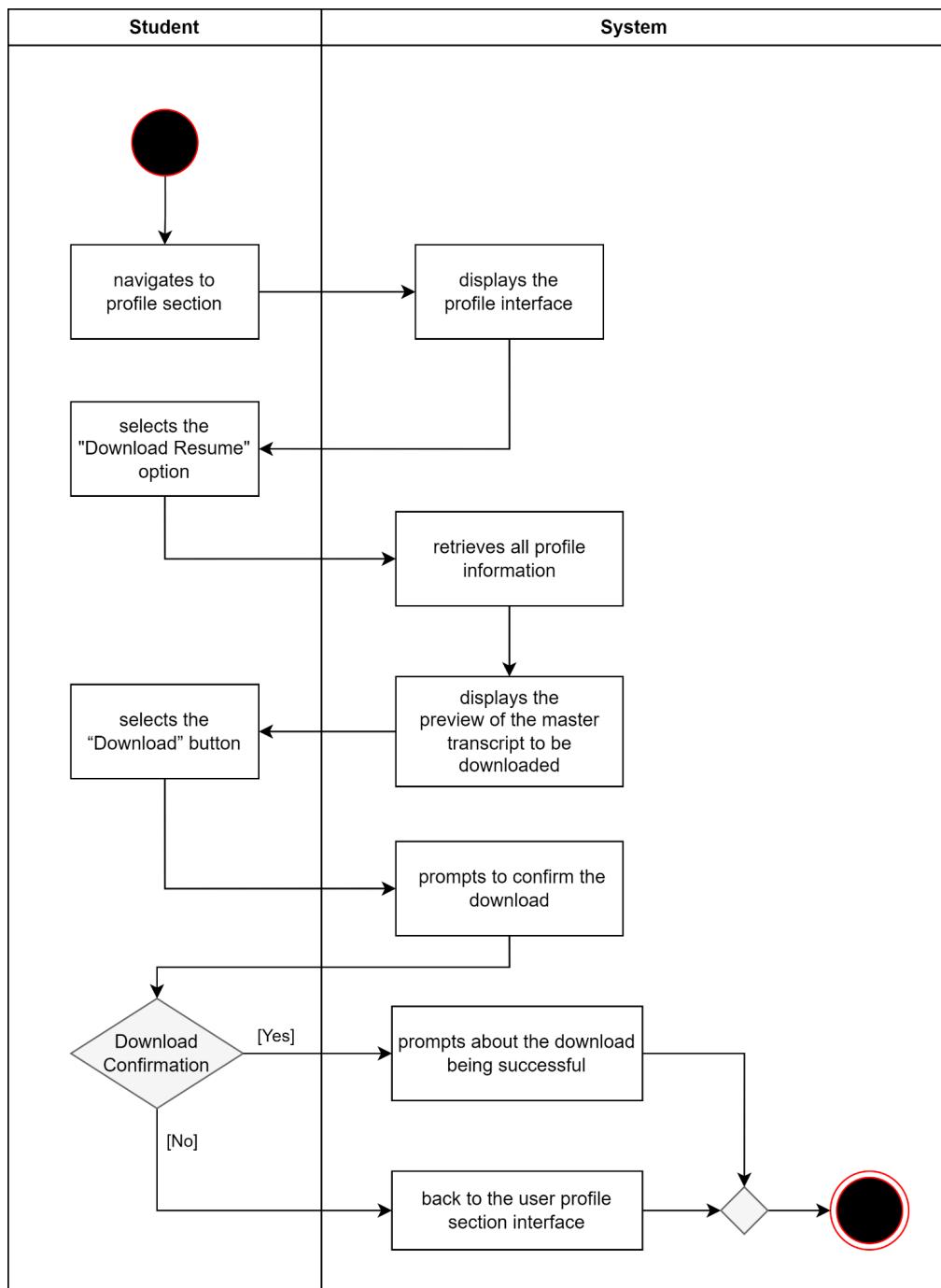


Figure 7.2.7.3: Activity Diagram of Give Reward

## **Module 008: Resume Module**

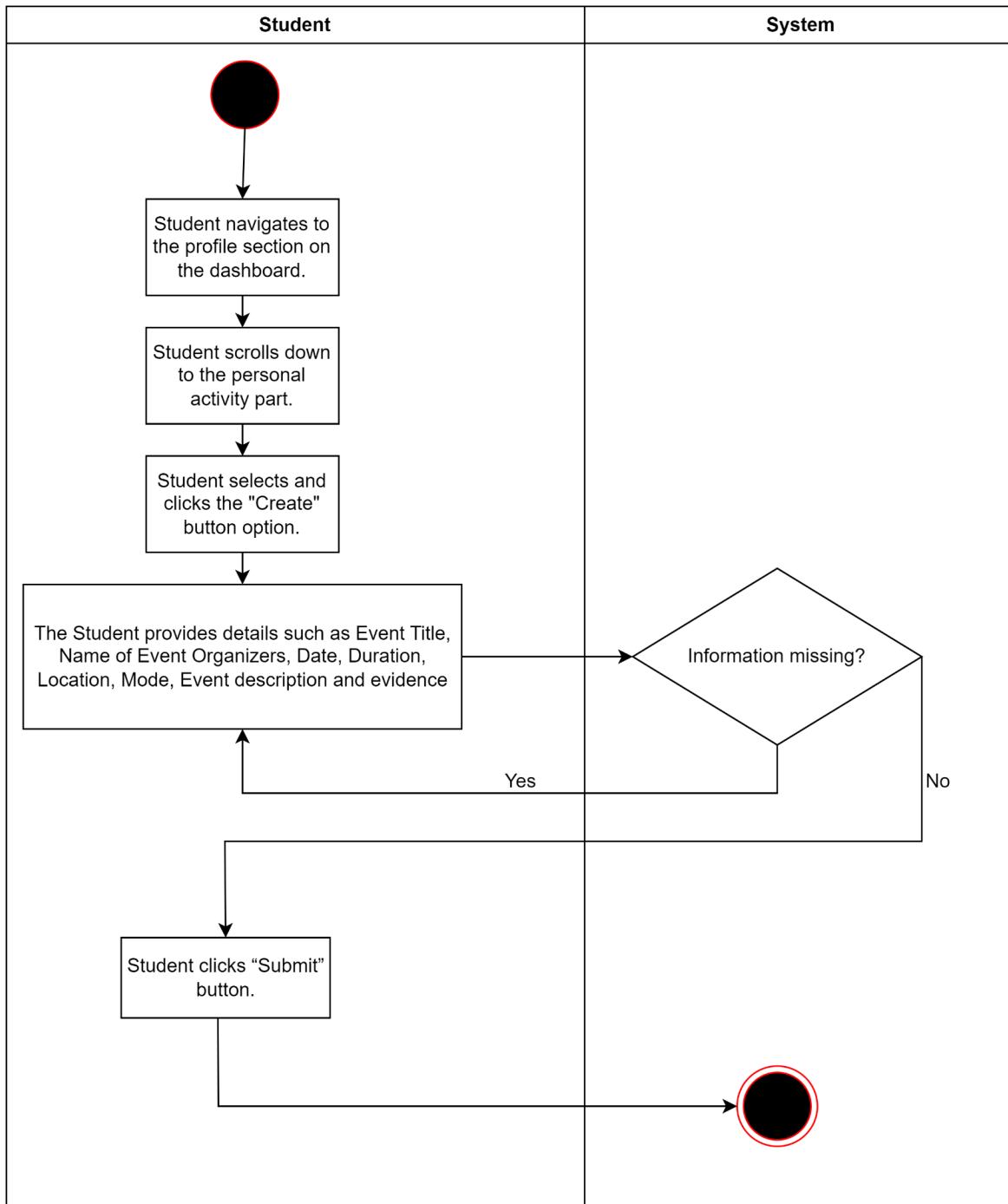
UC021: Use Case <Download Master Transcript>



*Figure 7.2.8.1: Activity Diagram of Download Master Transcript*

## **Module 009: Personal Activity Module**

UC022: Use Case <Add Student Personal Activity>



*Figure 7.2.9.1: Activity Diagram of Add Student Personal Activity*

## UC023: Use Case <Edit Student Personal Activity>

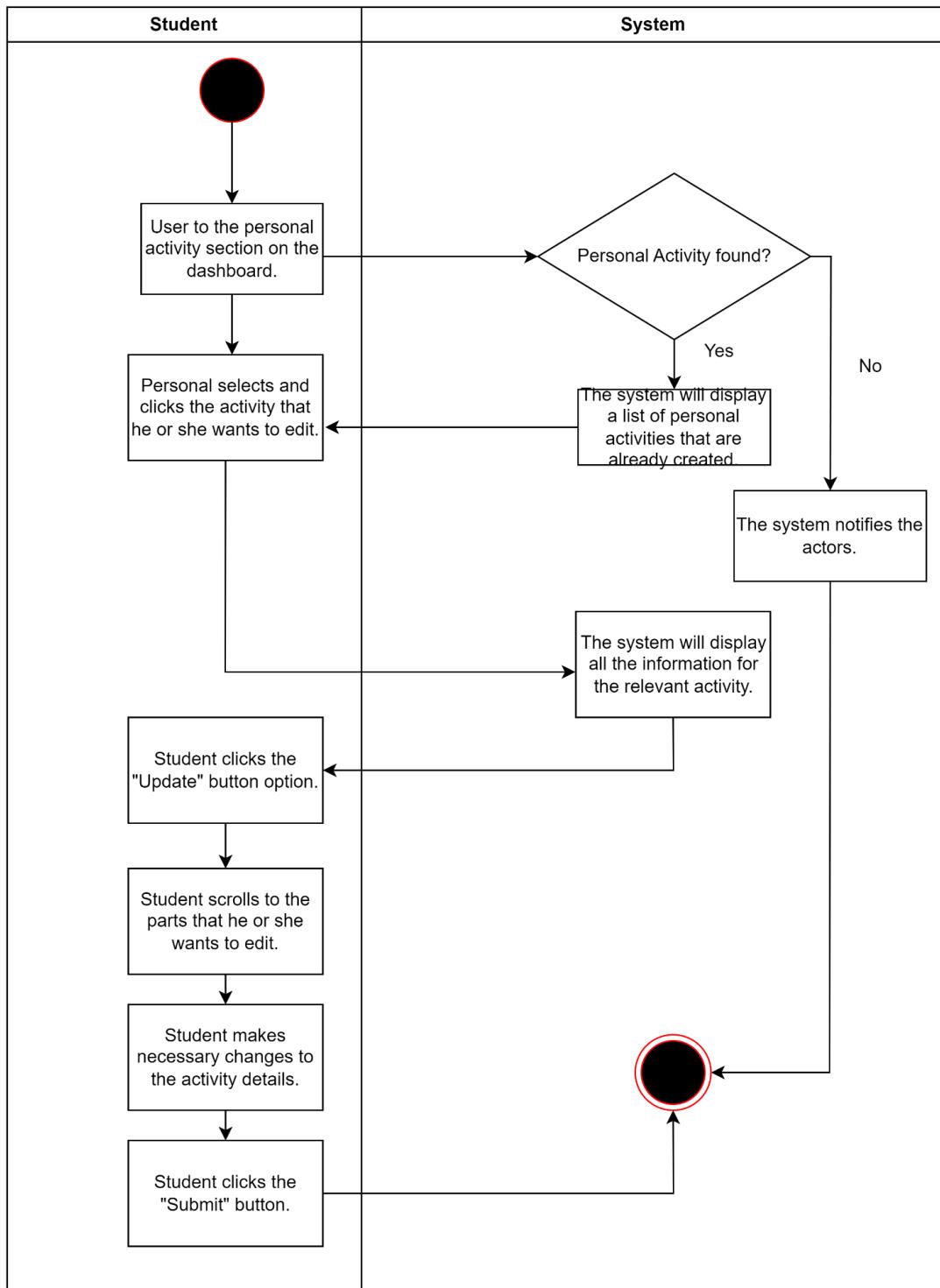
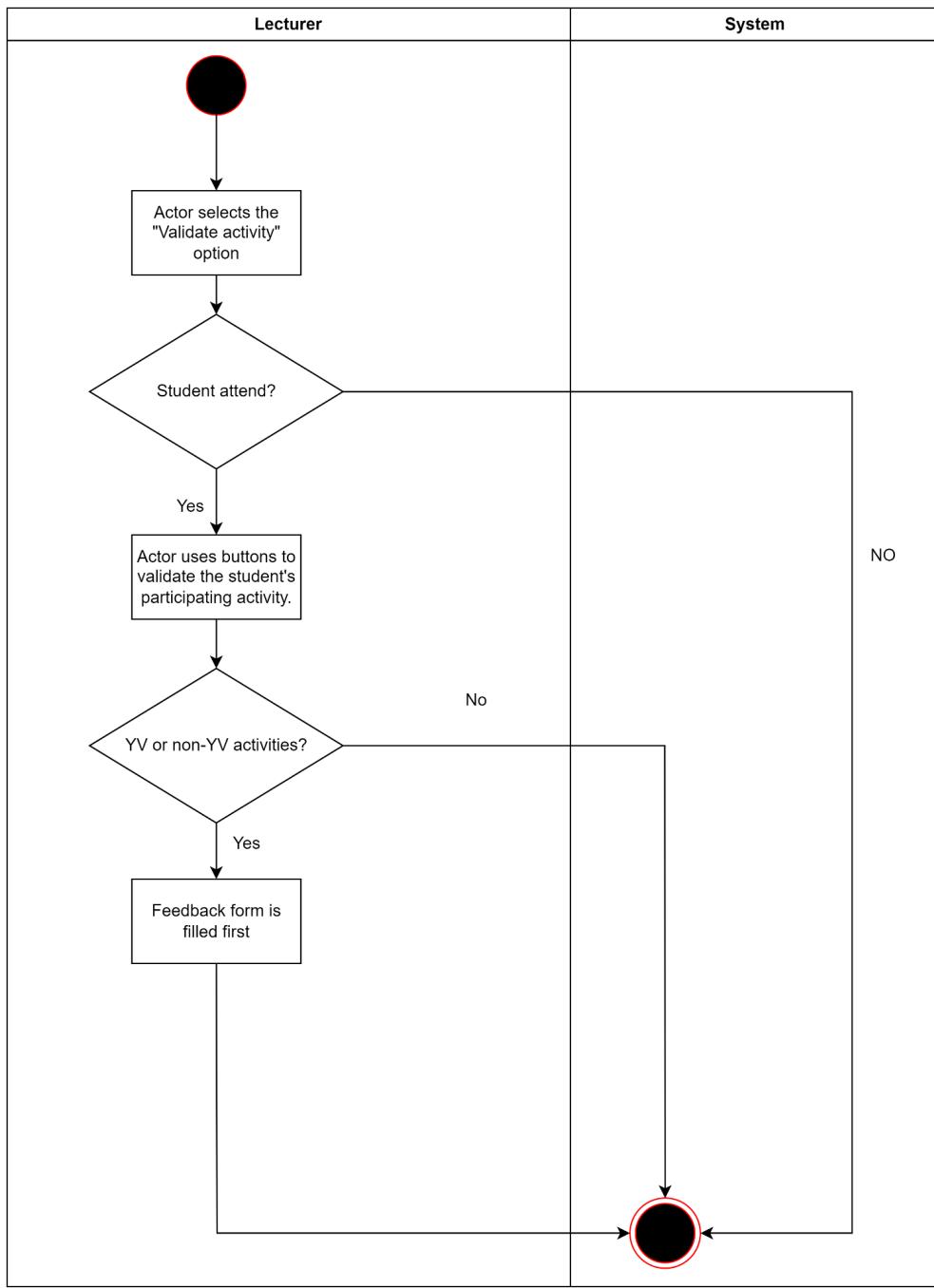


Figure 7.2.9.2: Activity Diagram of Edit Student Personal Activity

## UC024: Use Case <Validate Student Personal Activity>



*Figure 7.2.9.3: Activity Diagram of Validate Student Personal Activity*

UC025: Use Case <View Student Personal Activity>

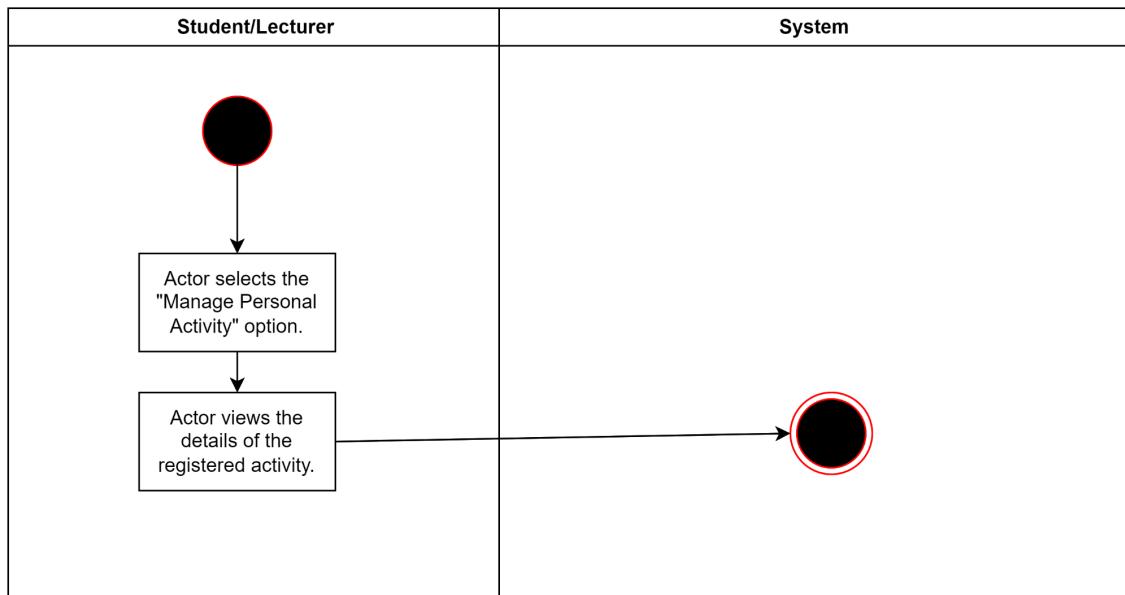


Figure 7.2.9.4: Activity Diagram of View Student Personal Activity

## 7.3 Meeting Log

### Log Book #1



SECP2523 – Database (WBL)

Log Meeting

### MEETING AGENDA

<b>Group Name:</b>	Explorer	<b>Time:</b>	8:00 pm - 11:30 pm
<b>Date of Meeting:</b> (DD/MM/YYYY)	30/1/2023	<b>Location:</b>	Online (Google Meet)

#### 1. Meeting Objective

- To conduct a comprehensive review of the recently developed system.
- To identify areas that have been made updates or improvements
- To ensure that the associated documentation, particularly the system report, is in its latest and most accurate form.

#### 2. Attendees

Name	Department/Division	E-mail	Phone
KOH LI HUI	School of Computing	kohhui@graduate.utm.my	012-768 2618
KOH SU XUAN	School of Computing	koh.xuan@graduate.utm.my	011-1077 8126
LEE YIK HONG	School of Computing	lee.hong@graduate.utm.my	018-378 8589
VINESH A/L VIJAYAKUMAR	School of Computing	vinesh03@graduate.utm.my	012-346 5289

#### 3. Meeting Agenda

Topic	Person in Charge (PIC)	Action
- Review of System Development	Koh Li Hui	- Brief presentation on the key features and functionalities of the developed system.
- Live System Demonstration	Lee Yik Hong	- Conduct a live demonstration of the system, specifically focusing on showcasing the parts that have been updated.
- Detailed Analysis and Discussion	Koh Li Hui	- To conduct a detailed analysis of specific updated features or modules. - To identify which modules or features have

		undergone changes since the last review.
- Review of System Documentation	Koh Su Xuan	<ul style="list-style-type: none"> <li>- To review the existing documentation to identify any sections that mention recent updates.</li> <li>- Present the current state of the system report and associated documentation.</li> </ul>
- Validation and Updating of System Report	All members	<ul style="list-style-type: none"> <li>- Collaboratively validate the system report against the live demonstration and recent updates.</li> <li>- Assign responsibilities for updating the report to reflect the latest changes.</li> </ul>
- Integration Check	Vinesh A/L Vijayakumar	<ul style="list-style-type: none"> <li>- Ensure that the updated components seamlessly integrate with the rest of the system.</li> </ul>

#### 4. Meeting Reflection

The meeting proved to be highly productive, achieving its objectives of discussing and understanding the recent system updates. The collaborative approach not only facilitated effective communication but also led to the identification of specific areas for improvement. The commitment to continuous communication and the discussion on update of system documentation position the team for the successful implementation of the identified updates while maintaining accurate and up-to-date documentation.

Prepared by:



(                   Koh Li Hui                   )

Approved by:

(Signature & Stamp)