Follow App: A Web Application for School Counseling

A Thesis Project Presented to the Faculty of

the School of Computer Studies

University of San Jose - Recoletos

Cebu City, Philippines

In Partial Fulfillment

of the Requirements for the

Degree Bachelor of Science in

Information Technology

by:

Epifanio U. Villaro Jr.

Precious Gift A. Canovas

Rebecca Lander T. Evangelista

December 2022

ENDORSEMENT

This project entitled "**Follow App: A Web Application for School Counseling**" prepared by VILLARO, EPIFANIO U. JR., CANOVAS, PRECIOUS GIFT A., EVANGELISTA, REBECCA LANDER T. in partial fulfillment of the requirements for the degree of BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY MAJOR IN MOBILE DEVELOPMENT, has been endorsed and is recommended for the acceptance and approval for ORAL EXAMINATION.

ENGR. CARMEL M. TEJANA, MManE, MSIT
Adviser

MR. JOHN LEEROY A. GADIANE, MCS
Coordinator

MR. ROY ADRIAN M. SALARES
Member

MR. RODERICK A. BANDALAN, MSIT
Chairperson, CS/IT Department

DR. JOVELYN C. CUIZON
Dean

APPROVAL

Approved by the Tribune for Oral Examination with the grade of **PASSED.**

ENGR. CARMEL M. TEJANA, MManE, MSIT
Adviser

MR. JOHN LEEROY A. GADIANE, MCS
Coordinator

MR. ROY ADRIAN M. SALARES III
Member

MR. RODERICK A. BANDALAN, MSIT
Chairperson, CS/IT Department

DR. JOVELYN C. CUIZON
Dean

ACKNOWLEDGEMENT

ABSTRACT

Counseling is one of the important services that a school should have. It helps students whenever they have problems and also provides services that can also help the mental health of students. However, during the pandemic, face-to-face counseling was somehow prohibited. Because of this, a system was designed to help counselors, teachers, and students connect with each other. Follow App is a web application that will help teachers manually refer students virtually, counselors manage referrals, and automatically refer students based on data of EDP through the institution's School Information Service and Learning Management System (LMS). The system will schedule students for a counseling session and will provide a platform for video meetings. As a result of this endeavor, the researchers ensured that the system was ready for deployment; however, the system can still be improved further.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Rationale of the Study

The Guidance Office plays a vital role in every school. The guidance office comprises guidance counselors in charge of counseling students whenever the latter encounter personal and academic-related problems and conflicts. The function of school counselors in ensuring student achievement is fundamental. (Lapan, Gysbers, & Kayson, 2007; Stone & Dahir, 2006).

Most counseling in schools is done face-to-face through teachers' referrals or individual appointments in which the students need to go to the office to have the counseling session. At the University of San Jose-Recoletos (USJ-R), students are given the appointment slip and are called to the guidance office for their session, and then feedback is given to each teacher. But during the pandemic, the system of counseling has changed. Due to the fact that there were restrictions for physical encounters, the implementation of online classes has been introduced. In lieu of this, counseling session has been shifted to online as well to meet the student's need even in the online set-up.

Follow App is a web application for counseling that will help teachers, counselors, and students have the session in an online set-up. The application focuses on scheduling students for the session, allowing counselors to give feedback online and refer students based on academic performance and behavior.

There are some applications developed that bridged people to different counselors. One of these is Talkspace, an online counseling app for all ages that provides counseling to those who have a mental disorder or those who have personal problems. Another application is the LARKR for on-demand mental health care and TeenCounseling for adolescents who want to explore therapy. These are actually for any users and focus on therapy alone.

Follow App is for school counseling and focuses on student, teacher, and counselor appointments. This study aims to assist and help the institution's existing counseling system online, improving the counseling schedule to be hassle-free and more organized. One of the features of the system is the automatic referral and automatic scheduling based on some criteria, such as grades, absences, and assignment submissions. However, in case the teacher would like to refer students because of their behavior, they can manually refer them online. A schedule for counseling sessions will be automatically created. Feedback is also given to teachers from the counselor, resulting in a paperless transaction. The tools used for this application are Django, Python, HTML, CSS, and Postgresql.

REVIEW OF RELATED LITERATURE

The duties of counselors differ slightly depending on what area of school they engage in. Naturally, the issues students confront will alter and develop as they reach different ages and stages of development. In order to optimize their value to children, school counselors are accountable for reviewing data on their own efficacy and making adjustments to their methods. As a result, the counselor's job is constantly evolving and is influenced by a number of variables at play in the school. A counselor can help develop self-acceptance as well as coping, organizing, and communication skills.

In Doverspike, 2009; Heinlen, Welfel, Richmond, & Rak, 2003, cyber counseling, also called electronic therapy, e-therapy, Internet counseling, Internet psychotherapy, online counseling, computer-mediated counseling, services like telehealth and web counseling are an increasing topic of ethical discussion in the industry. In Mallen & Vogel, 2005, any therapy, counseling, and psychoeducation delivery by a qualified professional to a client in a setting without face-to-face interaction using distance communication technologies such the phone, asynchronous email, synchronous chat, and videoconferencing. While this definition may seem broad, it attempts to cover many of the emerging delivery systems being used by counselors. According to Doverspike, 2009, a counselor can be labelled as cybercounselor if he or she hosts either an information-based or an interactive-based website; receives or exchanges e-mails with clients, even to schedule appointments; provides any online service, such as testing, counselling, or assessments; consults with colleagues via electronic medium; or transmits records to colleagues or insurance companies electronically.

School counselors working in an online setting provide a school counseling program through today's technology (virtual/online/e-learning), counseling with the same standards and adherence to ethics as school counselors working in traditional school settings. This statement was in response to overwhelming enrolment throughout the United States in online or virtual educational environments (Setzer & Lewis, 2005).

In 2017, there were 528 virtual schools established throughout the United States, with 278,500 students in 34 states including Georgia (Molnar et al., 2017). As virtual counseling is still evolving, Steele (2017) reported offering synchronous services such as mini-courses, group counseling, individual counseling, and workshops as part of an online counseling program. These services were rendered via various technological platforms such as Zoom, Adobe Connect, Blue Jeans, Skype, and Go to Meeting. Asynchronous services were also provided, such as websites, articles, discussion forums, webinars, emails, and recorded presentations. Steele et al. (2017) surveyed the usage of online technology within school counseling programs.

The study focuses on the use of online counseling in universities, particularly after the pandemic there made it clear how beneficial it is because it makes it easier for counselors to connect with and monitor their students' academic and behavioral needs. The researchers developed the Follow App, where the idea is to provide online counseling for students so they may access counseling conveniently and avoid going to the counselor for face-to-face counseling. This will benefit not only the counselors and students but also the teachers as they have the closest relationships with the students.

REVIEW OF RELATED WORKS

BetterHelp is a mental health platform that offers customers immediate access to online mental health treatments, and it is one of the largest therapy platforms in the world. Users can message a licensed therapist anytime since it provides online counseling and therapy assistance by phone, video session, live chat, or message exchange. BetterHelp provides accessible and reasonable care, and it alters how people approach their mental health and assists them in overcoming life's obstacles. According to BetterHelp, the cost of therapy ranges from $60 to $90 per week (billed every four weeks), based on your location, preferences, and therapist availability.4 weeks) and it is based on your location, preferences, and therapist availability.

Another existing platform is Talkspace, a New York City-based online and mobile therapeutic service. Oren and Roni Frank founded it in 2012 to make mental health services more accessible for everyone. According to Talkspace, the service has helped users manage several mental health conditions, including anxiety, addiction, and depression. It was among the first businesses to provide online therapy, and users can live chat through messaging or schedule a video chat with the therapist. Talkspace is in tune with any devices iOs, iPhone and Android. Users can communicate with their therapist via text-based services at any time, from anywhere, and get a response as soon as their therapist has time to do so. Through their preferred web browser or the Talkspace mobile app, users can enter Talkspace rooms where a safe virtual therapy room is generated after the user selects a therapist.

Teen Counseling connects with a licensed therapist who specializes in assisting young people ages 13-19 who can seek internet therapy from a professional therapist.

Wherever you are, whenever you need, just text your therapist, or arrange a phone or video call. The platform provides the parent and the teen with two distinct session settings. Unless there is a major risk, therapy is always private between the user and the therapist. Through a computer, tablet, or smartphone, the platform offers professional treatment that is discreet, affordable, and easily accessible.

Every teen can gain from having a qualified therapist nearby to talk about problems like coping mechanisms, bullying, rage, self-esteem issues, sadness, anxiety, stress, eating disorders, or any other mental health issues. In accordance with Teen Counseling, the therapy is done in four different ways; message-exchanging, live conversation, phone call, and connecting with the therapist via video chat.

PROJECT OBJECTIVE

This study aims to help counselors conduct their online counseling to the students with the following objectives:

General Objectives:

1. Give counselor easier and hassle free management of referrals
2. Improve and automate referral system of SDPC
3. Connect students and teachers to SDPC virtually
4. Create platform where student and counselor can meet virtually

Specific Objectives:

1. Implement Automatic Scheduling System
2. Implement Online Referral System
    2.1. Implement Manual Referral of Student
    2.2 Implement Automatic Referral of Student

3.1. Implement Online Appointment System

3.2 Implement Online Giving of Feedback

4. Implement a Tele-Counseling System
    4.1 Integrate Video Meeting Feature

Project Scope and Limitation

The study focuses on the counseling system of SDPC at the University of San Jose-Recoletos. A student will be referred manually based on behavior and referred automatically based on academic performance. The system will recommend a schedule for a counseling session to be accepted by the student and counselor. This study covers only the college students of the University of San Jose-Recoletos, and only focuses on the counseling service of the guidance office or the Student's Development and Placement Center (SDPC) office of the University of San Jose-Recoletos. The users of the system are the counselor, student, teacher, director, and administrator. The student must be enrolled first in order to use the application.

The study's limitations are as follows: The application's platform is web. It is only developed for web usage.

CHAPTER II

SOFTWARE REQUIREMENTS AND DESIGN SPECIFICATION

This chapter specifies the user and system requirements that are expected to be accomplished and the structure and process of achieving these. It contains sections for Use Case Diagram, Use Case Narrative, Activity Diagram, Class Diagram, and User Interface Design.

Application Overview

This system provides online referral system and automatic scheduling system allowing faster and smooth referral and counseling transaction that can help students, teachers, and most especially the counselor.

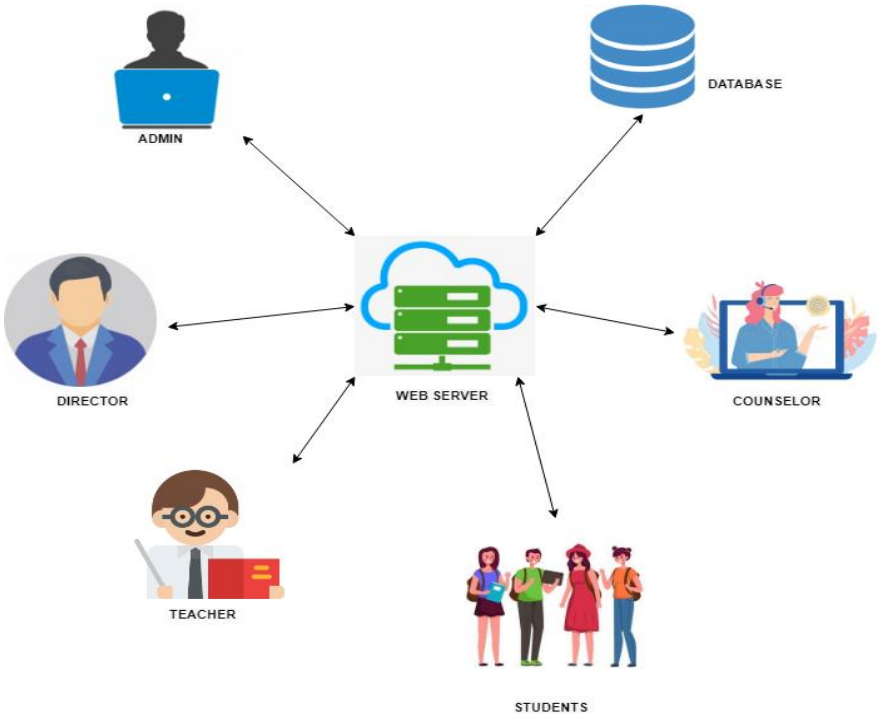

Figure 1: Architectural Diagram

Figure 1 explains the structure of “Follow App” Application. The structure consists of clients, a web server, and a database. The application has five clients, namely the administrator, director, teacher, counselor, and student. The web server handles the client’s requests. A web server is also responsible for the communication and connection of the clients. The database contains and stores the data needed by the clients for their system.

Use Case Diagram

The use case diagram shows the relationship between actors and different use cases of the system in which an actor is involved.

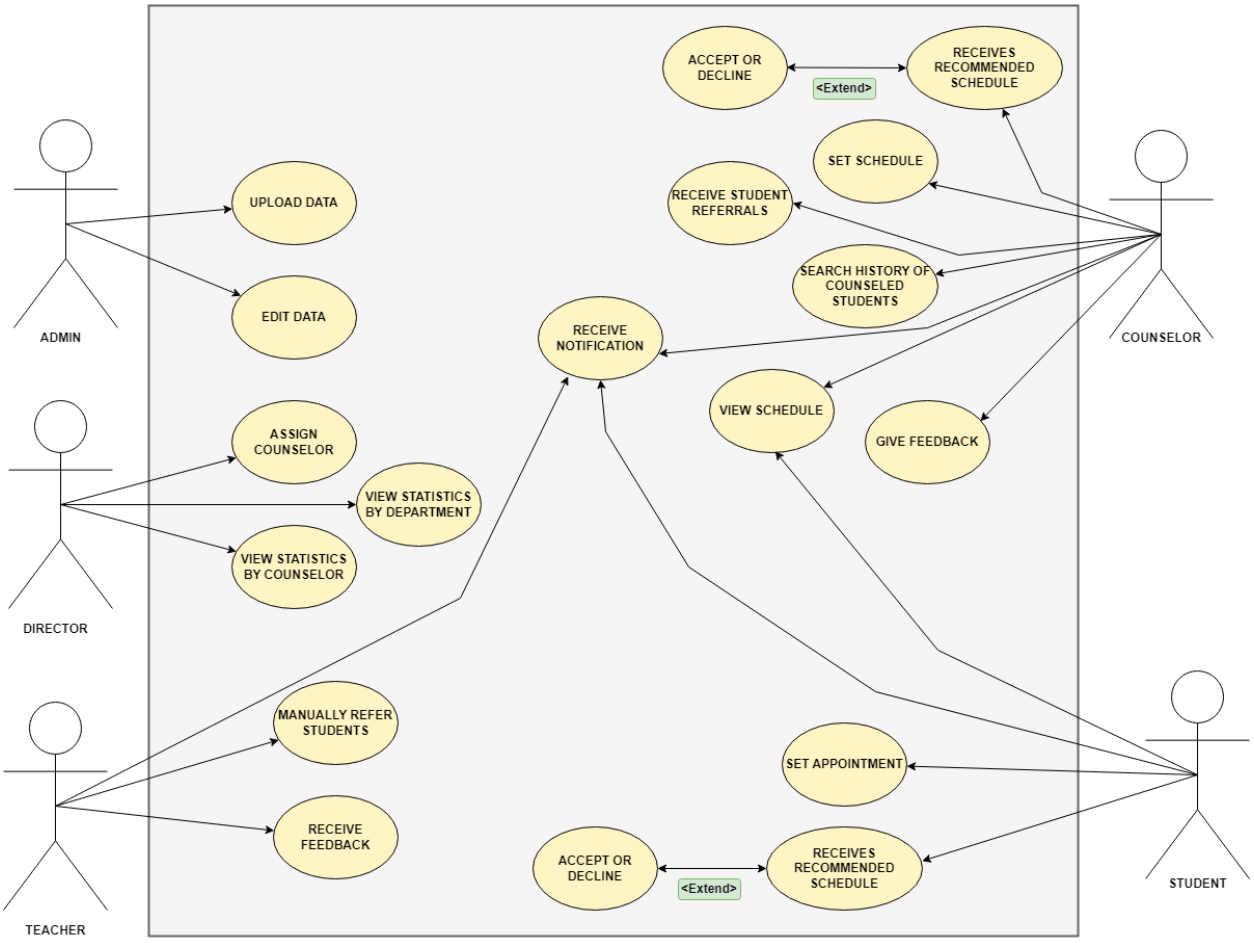

Figure 2: Use Case Diagram

Figure 2 shows the functionalities in the application. The administrator or admin can upload data, and can edit, and view the uploaded data. The director can assign counselors to different degree programs and can view statistics of referrals. The teacher can manually refer a student and can receive feedback from the counselor about the referred student. The counselor can accept or decline recommended schedule for counseling session, can set schedule when not available, can give feedback after counseling session and can view schedule of class and counseling. The student can set an appointment for a counseling session, can accept and decline recommended schedule for counseling, and can view schedule for class and counseling session.

USE CASE NARRATIVES

The Use Case Narratives of Follow App shows the flow of how the users interact with the web application to achieve a certain goal. The narrative provides different scenarios for each use case and the response of the system.

Use Case: UC001 User Signup

The purpose of this narrative is to show the flow of how the user inputs his information in creating account. The system will store the information of the user on the database.

Name:	**User Signup**
Description:	**User creates account**
Actors:	User
Type:	Essential
Precondition:	The user is required to sign up
Post condition:	The user will have an account
FLOW OF EVENTS	
Actor Action	**System Response**
1. The user clicks the sign - up button	2. The system displays the form to be filled in by the user
3. The user enters the ID number and mobile number for verification purposes	4. The system sends the code to the mobile number of the user
5. The user enters the code received	6. The system verifies the code. After the verification, the system displays the sign-up form for the ID number and password
7. The user enters the ID number and password	8. The system creates the account of the user

Table 1: Use case narrative for User Sign Up

Use Case: UC002 Edit Profile

The purpose of this narrative is to show the flow of how the student modifies his/her information.

Name:	**Edit Profile**
Description:	Student can edit his/her profile information
Actors:	Student
Type:	Essential
Precondition:	The student edits his/her information
Post condition:	Student will have an updated information
FLOW OF EVENTS	
Actor Action	**System Response**
1. The student clicks on the edit profile tab	2. The system displays the student's information
3. The student edits some of his/her information and click submit button	4. The system displays the updated information of the user and saves the updated data in the database

Table 2: Edit Profile Use Case Narrative

Use Case: UC003 Sets Appointment

The purpose of this narrative is to show the flow of how the student sets appointments for the counselling session.

Name:	**Sets Appointment**
Description:	Student sets an appointment
Actors:	Student
Type:	Essential
Precondition:	The student has logged in and has no existing appointment
Post condition:	Student will have an appointment

FLOW OF EVENTS	
Actor Action	**System Response**
1. The student clicks the set schedule tab	2. The system displays a form page
3. The student enters the reason for appointment then click submit	4. The system sets schedule for counseling session

Table 3: Sets Appointments of the Student Use Case Narrative

Use Case: UC004 Receives Notifications

The purpose of this narrative is to show the flow of how the student receives notifications.

Name:	**Receives Notification**
Description:	Student receives notification
Actors:	Student
Type:	Essential
Precondition:	Student must set an appointment
Post condition:	Appointment already scheduled
FLOW OF EVENTS	
Actor Action	**System Response**
1. The student clicks on the notifications tab	2. The system displays the list of notifications
3. The student clicks specific notification	4. The system displays the details of the notifications

Table 4: Receives Notifications of the Student Use Case Narrative

Use Case: UC005 List of Pending Referrals

The purpose of this narrative is to show the flow of how the counselor views the list of referrals.

Name:	**List of Pending Referrals**
Description:	Counselor views list of pending referrals
Actors:	Counselor
Type:	Essential
Precondition:	The counselor clicks the Referral tab
Post condition:	The counselor will be able to view the list of pending referrals
FLOW OF EVENTS	
Actor Action	**System Response**
1. The counselor clicks the referral tab	2. The system displays a dropdown of list of referrals and list of pending referrals
3. The counselor clicks the List of Pending Referrals	4.The system displays the list of pending referrals
4. The counselor clicks the view information in the right part of the list	5. The system displays the details of the referral

Table 5: List of Referrals Use Case Narratives

Use Case: UC006 Gives Feedback

The purpose of this narrative is to show the flow of how the counselor gives feedbacks to students after the counseling session. The system will store the feedbacks on the database.

Name:	**Gives Feedback**
Description:	**Counselor provides a feedback information**
Actors:	Counselor
Type:	Essential

Precondition:	The counseling session must be done
Post condition:	Feedbacks will be given
FLOW OF EVENTS	
Actor Action	**System Response**
1. The counselor clicks the end call in the meeting	2. The system display the form for the feedback
3. The counselor inputs the feedback and clicks submit	4. The system saves the feedback on the database and sends it to the teacher referring

Table 6: Use case narrative for Gives Feedbacks of the Counselor

Use Case: UC007 Sets Schedule

The purpose of this narrative is to show the flow of how the counselor sets a schedule.

Name:	**Sets Schedule**
Description:	**Counselor sets a schedule**
Actors:	Counselor
Type:	Essential
Precondition:	Counselor sets schedule if he is not available
Post condition:	Counselor sets a new schedule
FLOW OF EVENTS	
Actor Action	System Response
1. The counselor clicks on the set schedule tab	2. The system displays a form to set a schedule
3. The counselor chooses the date and time then submit button	4. The system saves the data

Table 7: Sets Schedule on the Counselor Use Case Narrative

Use Case: UC008 Manually Refer

The purpose of this narrative is to show how teacher manually refers students.

Name:	**Manually Refer**
Description:	Teacher can manually refer the student
Actors:	Teacher
Type:	Essential
Precondition:	The teacher must have subject load
Post condition:	Students will be manually referred
FLOW OF EVENTS	
Actor Action	**System Response**
1. The teacher clicks on the subject where the student is enrolled	2. The system displays the list of students who are enrolled on the subject chosen
3. The teacher clicks on the refer button on the student chosen	4. The system displays the form where the teacher enters the reasons of referral
5. The teacher clicks submit	6. The system sets a counselling session and notify the counselor and student

Table 8: Manually Refers for Teacher Use Case Narrative

Use Case: UC009 Receives Feedback

The purpose of this narrative is to show how teacher receives feedback.

Name:	**Receives Feedback**	
Description	Teacher receives feedback	
Actors:	Teacher	
Type:	Essential	
Precondition:	There must be feedback given by the counselor	
Post condition:	None	
FLOW OF EVENTS		
Actor Action	**System Response**	

1. The teacher clicks on the notifications tab	2. The system displays the list of notifications	
3. The teacher clicks on specific notification	4. The system displays the details of the student's referred information and the feedback given	

Table 9: Receives Feedback for Teacher Use Case Narrative

Use Case: UC0010 View Referred Students

The purpose of this narrative is to show how teacher views referred students.

Name:	**View Referred Students**
Description	Teacher views the referred students
Actors:	Teacher
Type:	Essential
Precondition:	There must be referred students
Post condition:	None
FLOW OF EVENTS	
Actor Action	**System Response**
1. The teacher clicks on the view referred student tab	2. The system displays the list of referred students
3. The teacher clicks the view information	4. The system displays the details of that specific student

Table 10: View Referred Students Use Case Narrative

Use Case: UC0011 Assign Counselor

The purpose of this narrative is to show the flow of how the director assigns a specific counselor to its designated department.

Name:	**Assign Counselor**
Description:	**Directors assigns a specific counselor**
Actors:	Director

Type:	Essential
Precondition:	The director is required to assign a counselor
Post condition:	The director will assign a counselor
FLOW OF EVENTS	
Actor Action	**System Response**
1. The director clicks the view degree program tab	2. The system display all the list of degree program
3. The director choose a specific degree program and clicks the assign counselor button	4. The system displays a bit of information of degree program and a dropdown list of counselors
5. The director chooses a counselor from the dropdown list and clicks the submit button	6. The system assigns the counselor on that designated degree program

Table 11: Assign Counselor Use Case Narrative

Use Case: UC0012 Upload Data

The purpose of this narrative is to show the flow of how the Admin uploads a data.

Name:	**Upload Data**
Description:	Admin uploads a specific data
Actors:	Admin
Type:	Essential
Precondition:	The admin logs on to the system
Post condition:	The admin will upload the data and displays the list of data
FLOW OF EVENTS	
Actor Action	**System Response**
1. The admin clicks the semester in the dropdown list and choose a file button.	2. The system displays the screen that shows the data is being upload

Table 12: Upload Data on the Admin Use Case Narrative

Use Case: UC0013 View Data

The purpose of this narrative is to show the flow of how the Admin views a data.

Name:	**View Data**
Description:	**Admins views a specific data**
Actors:	Admin
Type:	Essential
Precondition:	The admin must have an upload data
Post condition:	The admin will choose a specific data to view.
FLOW OF EVENTS	
Actor Action	**System Response**
1. The admin clicks the tab view and choose a data to show	2. The system displays the data
Alternate Flow of Events:	
A1: Edit Data	
Actor Action	**System Response**
1. The admin clicks the edit button	2. The system displays the data where it can be editable
A2: Search Data	
Actor Action	**System Response**
1. The admin inputs a data in the search box	2. The system displays the specific data
A3: Filter Data	
Actor Action	**System Response**
1. The admin clicks the dropdown list and choose a data to be filtered	2. The system displays the data's that has been filtered

Table 13: Upload Data on the Admin Use Case Narrative

Use Case: UC0014 Automatic Referral

The purpose of this narrative is to show the flow of how the student is referred automatically

Name:	**Automatic Referral**
Description:	Admin uploads data from LMS and EDP
Actors:	Admin
Type:	Essential
Precondition:	The admin is required to upload the data.
Post condition:	The system will schedule the students for referral
FLOW OF EVENTS	
Actor Action	**System Response**
1. The admin upload EDP and LMS data	2. The system saves the data
3. The admin click the referral button	4. The system schedules the student for their counseling and notify both student and counselor

Table 14: Automatic Referral of Admin Use Case Narrative

ACTIVITY DIAGRAM

The activity diagrams here present the series of actions and flows of the most prominent features, describing how they interact step-by-step.

User: Signup Page

This activity diagram displays how the user creates an account by providing personal information and other needed information.

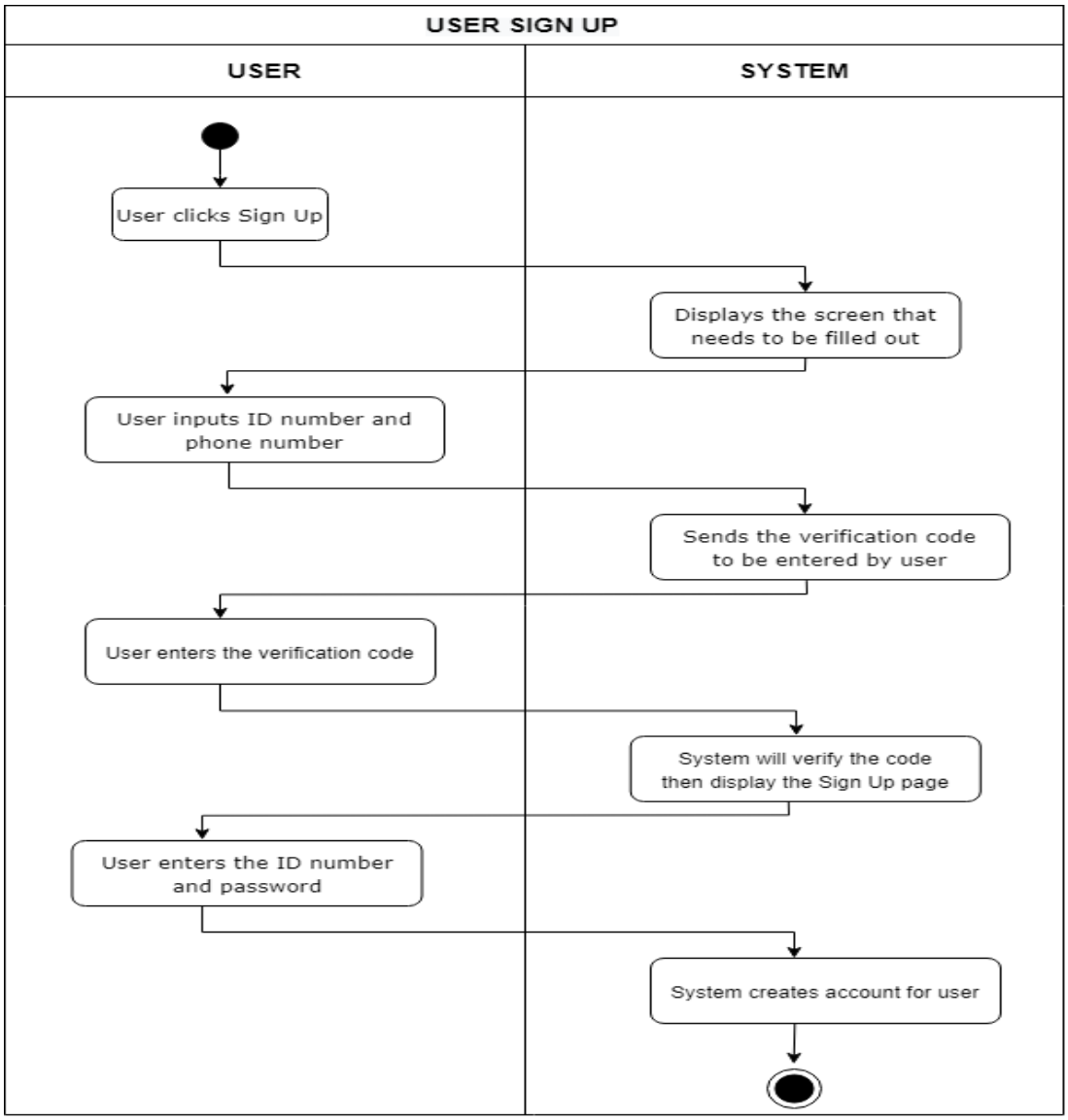

Figure 3: Sign Up Page Activity Diagram

Figure 3 shows the account creation diagram. The user will click sign up and input the ID and phone numbers. After clicking the submit button, the user will receive the verification code. Then the user will input the code, and the system will verify whether it is valid. If it is, the user will again input the ID number together with the password, and the system will create the account and save the data in the database.

User: Login Page

This activity diagram displays how the user logins his/her account.

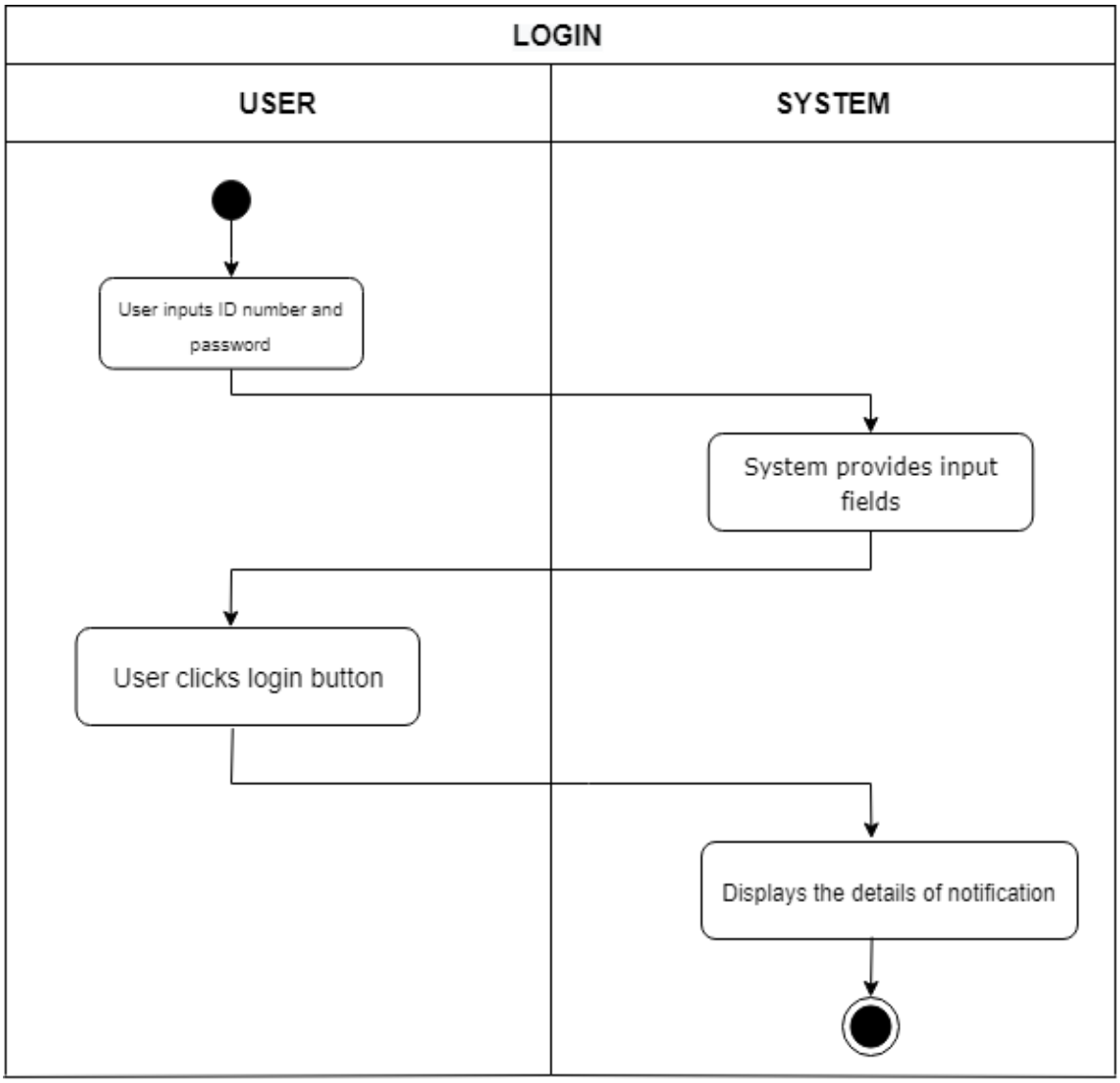

Figure 4: User Login Activity Diagram

Figure 4 shows the login process. The user will input the ID number and password. Then the system will verify whether it is valid. If it is, the user will navigate to its specified homepage.

Admin: Upload Data

This activity diagram displays how the admin uploads a data.

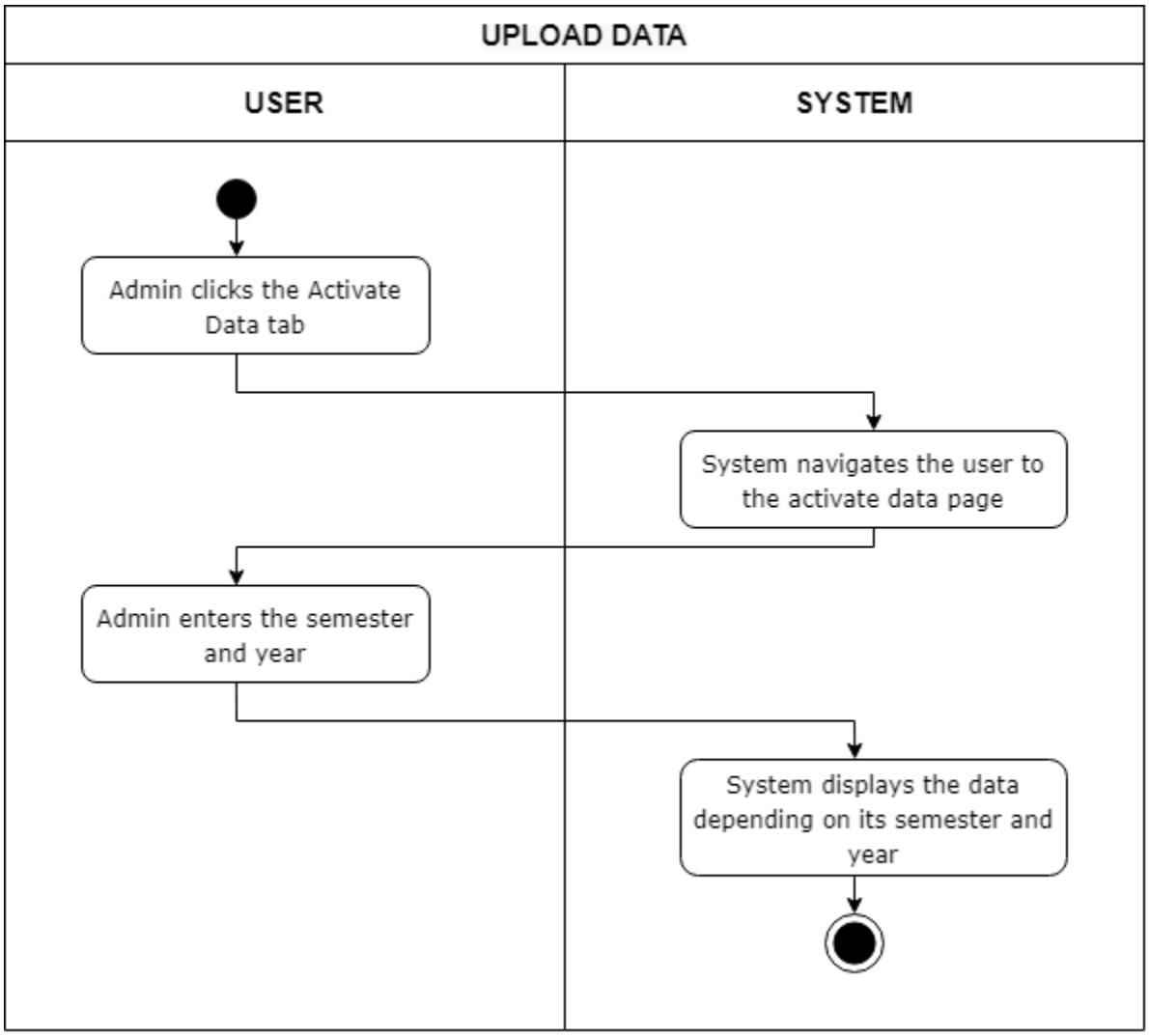

Figure 5: Upload Data Activity Diagram

Figure 5 shows how the administrator populates the database. The admin will click the upload data in the navbar and choose what data to upload. The choices are subject offerings, a list of faculties and students, and the student load. If the admin clicks one of the choices, the upload UI will display, and the admin will choose what file to upload from the computer. After clicking the upload button, the system will save the data in the database and display the data that is being uploaded.

Admin: View/Edit Data

This activity diagram displays how the admin views, edit and search a specific data.

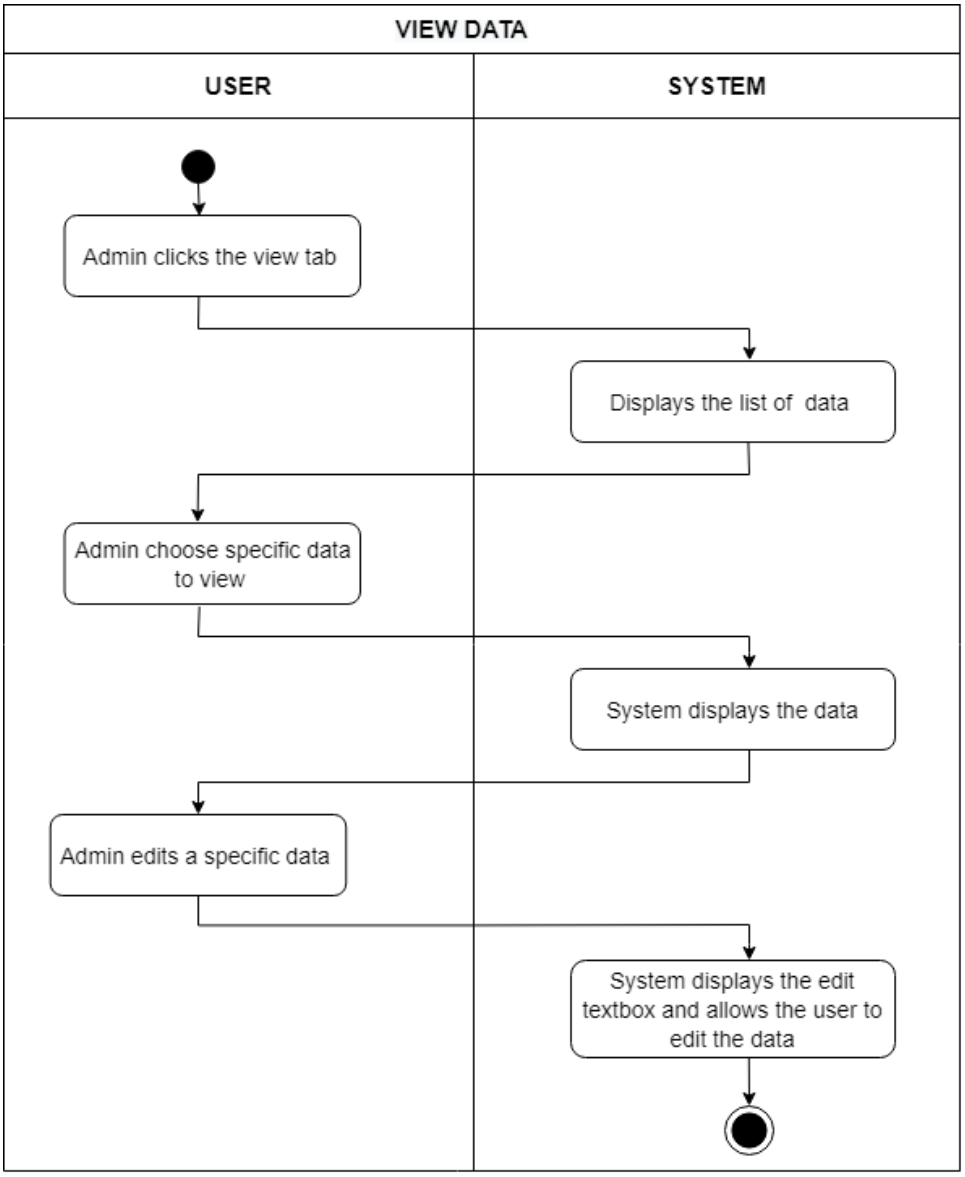

Figure 6: View Data Activity Diagram

Figure 6 shows how the administrator can view the data from the uploaded CSV file. The admin will click view data in the navbar. The admin can choose what data to view. After clicking the specific data, the system will display its information. The admin can also edit some data if it has an edit button. If the admin clicks the edit button, the textbox will display for the admin to input the updated data.

Admin: Activate Data

This activity diagram displays how the admin activates the data depending on the semester and year.

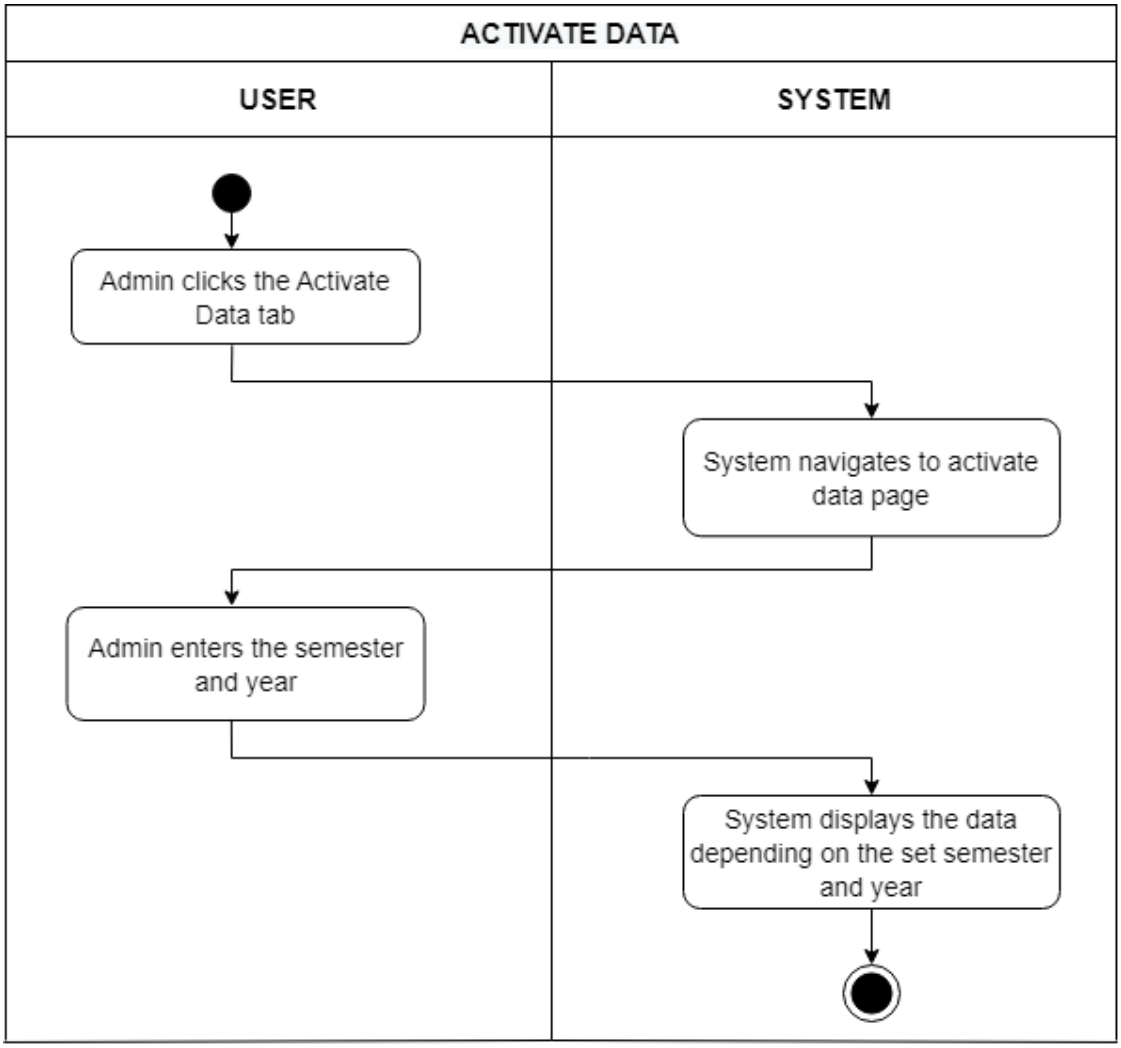

Figure 7: Activate Data Activity Diagram

Figure 7 shows how the administrator can set the activated year and semester. The admin clicks the activate data in the navbar. After clicking the tab, the system will display the UI for the admin to edit what semester and year will be activated, then click submit. After that, all data in the system will be based on the activated semester and year.

Admin: Automatic Referral

This activity diagram displays how the admin uploads the data from the LMS and EDP, and how the student is automatically referred.

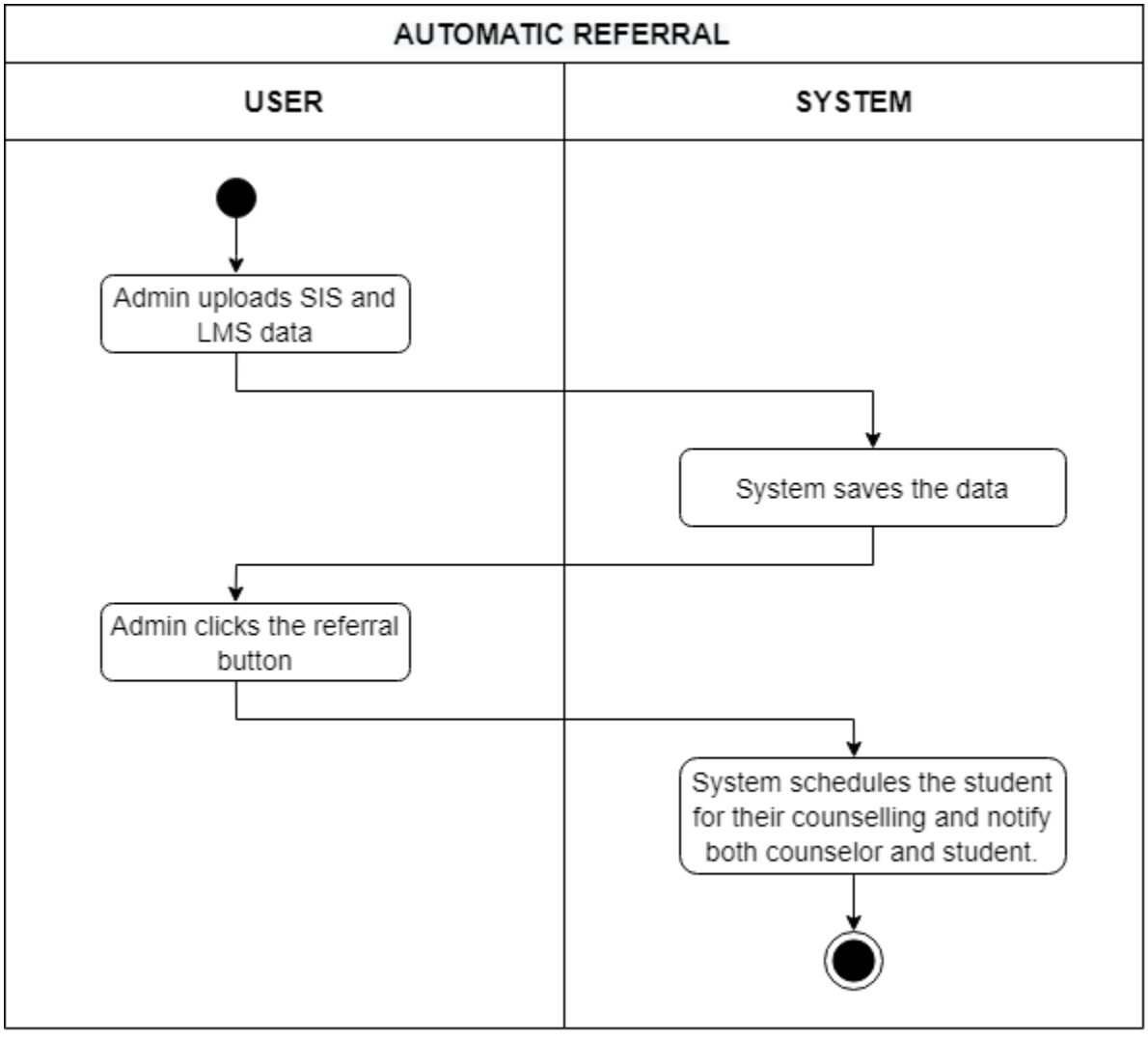

Figure 8: Automatic Referral of Admin

Figure 8 shows how the automation referral works. The admin will upload the CSV file from the SIS and LMS. After uploading, the system will save the data in the database. Then the refer button will be enabled, and if the user clicks it, the system will check those students who need to be referred based on the uploaded CSV file and will be given a recommended schedule for their counseling session and notify both counselor and student.

Director: Assigns Counselor per Degree Program

This activity diagram displays how the director assigns a specific counselor.

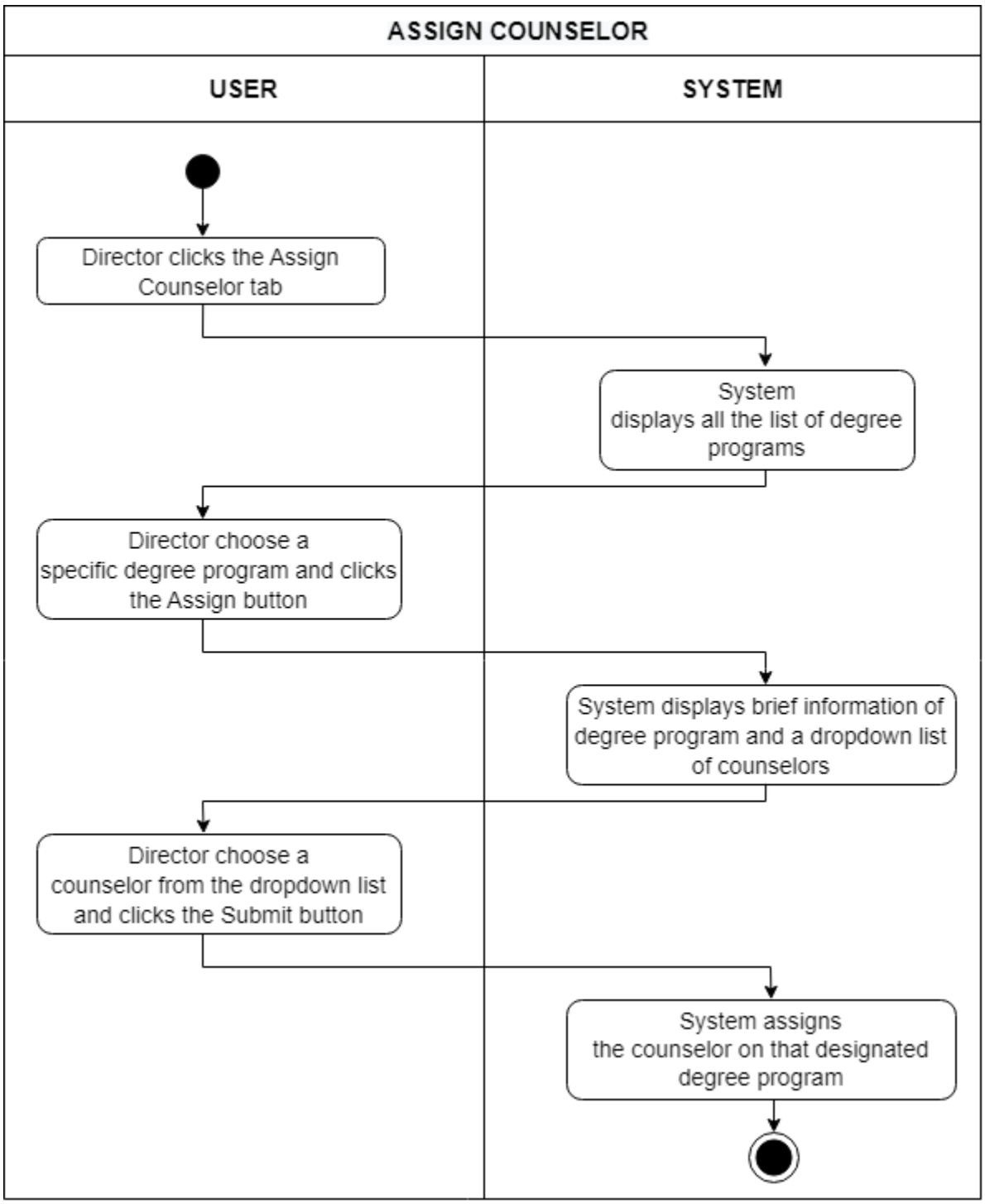

Figure 9: Assigns Counselor Activity Diagram

Figure 9 shows how the assigning of a counselor per degree program. The director will click the assign counselor tab, and the system will display the list of degree programs. The director will click the assign button and navigate to another page where a list of counselors is in the drop-down list. The director will select the counselor and click the assign button. The system will then save it in the database.

Director: View Statistics per Degree Program

This activity diagram displays how the director views the statistics per degree program.

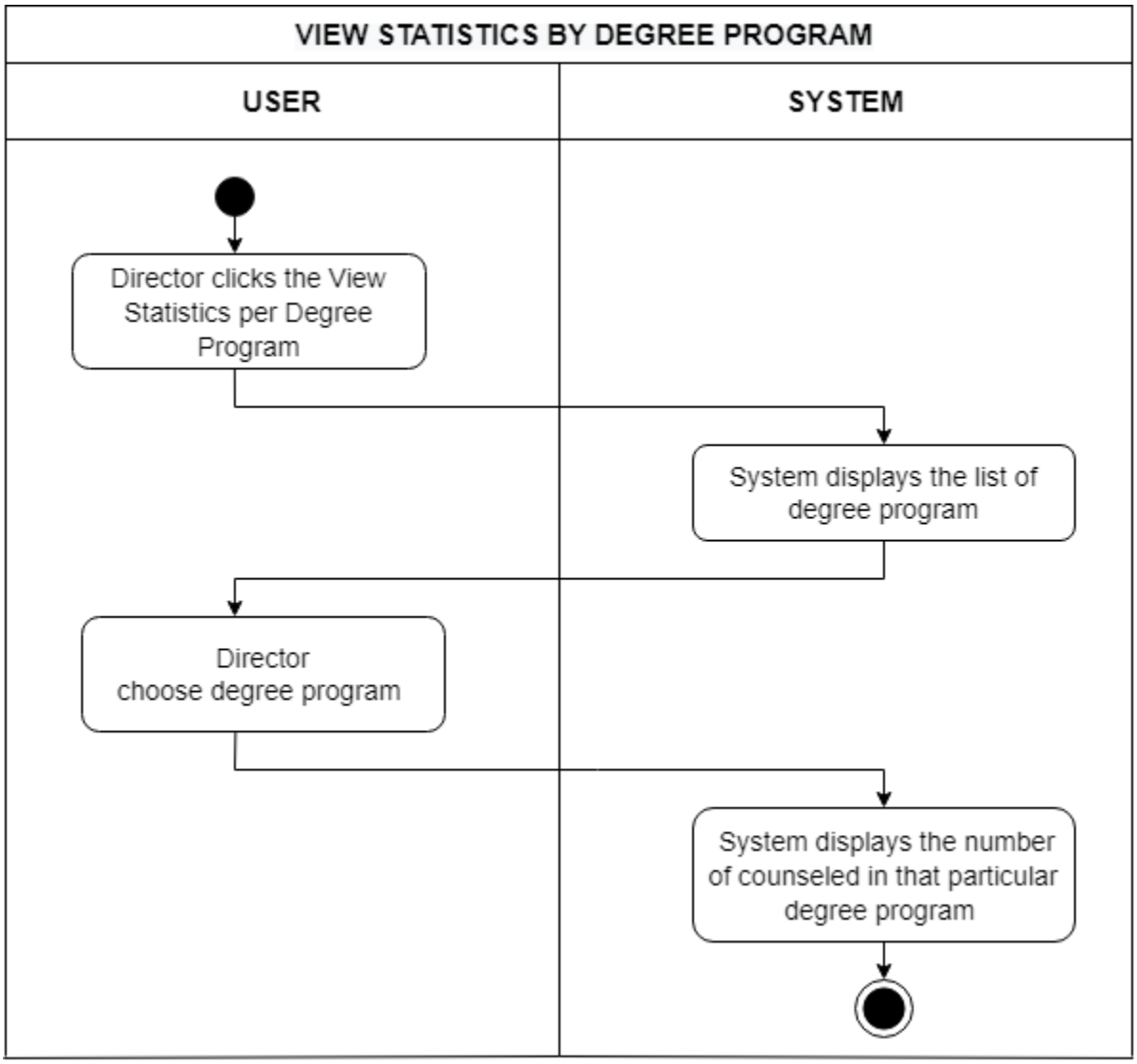

Figure 10: View Statistics per Degree Program Activity Diagram

Figure 10 shows the viewing statistics per degree program. The director will click the view statistic per degree program tab. The system will then display the list of the degree program. If the director clicks a specific degree program, the system will display the number of counseled in that particular degree program.

Director: View Statistics by Counselor

This activity diagram displays how the director views the statistics by Counselor.

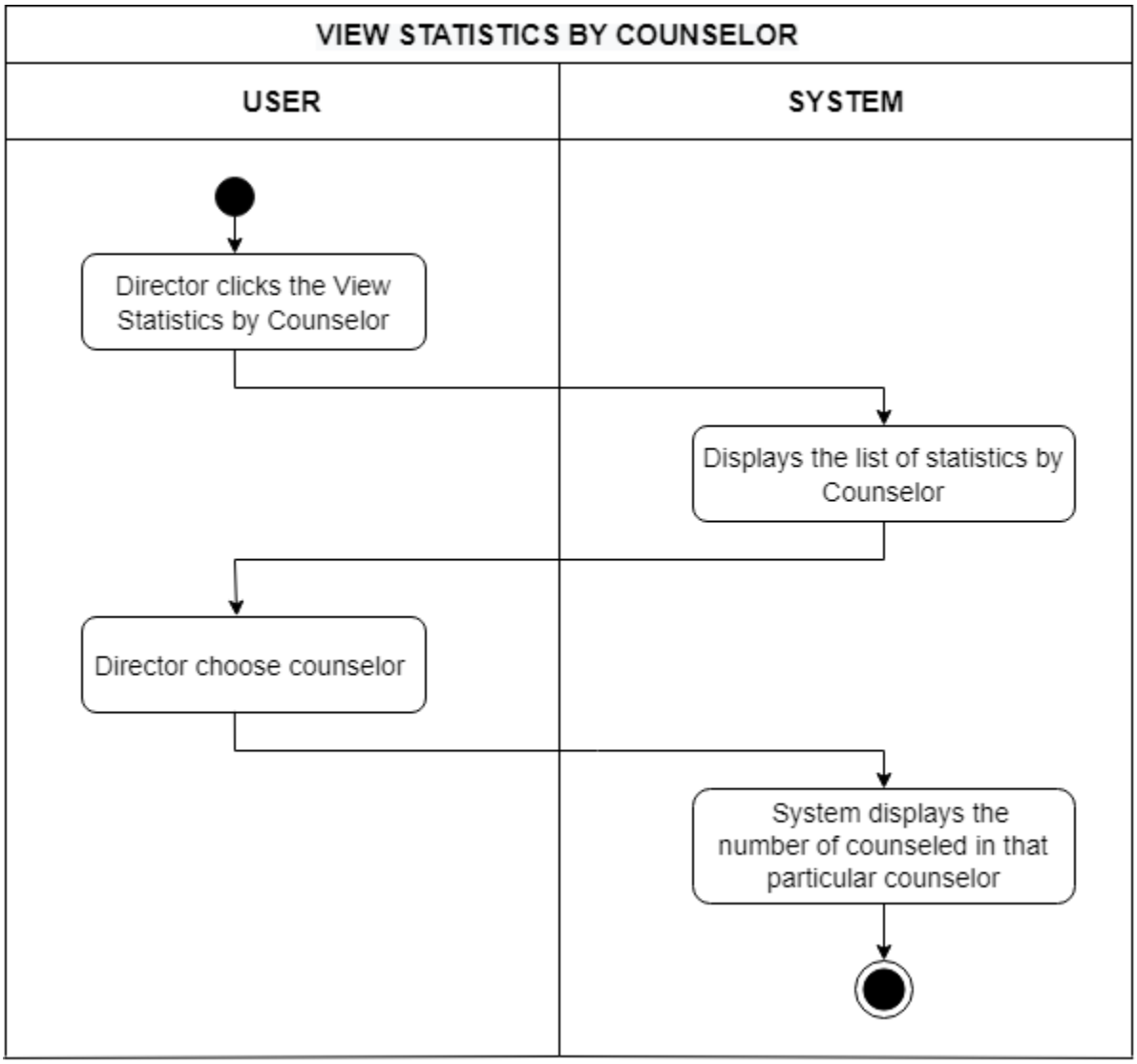

Figure 11: View Statistics by Counselor Activity Diagram

Figure 11 shows the viewing statistics per counselor. The director will click the view statistic per counselor tab. The system will then display the list of counselors. If the director clicks a specific counselor, the system will display the number of counseled in that particular counselor.

Counselor: View List of Referred Students

This activity diagram displays how the counselor views the list of referred students.

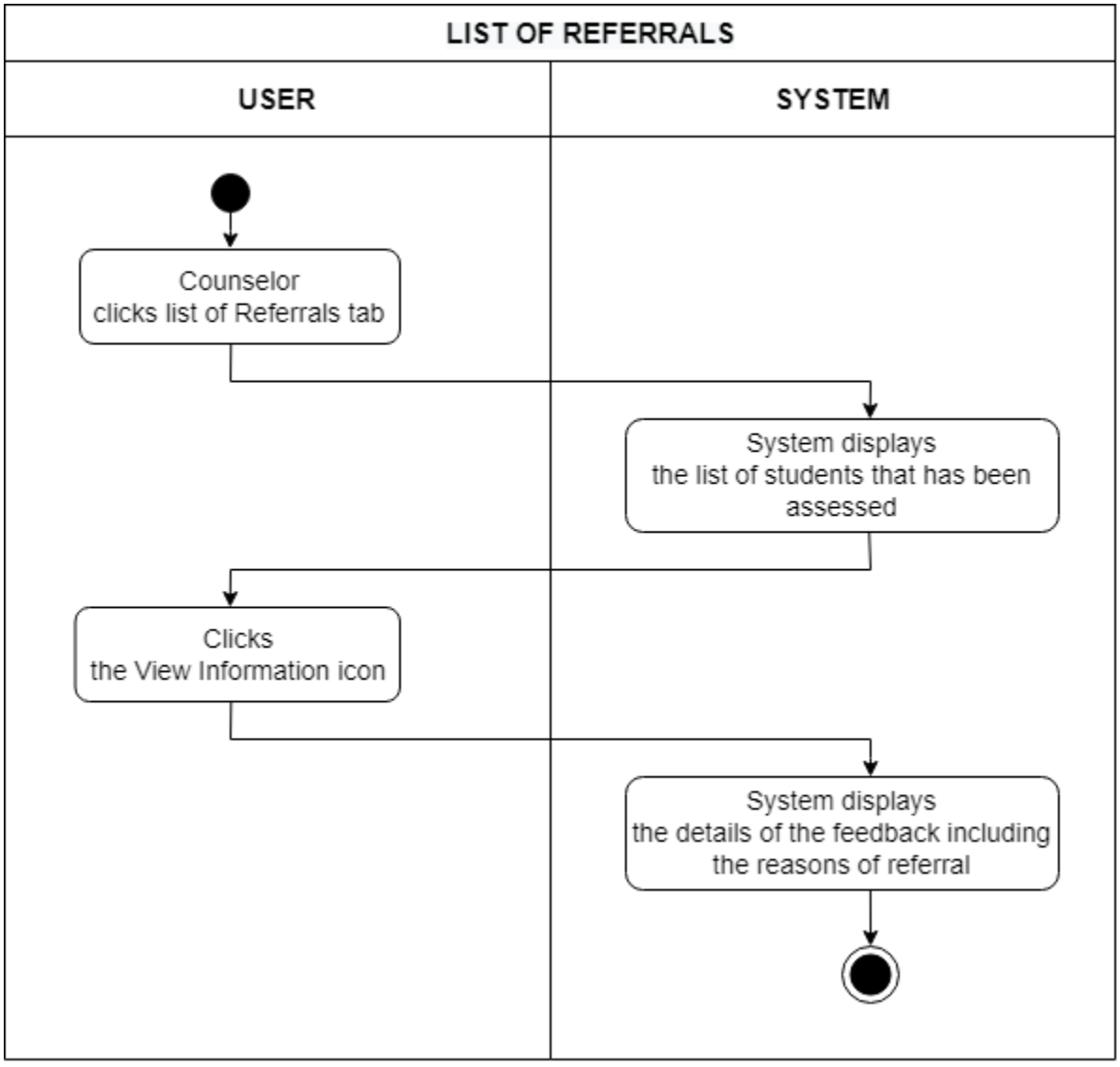

Figure 12: View List of Referred Students Activity Diagram

Figure 12 shows the viewing list of referred students. The counselor will click the list of referred Student tab. The system will then display the list of students that are already counseled. The counselor clicks a specific information icon then the system will display the information of that particular student together with the feedback.

Counselor: View List of Pending Referrals

This activity diagram displays how the counselor views the list of pending referrals.

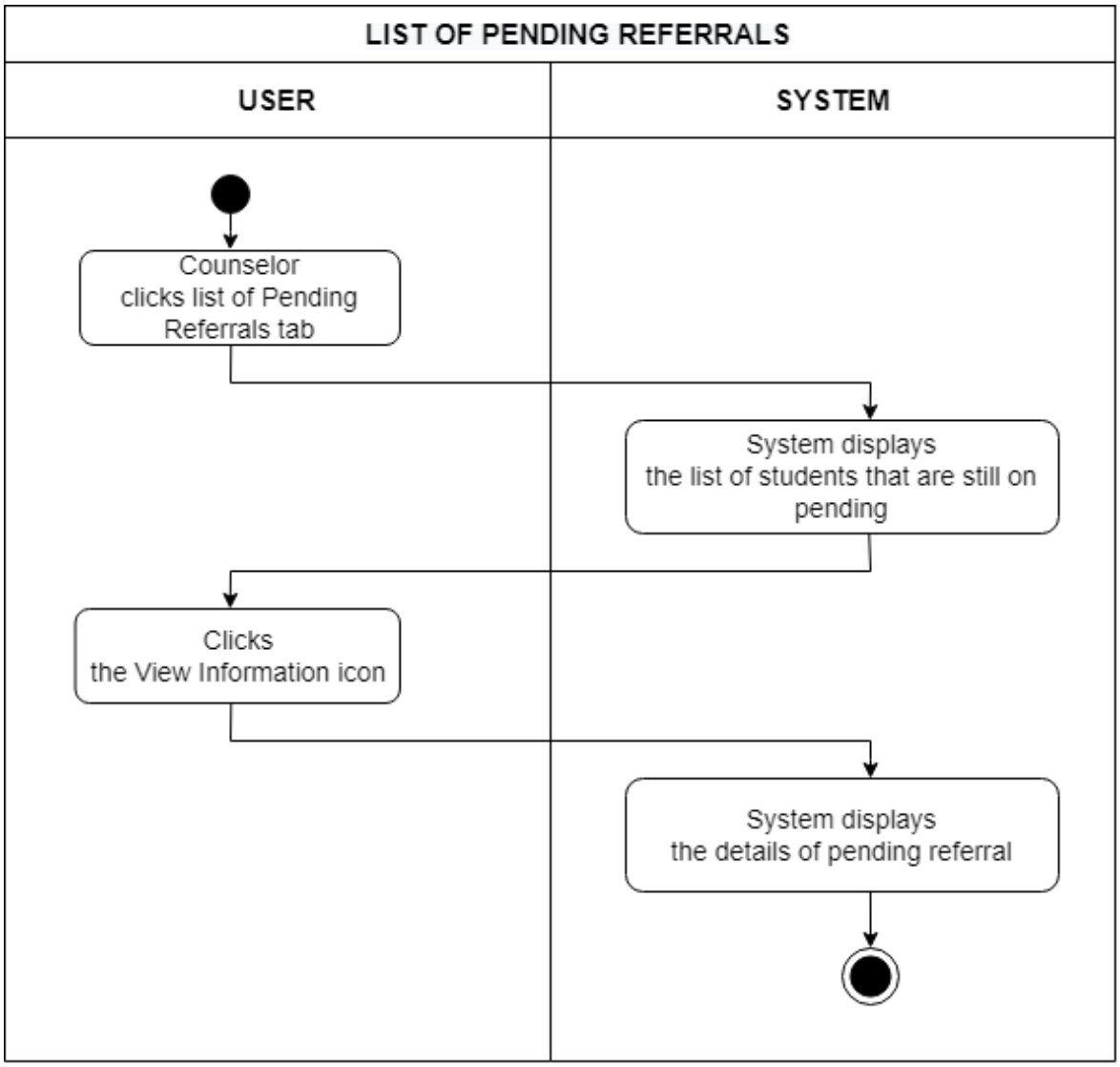

Figure 13: View List of Pending Referrals Activity Diagram

Figure 13 shows the viewing list of pending referrals. The counselor will click the list of pending referred students tab. The system will then display the list of students that still need to be counseled. The counselor clicks a specific information icon then the system will display the information of that particular student.

Counselor: Set Schedule

This activity diagram displays how the counselor sets a schedule.

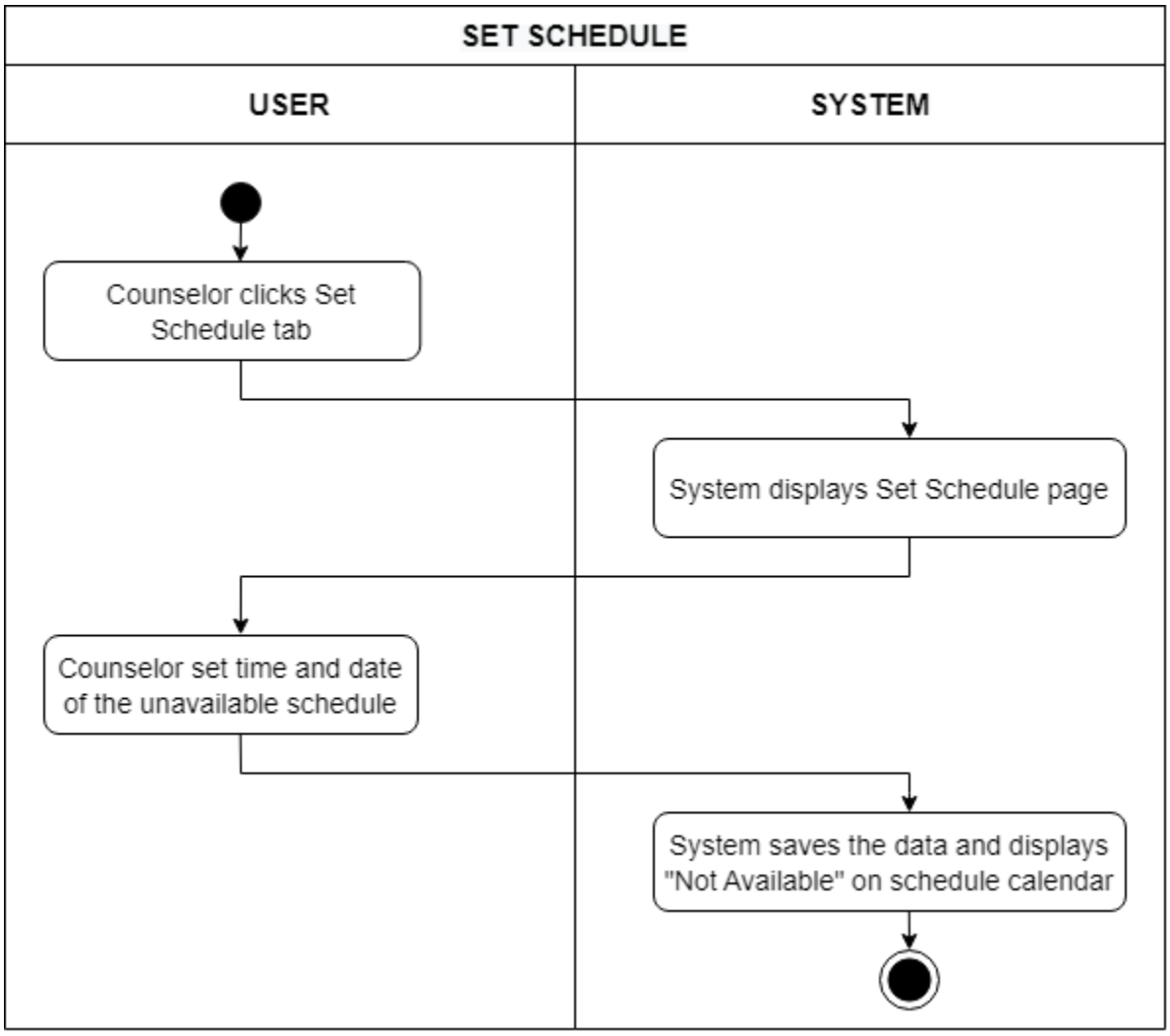

Figure 14: Set Schedule Activity Diagram

Figure 14 shows how the counselor can set a schedule. The counselor will click the set schedule tab. Then the system will display the UI where the counselor can set the date and time of his unavailable schedule. Then the system will save it and shows not available in the scheduled calendar.

Counselor: View Feedback

This activity diagram displays list of feedbacks made by the Counselor either it's a previous or a recent one.

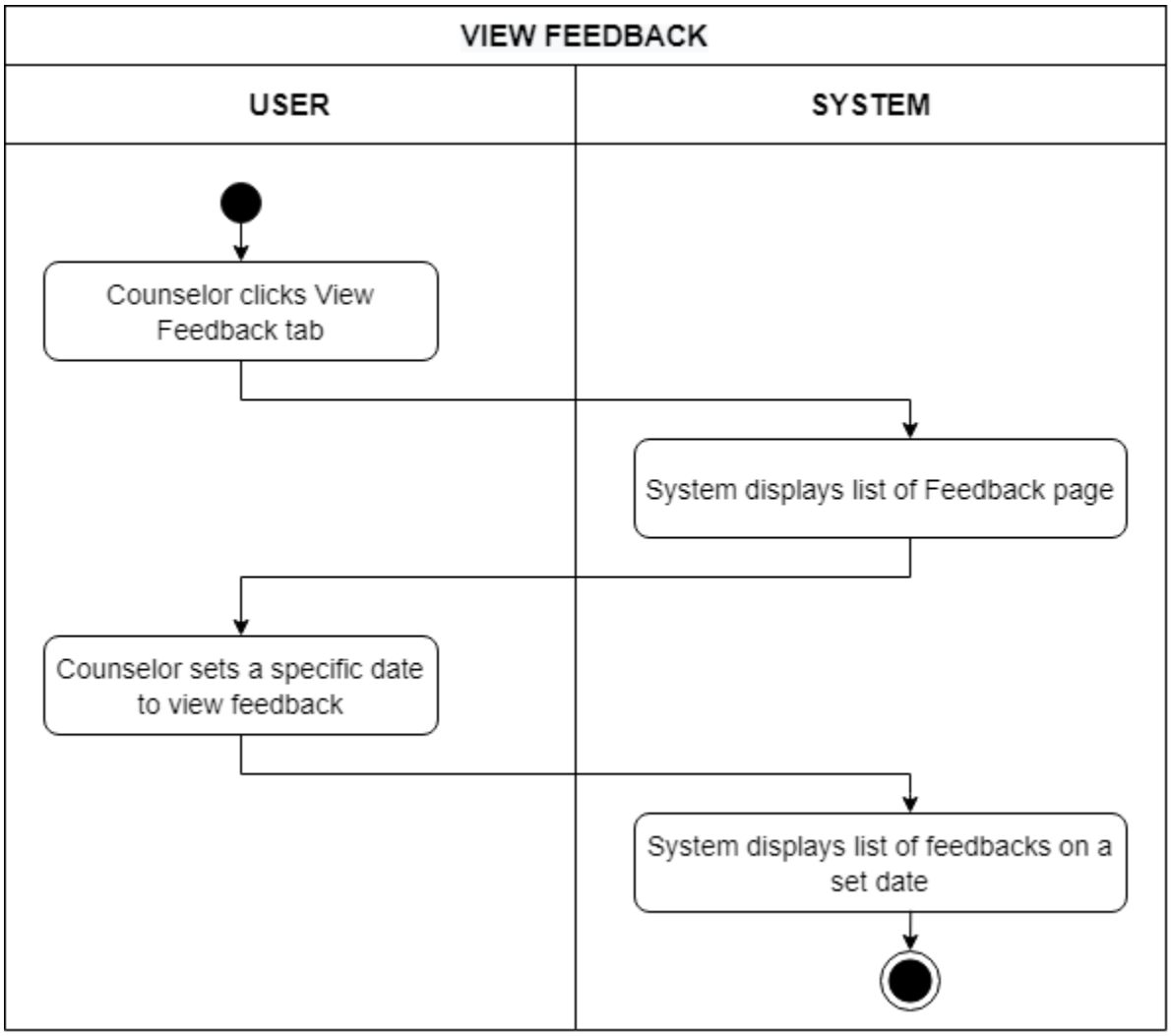

Figure 15: View Feedback Activity Diagram

Figure 15 shows the counselor's view of the previous feedback. The counselor clicks the view feedback tab. Then the system displays the list of feedback. The counselor filters the feedback by date and time then the system displays the feedback depending on the set date and time.

Counselor: Receives Referral Notification

This activity diagram displays how the counselor receives referral notification by Teacher and Student.

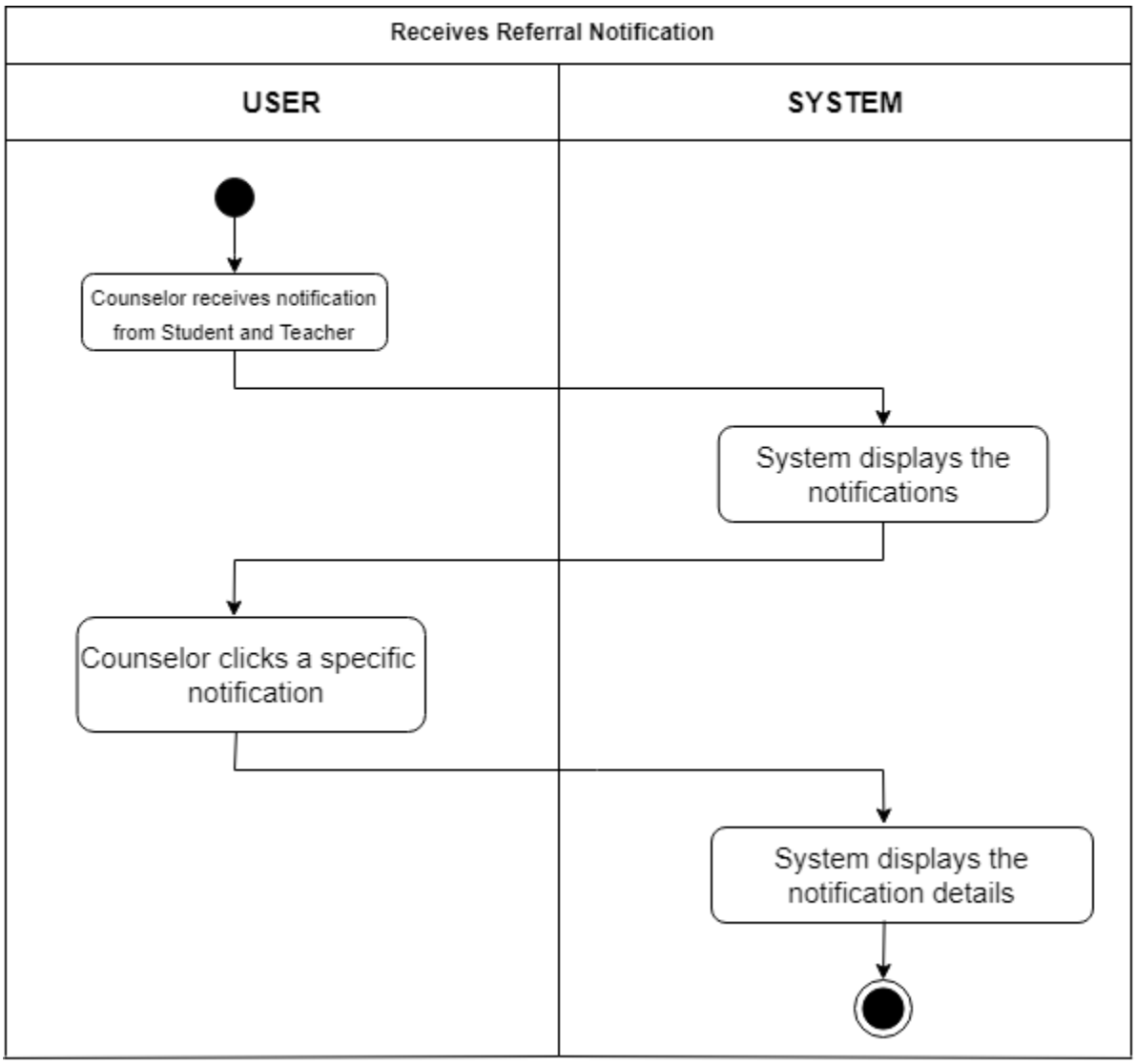

Figure 16: Receives Notification Diagram

Figure 16 shows the counselor receives the notification. The counselor clicks the notification tab. Then the system displays the list of notifications. The counselor clicks a specific notification then the system shows the details of the notification.

Teacher: Refers Student

This activity diagram displays how the teacher refers a student through to the courses where the student is enrolled.

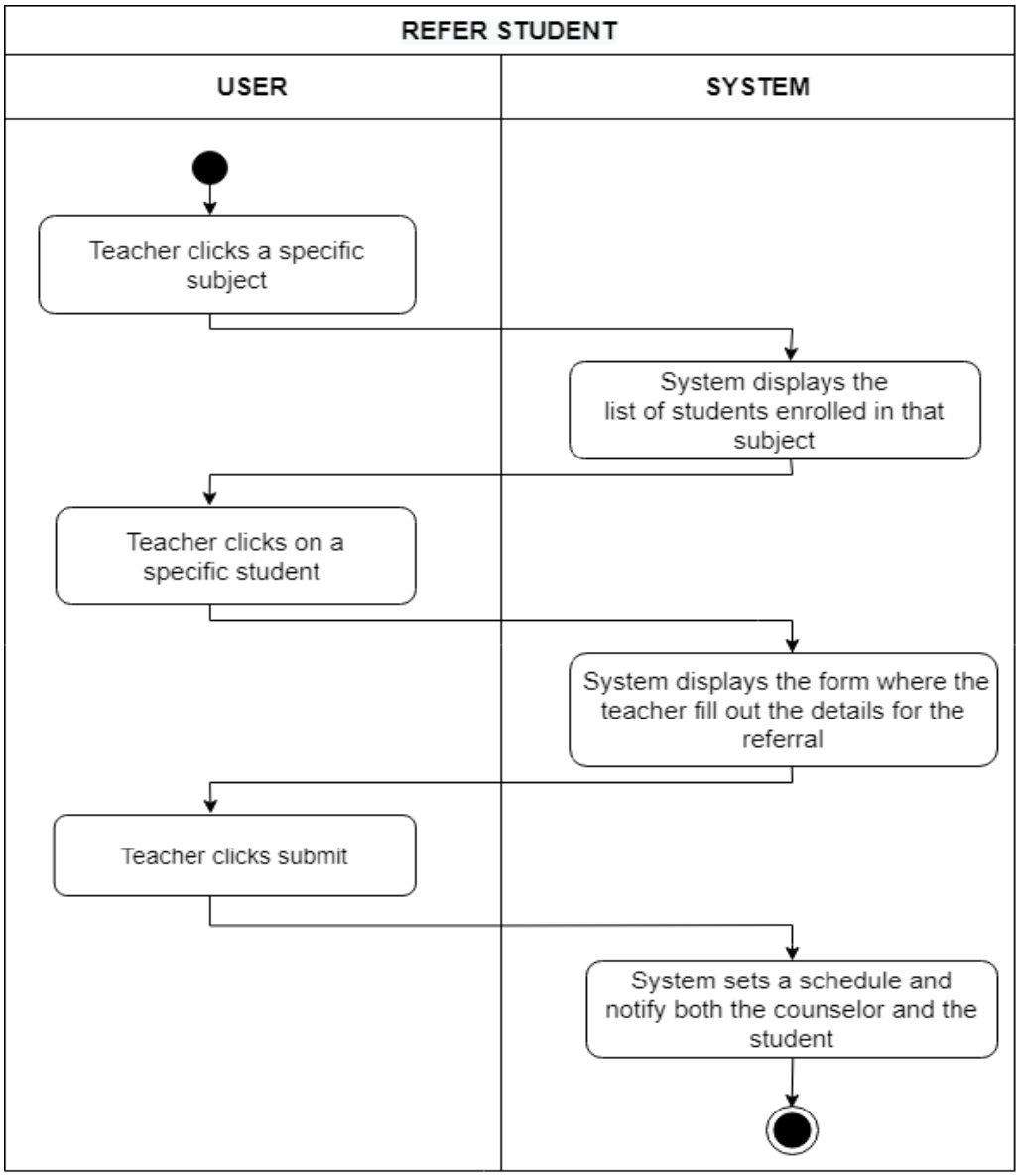

Figure 17: Refer Student Activity Diagram

Figure 17 shows the teacher manually referring a student. The teacher clicks a specific subject then the system displays the list of students enrolled in that subject. The teacher clicks on a specific student then the system displays the form where the teacher fills out the details for the referral; then, after clicking submit, the system sets a schedule and notify both the counselor and the student.

Teacher: View Referred Student

This activity diagram displays how the teacher views his/her referred students.

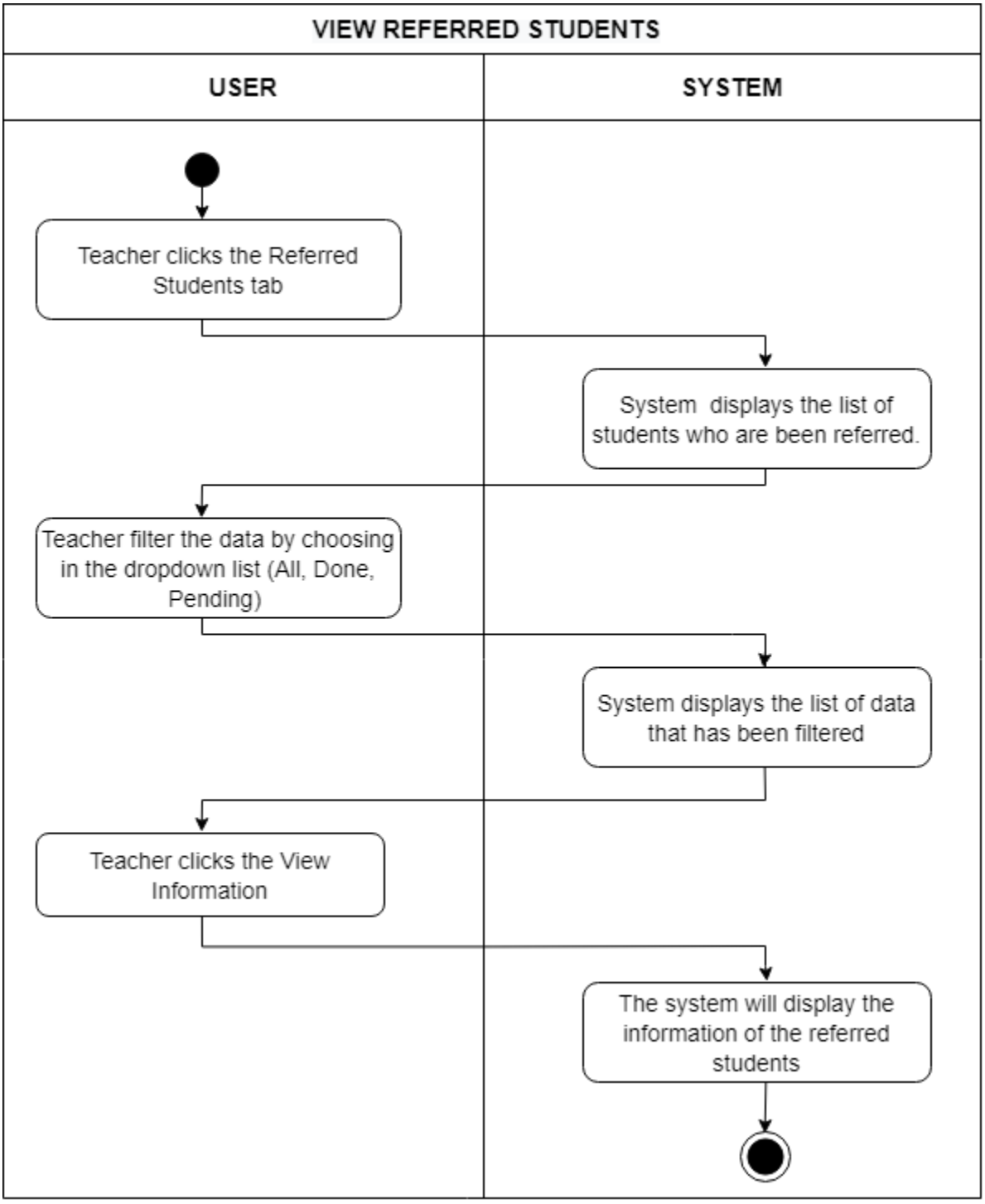

Figure 18: View Referred Student Activity Diagram

Figure 18 shows the teacher viewing referred students. The teacher clicks a view referred student tab. The system displays the list of students referred by that specific teacher. The teacher filters the status in the drop-down load as done, pending, or showing all students. After setting a filter, the system displays the students that are filtered.

Teacher: Receives Feedback Notification

This activity diagram displays how the teacher receives the feedback notification from the counselor.

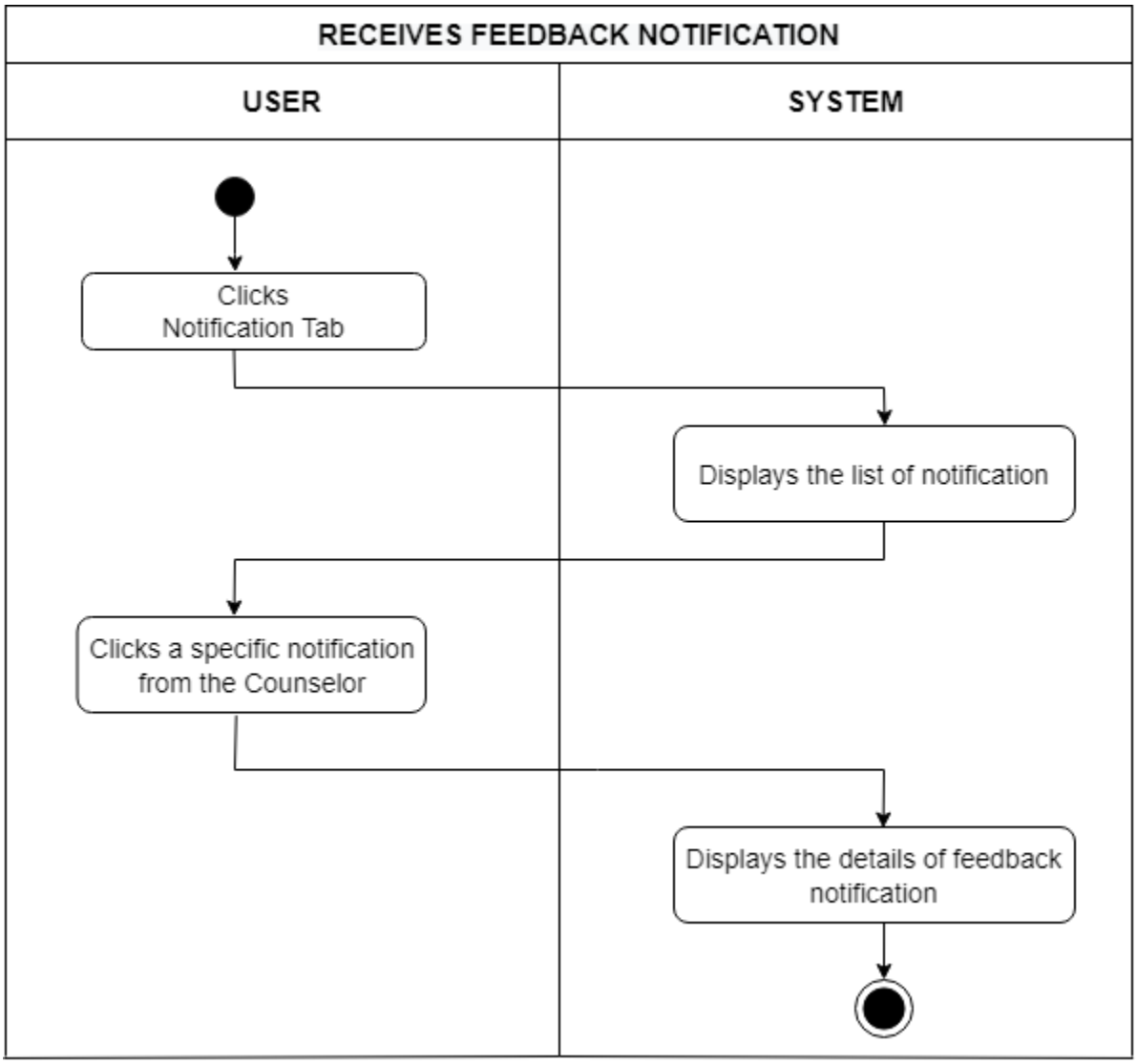

Figure 19: Receives Feedback Notification Activity Diagram

Figure 19 shows the received feedback notification. The teacher clicks the notification tab then the system will display the list of notifications. The teacher clicks a specific notification then the system will display the feedback details.

Student: Set an Appointment

This activity diagram displays how the student sets an appointment for counselling session.

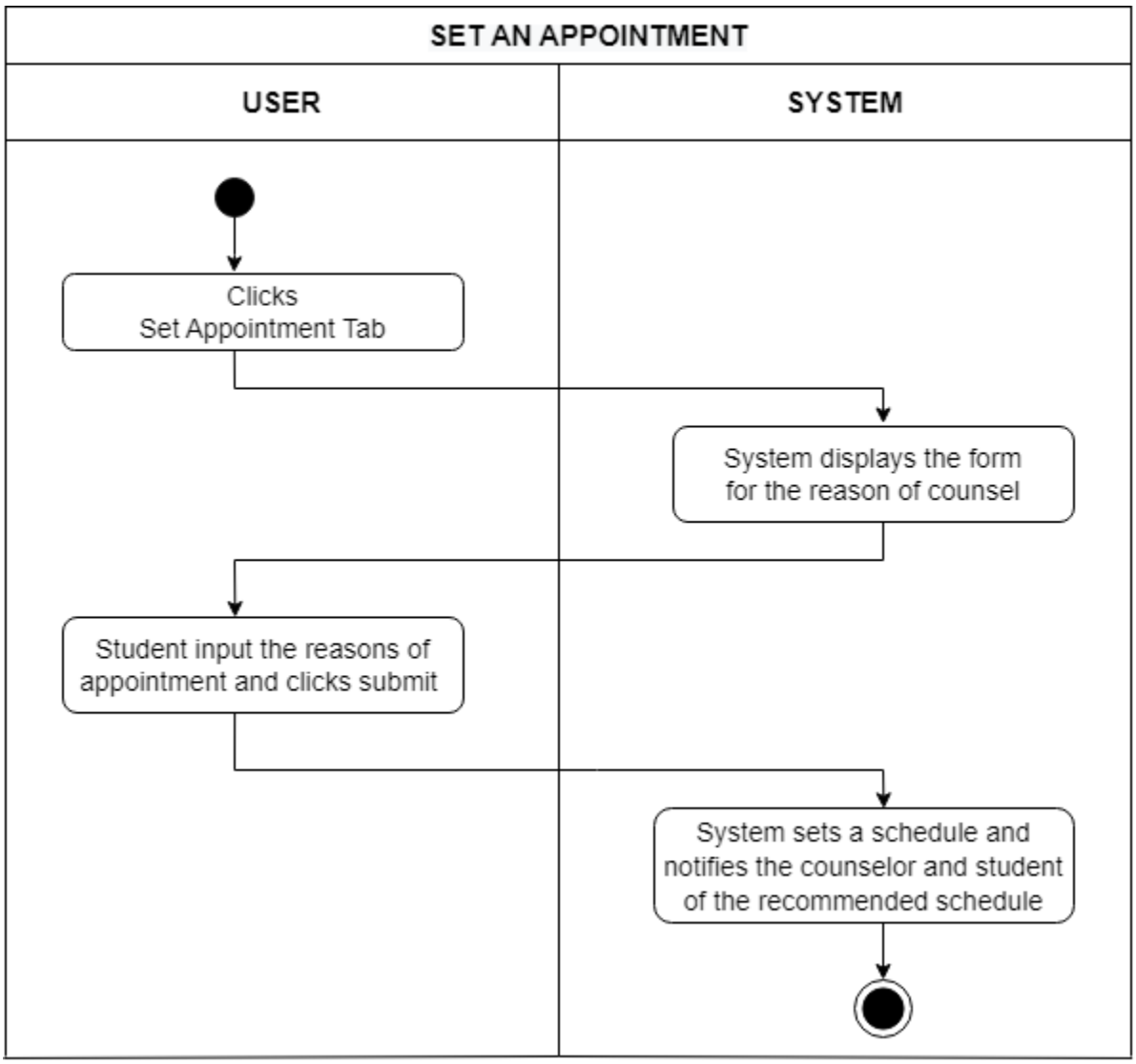

Figure 20: Set an Appointment Activity Diagram

Figure 20 shows a student setting a counseling schedule. The student clicks the set schedule tab then the system displays the form for the reason of counsel. After clicking submit, the system sets a schedule and notifies the counselor and student of the recommended schedule.

Student: View Schedule

This activity diagram displays how the student views the schedule.

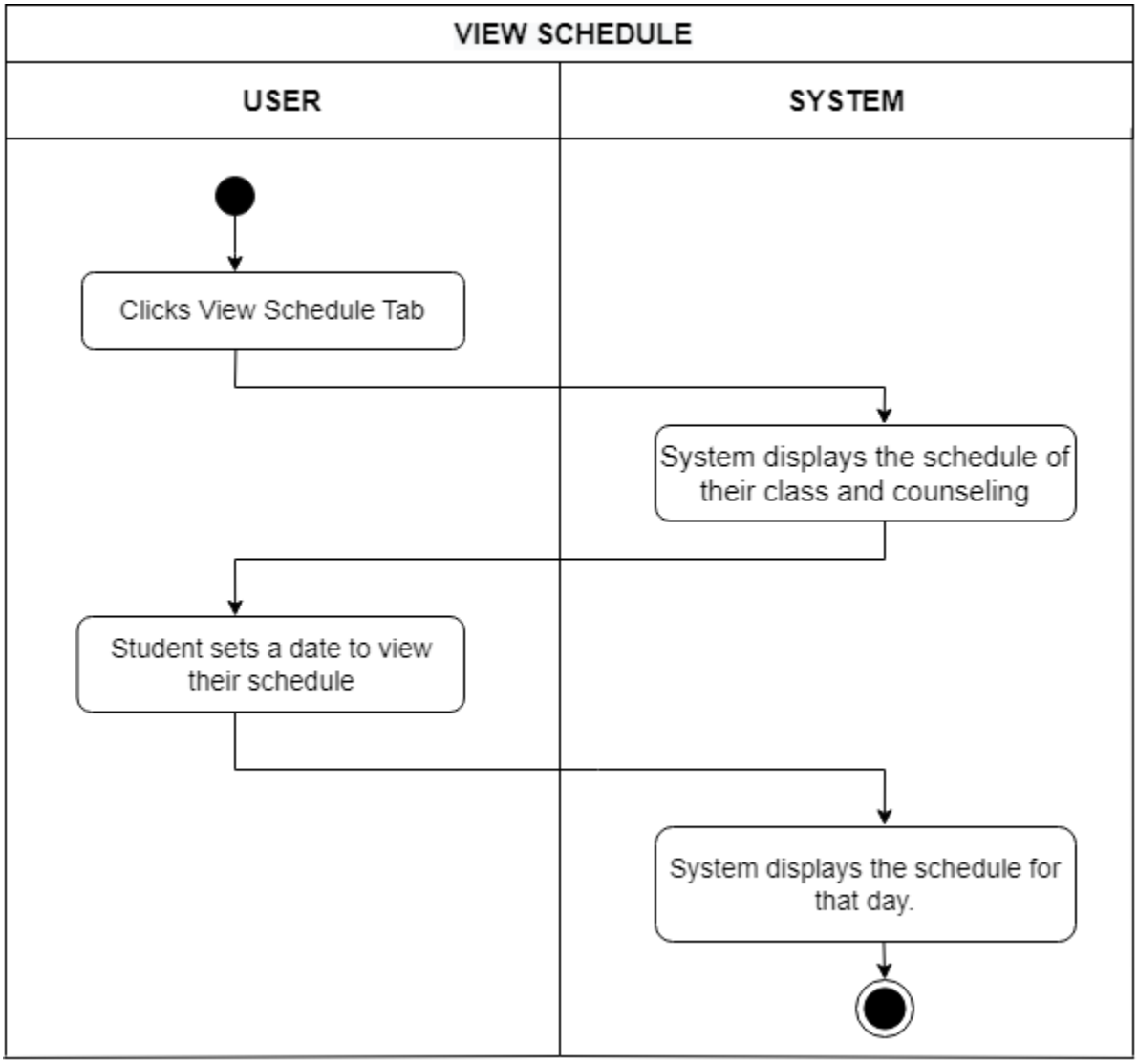

Figure 21: View Schedule Activity Diagram

Figure 21 shows a student viewing schedule. The student clicks the view schedule tab then the system displays the schedule of their class and counseling. Students set a date to view their schedule then the system displays the schedule for that day.

Student: View History

This activity diagram displays how the student views the history.

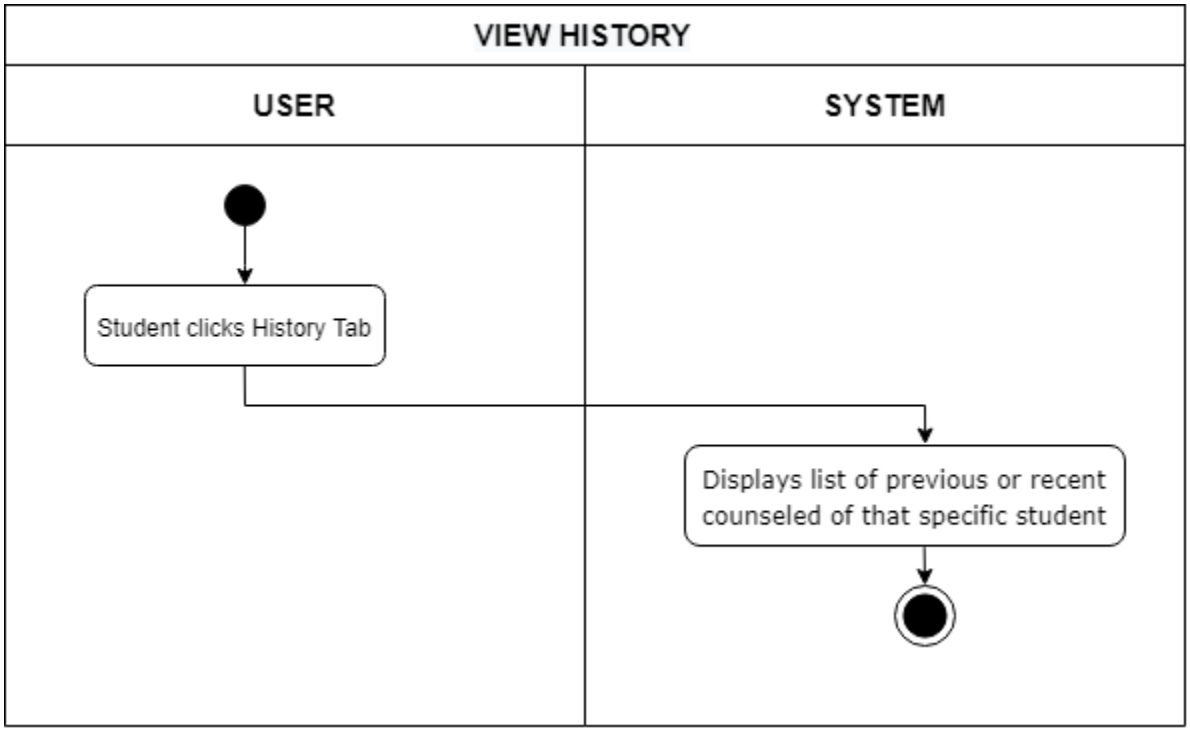

Figure 22: View History Activity Diagram

Figure 22 shows a list of history. The student clicks the history tab then the system displays the list of previous counseled of that specific student.

Student: View/Edit Profile

This activity diagram displays how the student view or update his/her credentials.

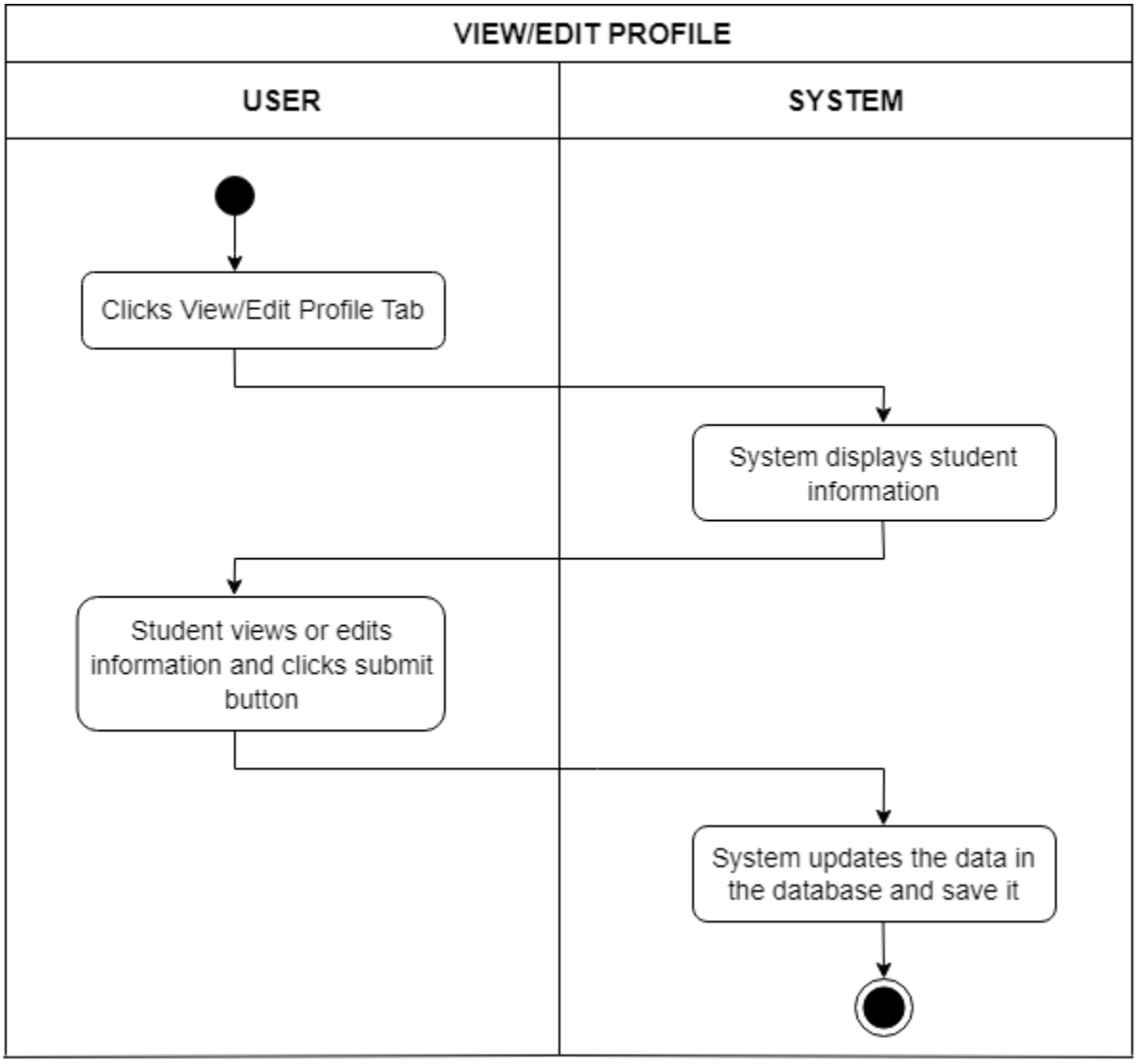

Figure 23: View/Edit Profile Activity Diagram

Figure 23 shows the edit of the student information. The student clicks the edit profile tab then the system displays the information. Then the student will modify some of the fields, and after clicking the submit button, the system updates the data in the database.

DATA STRUCTURE DESIGN (Class Diagram or Database Design)

The Entity Relationship Diagram shows all the relationships of the tables.

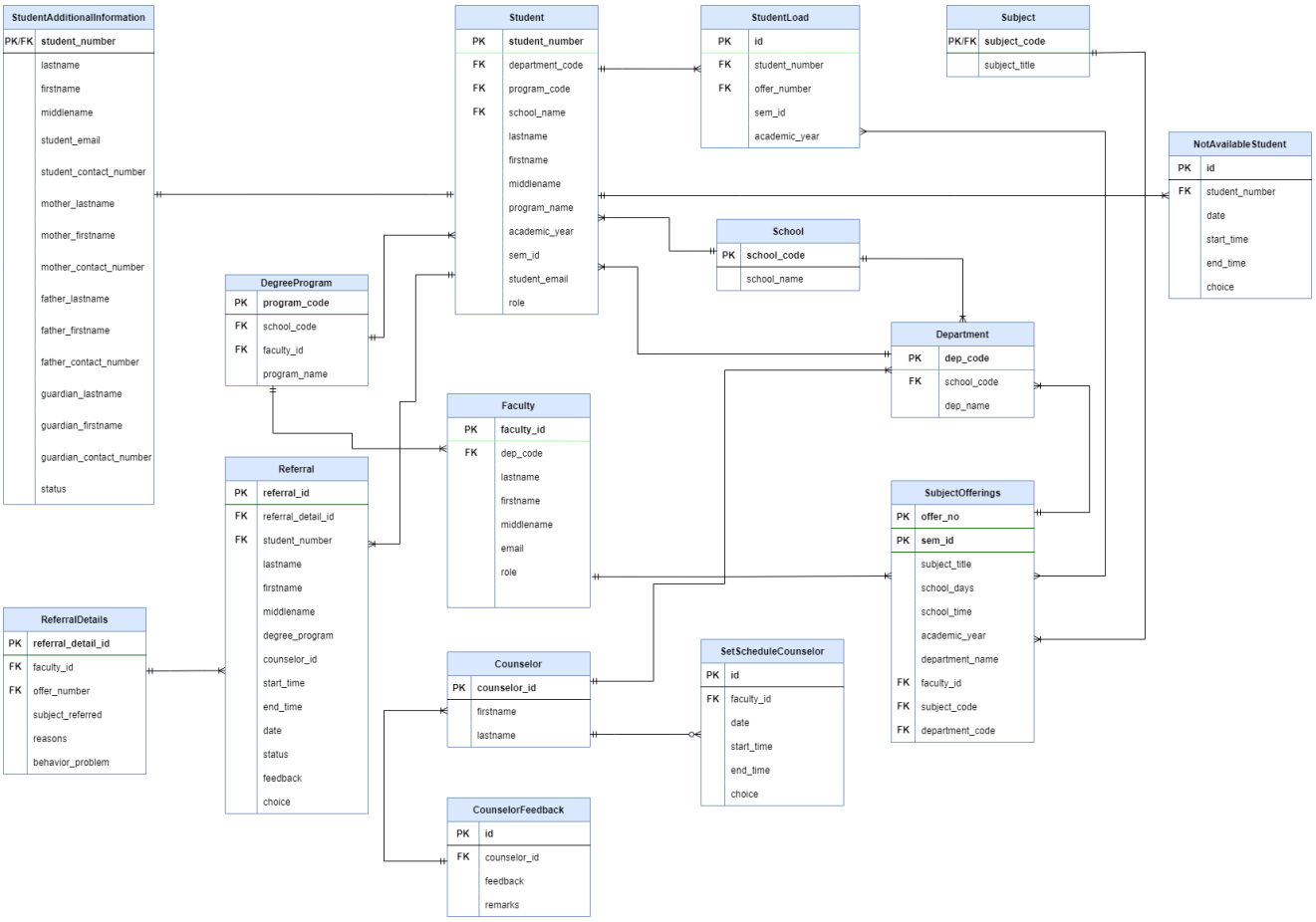

Figure 24: Follow App Entity Relationship Diagram

Figure 24 describes the structure and relationships between the system's components and their classes and the behaviors that make and complete the entity.

USER INTERFACE

Landing Page

This is the Welcome Page of the Follow App, where everything starts by clicking the button Get Start Here.

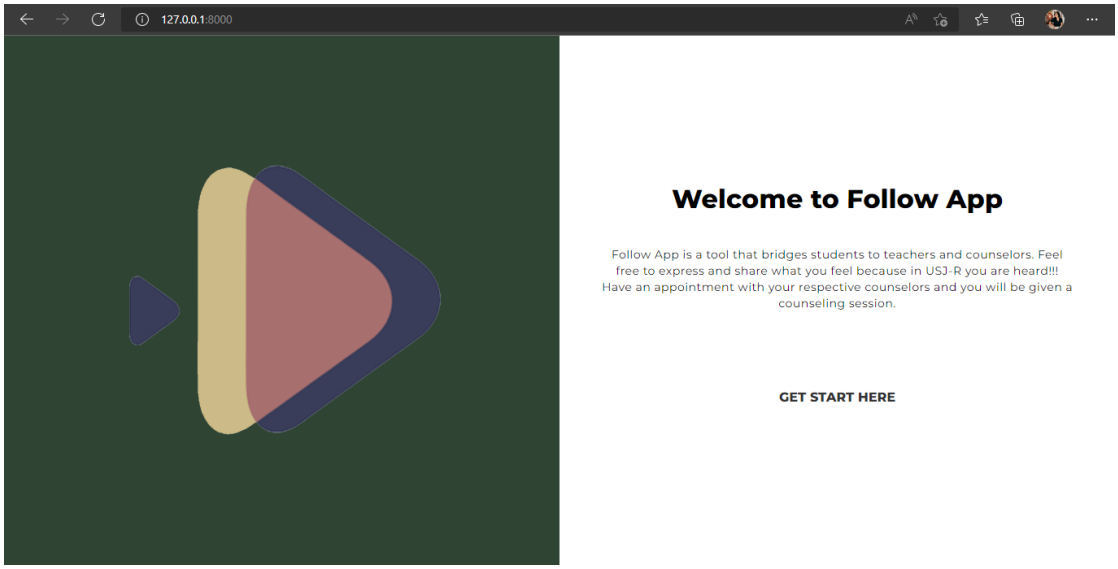

Figure 25: UI Design for Landing Page

Login Page

On this screen, the user must input his/her ID Number and Password to log in.

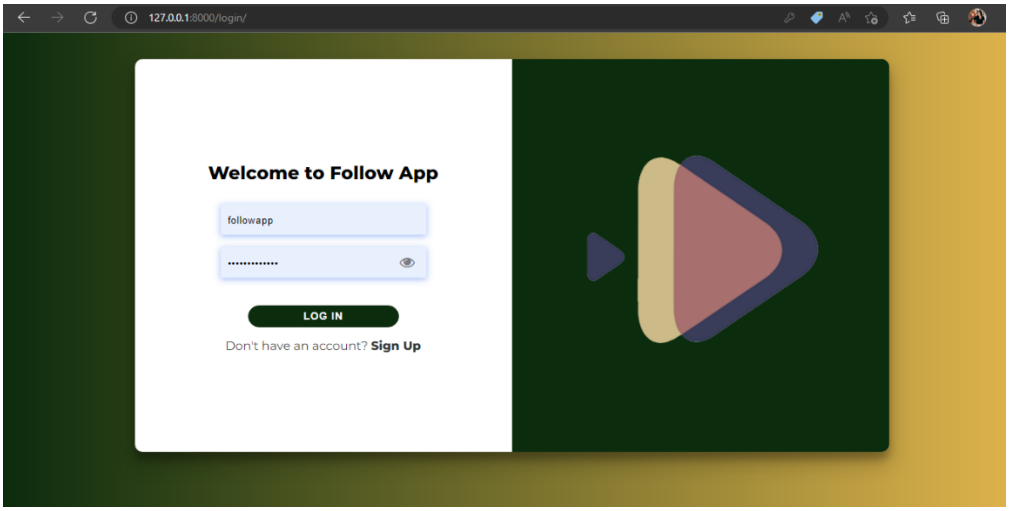

Figure 26: UI Design for Login

Sign Up Page

The user must input his/her provided ID Number and Phone Number for code verification.

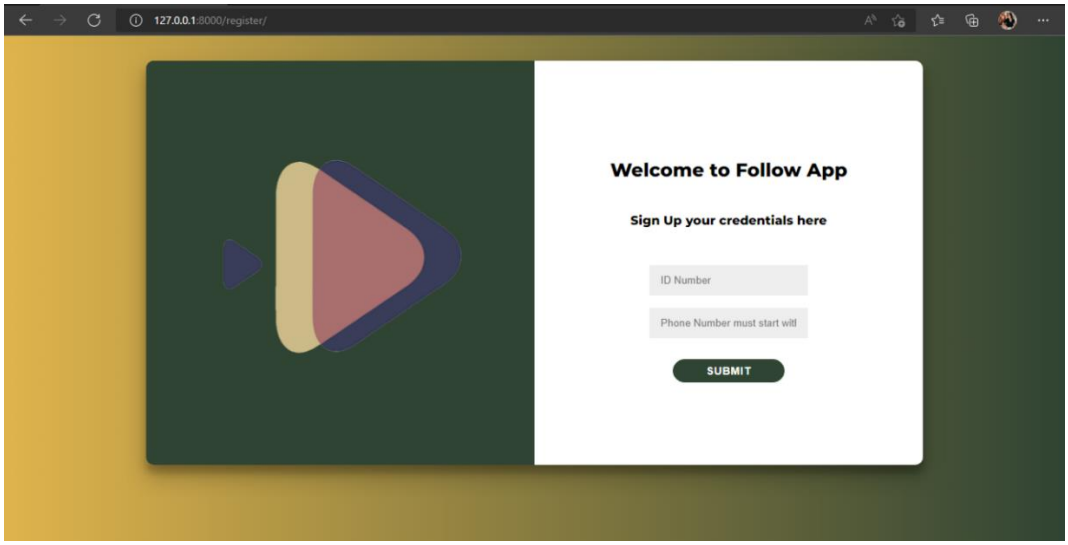

Figure 27: UI Design for Sign Up

Verification Code Page

Upon signing in, the user will input the verification code sent by SMS.

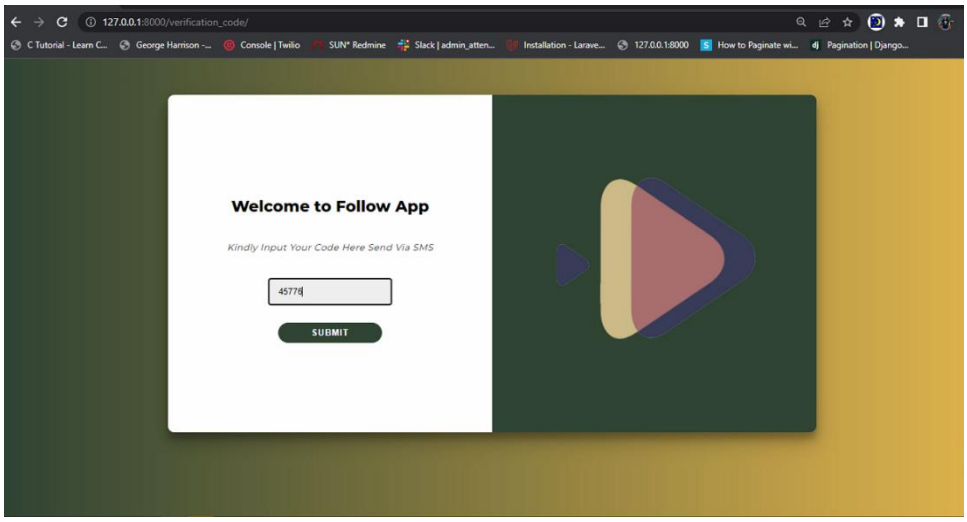

Figure 28: UI Design for Verification Code

Register Page

To complete the process, the user must input his/her ID Number again and creates his/her password.

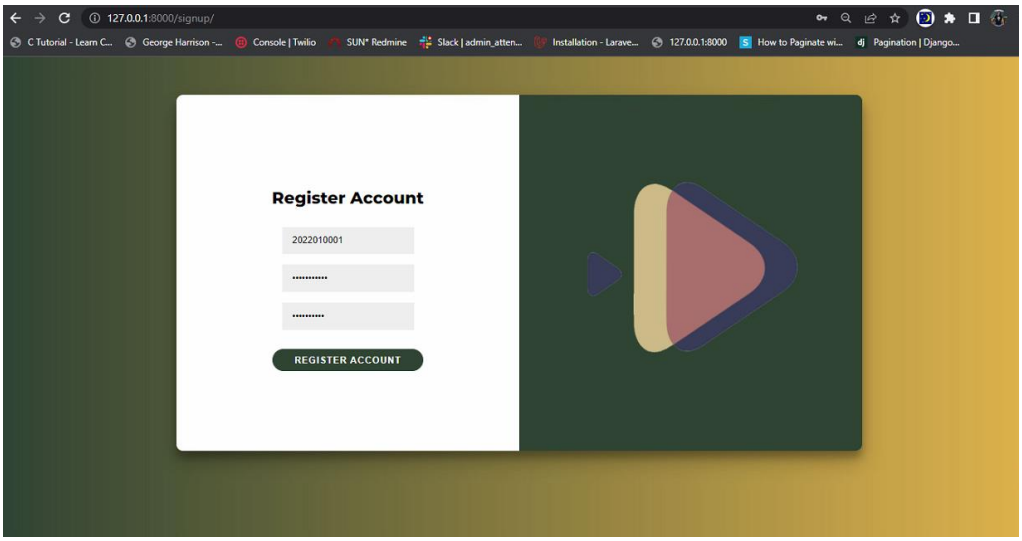

Figure 29: UI Design for Register

Admin Upload Data

The admin can upload the data through a csv file.

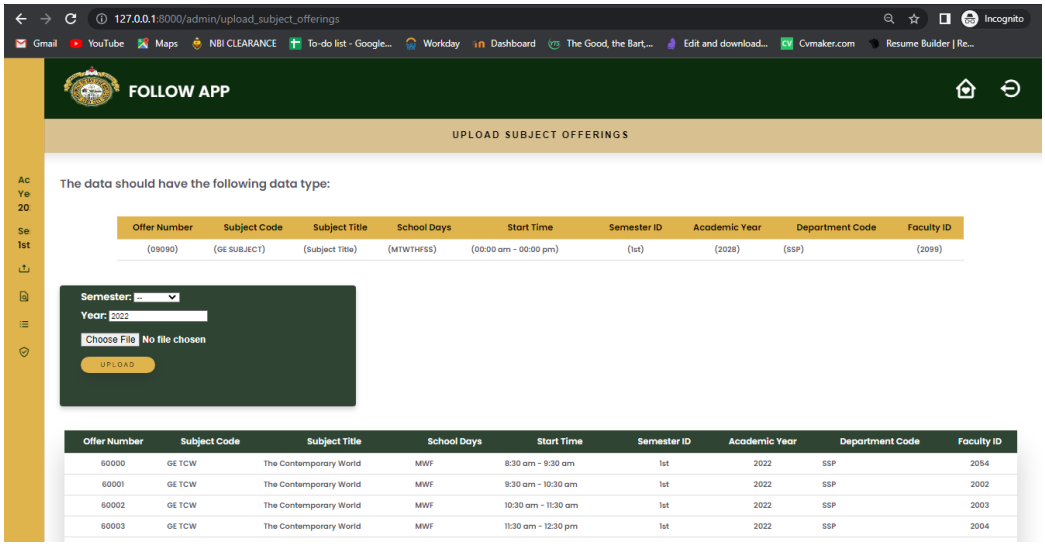

Figure 30: UI Design for Upload Data of Admin

Admin Uploads Data from EDP and LMS and Automatic Referral

The admin uploads data and students will be automatically referred.

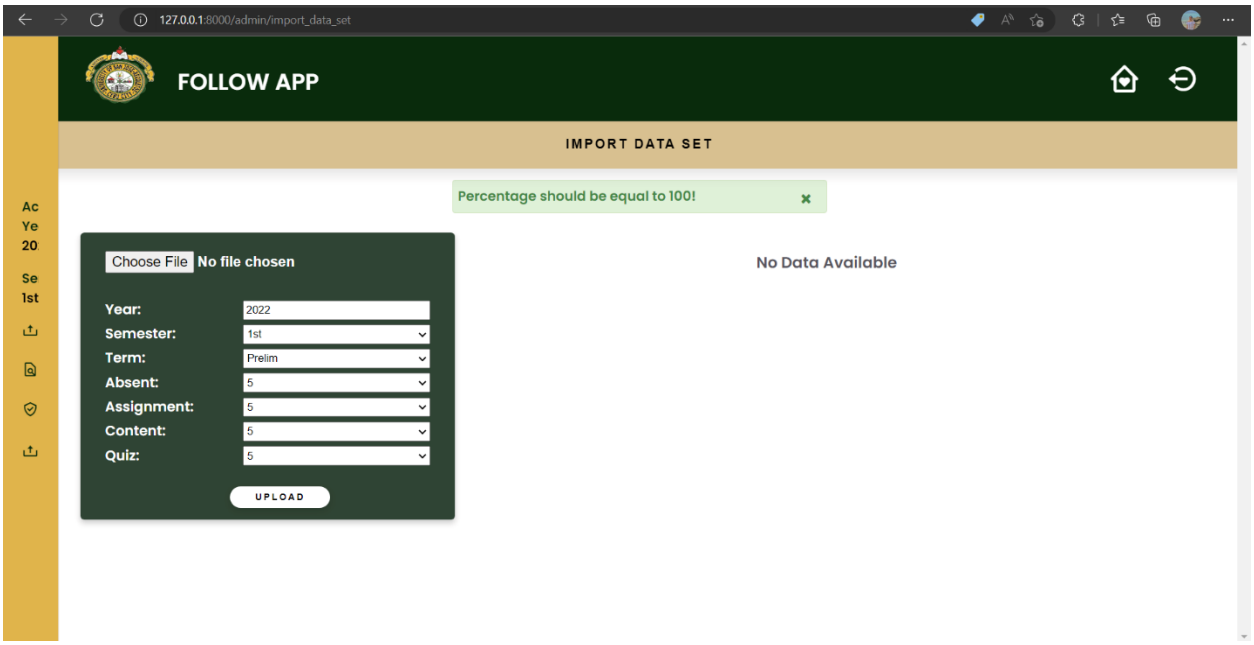

Figure 31: UI Design of Admin Upload of SIS and LMS

The list of students that are automatically referred based on the EDP and LMS data uploaded.

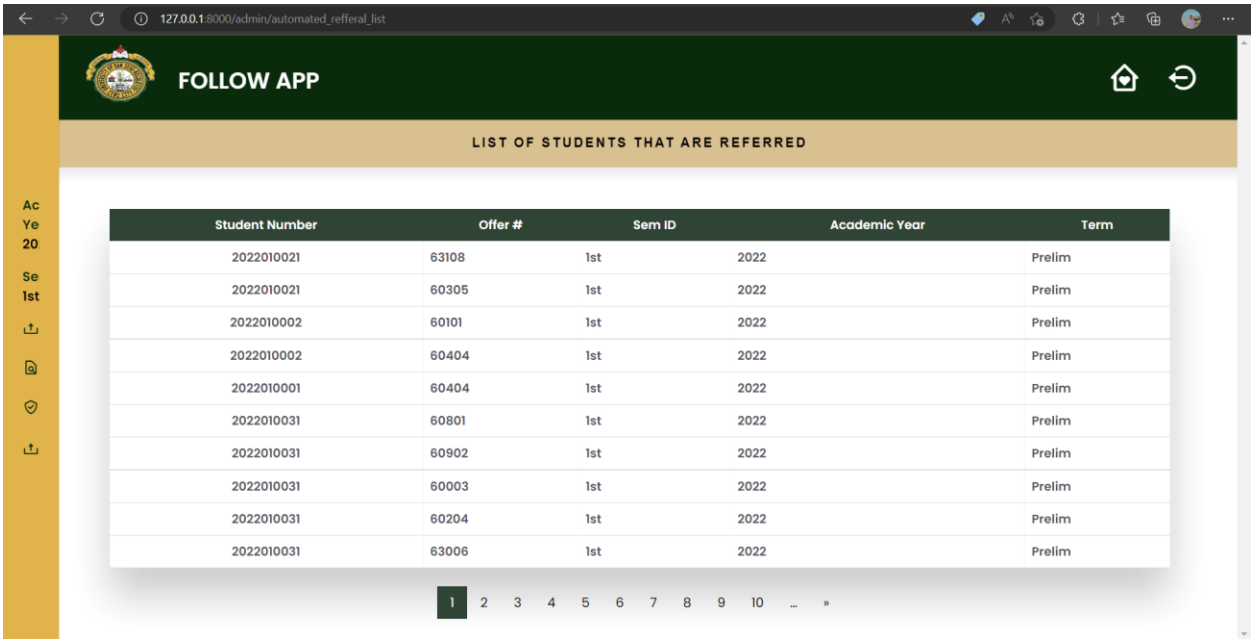

Figure 32: UI Design of List of Referred Students

View Degree Programs Page

Director views the different list of degree programs, where he/she can assign a specific counselor on that program.

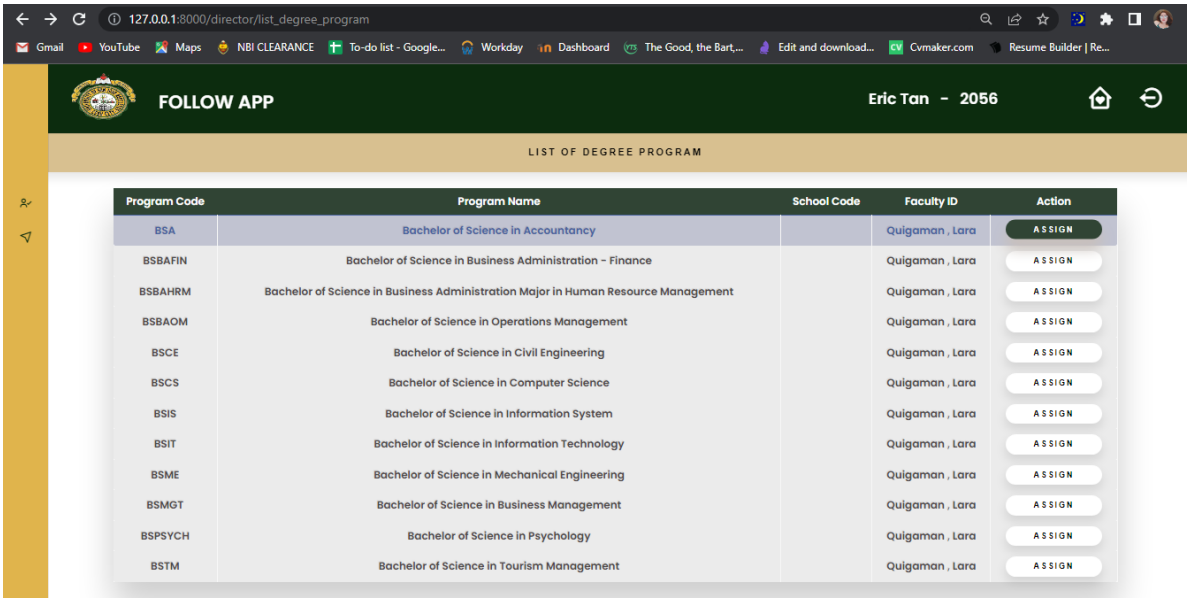

Figure 33: UI Design for View Degree Program in Director

Assign Counselor Page

Director chooses a counselor from the list assigns him to a degree program.

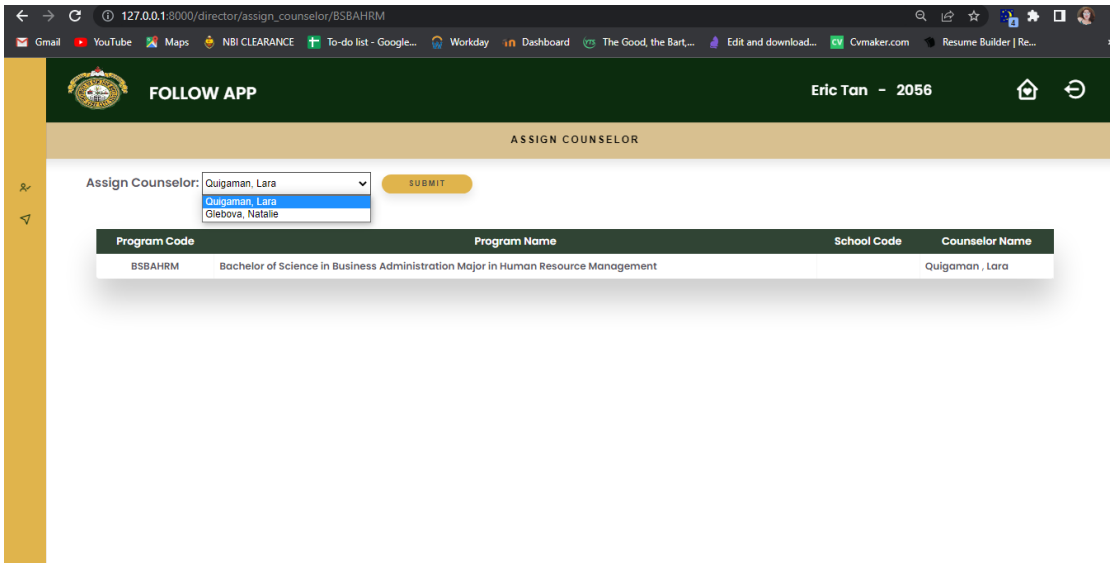

Figure 34: UI Design for Assign Counselor

Subject Page

Teacher clicks a specific subject to refer a student.

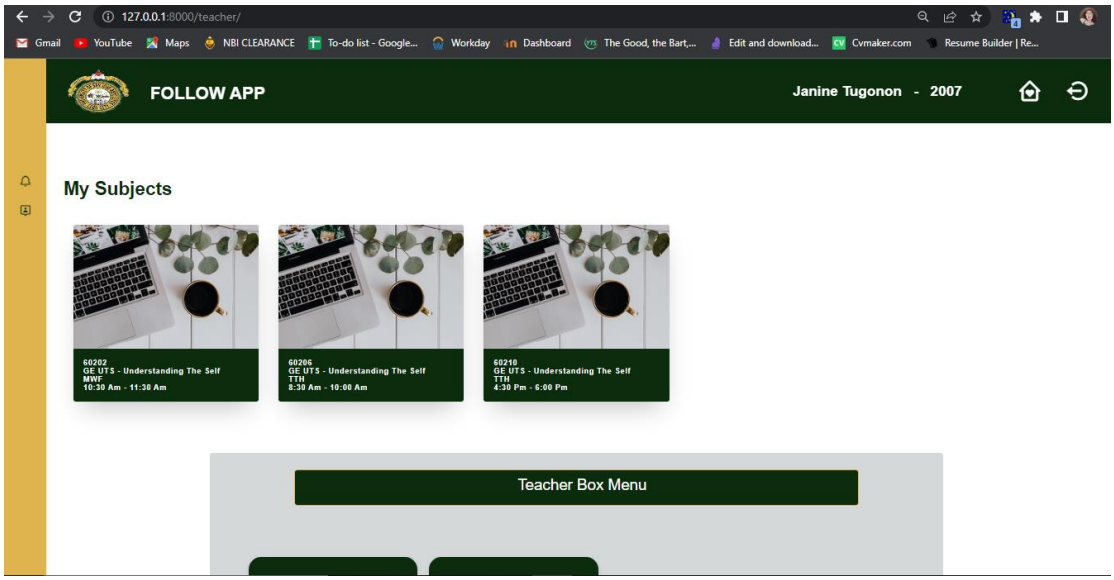

Figure 35: UI Design for Subject Page

Refer Student Page

Teacher clicks a subject where list of students is shown, and clicks Refer Student for a student referral.

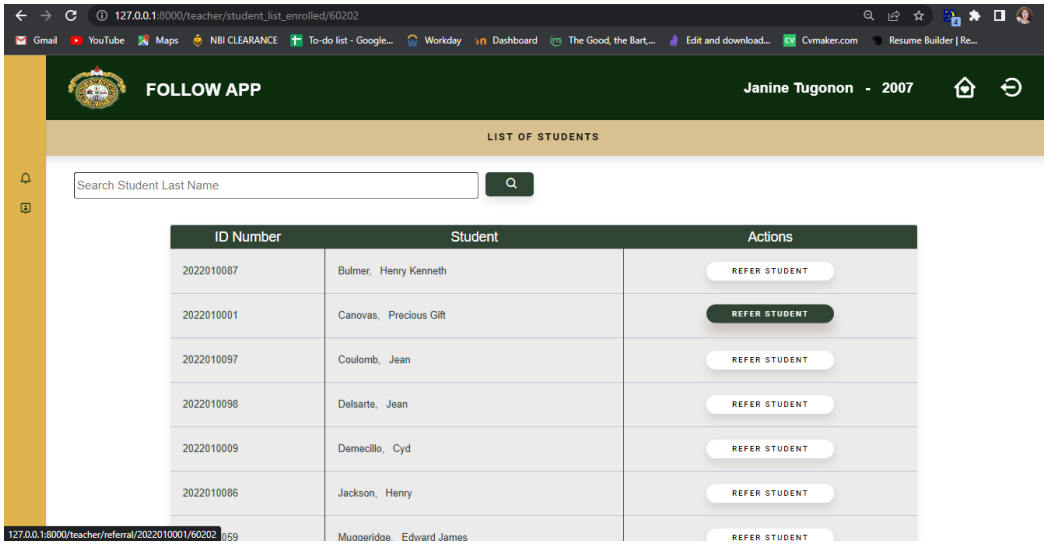

Figure 36: UI Design for Referring Student

Referral Form

Teacher provides the referral form

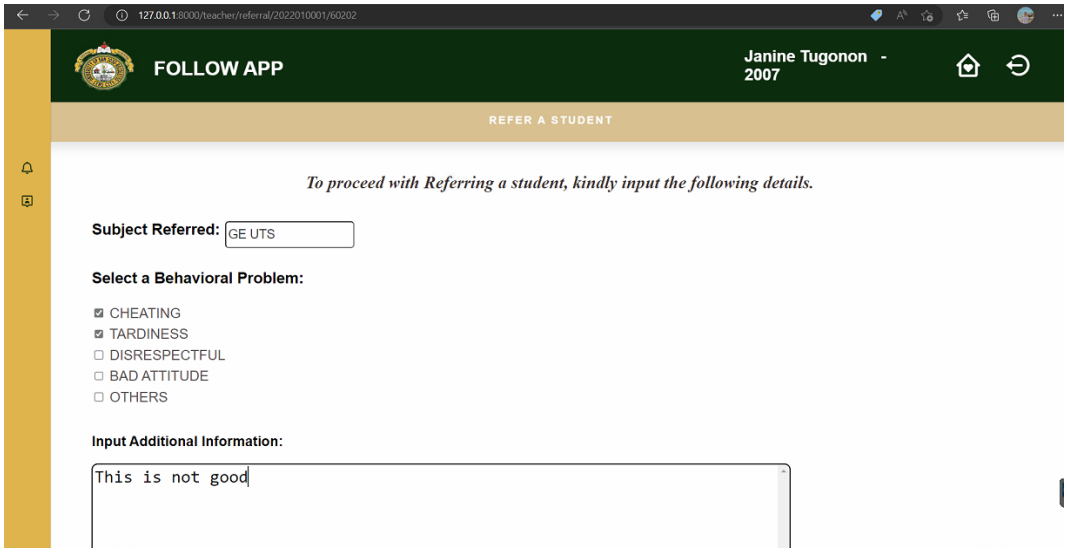

Figure 37: UI Design for Referral Form

Student Sets for An Appointment

Student can set an appointment to the Counselor for counseling session.

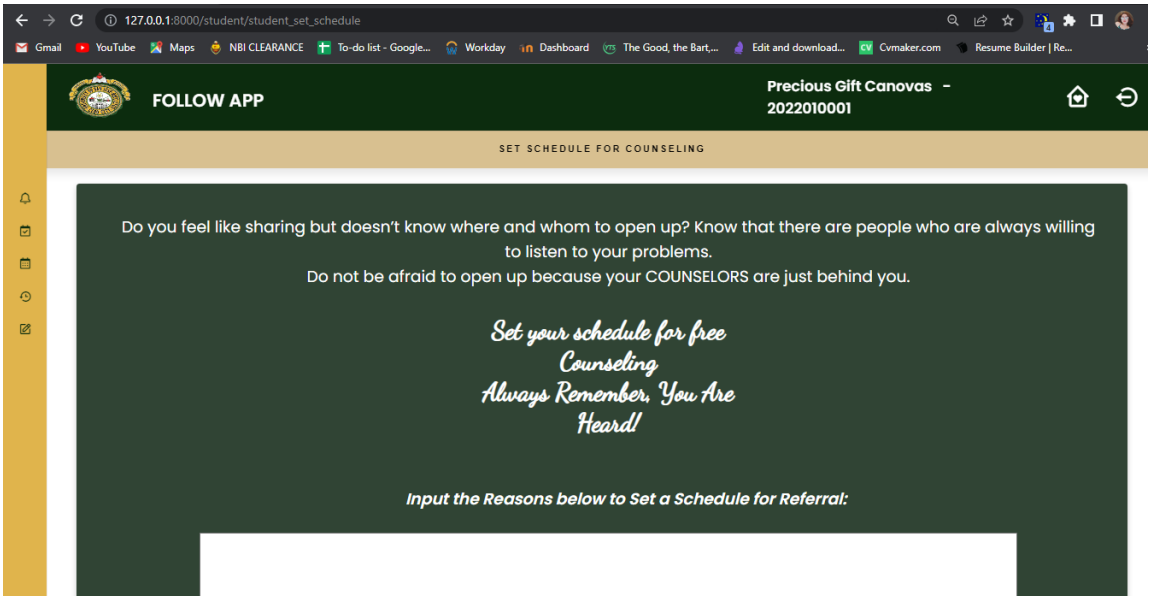

Figure 38: UI Design for Student Sets an Appointment

Counselor Receives Referral

The counselor views the information of the referral and chooses either to accept or decline. If the counselor chooses to accept the session, he will be notified 10 minutes before the counseling within that scheduled day.

On the other hand, if the counselor declines a session, the system provides another schedule for both the counselor and the student.

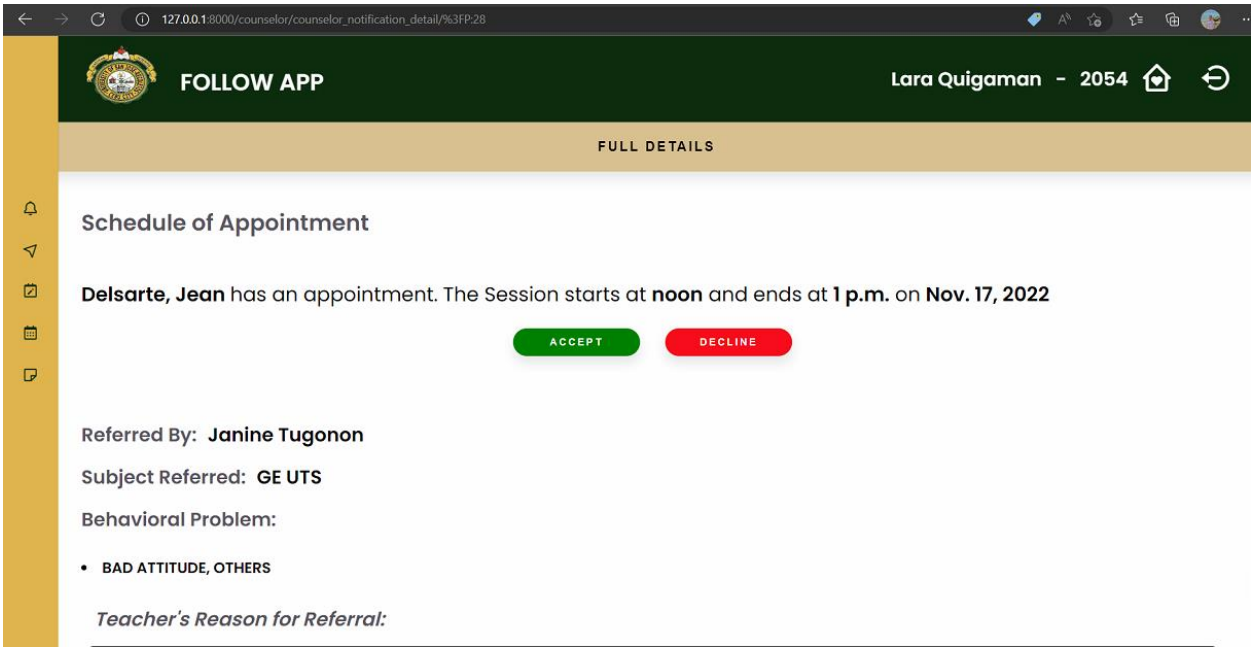

Figure 39: UI Design for Counselor Accepts/Decline Schedule

Counselor Receives the Appointment from Student

The notification details if the counselor receives an appointment from a student who wants a counseling session.

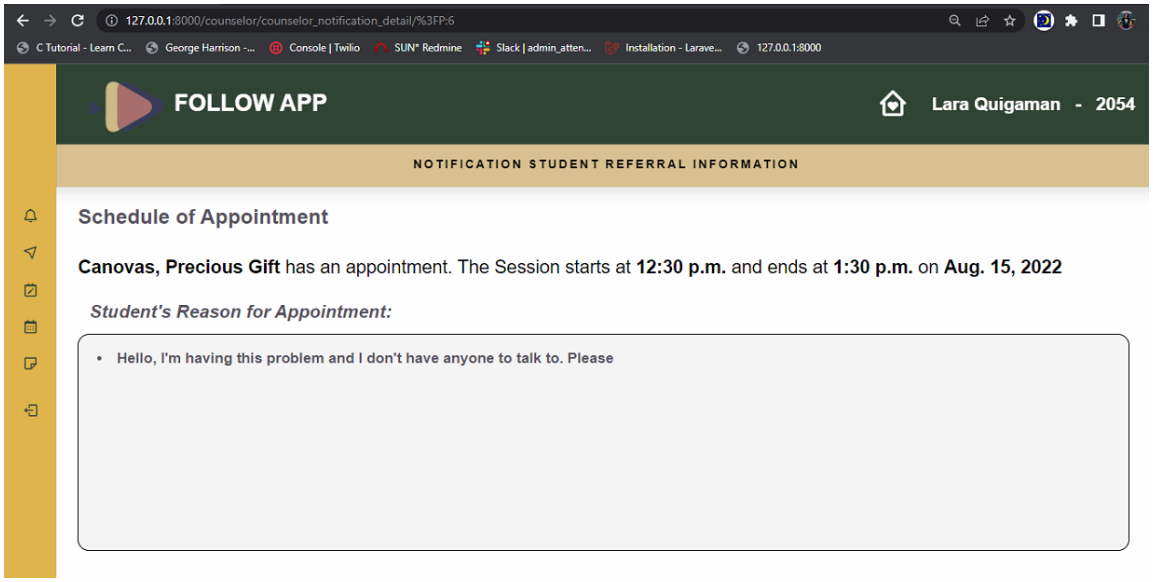

Figure 40: UI Design for Counselor Receives an Appointment

Counselor Receives Referral from Teacher

The notification details if the counselor receives a referral from a teacher.

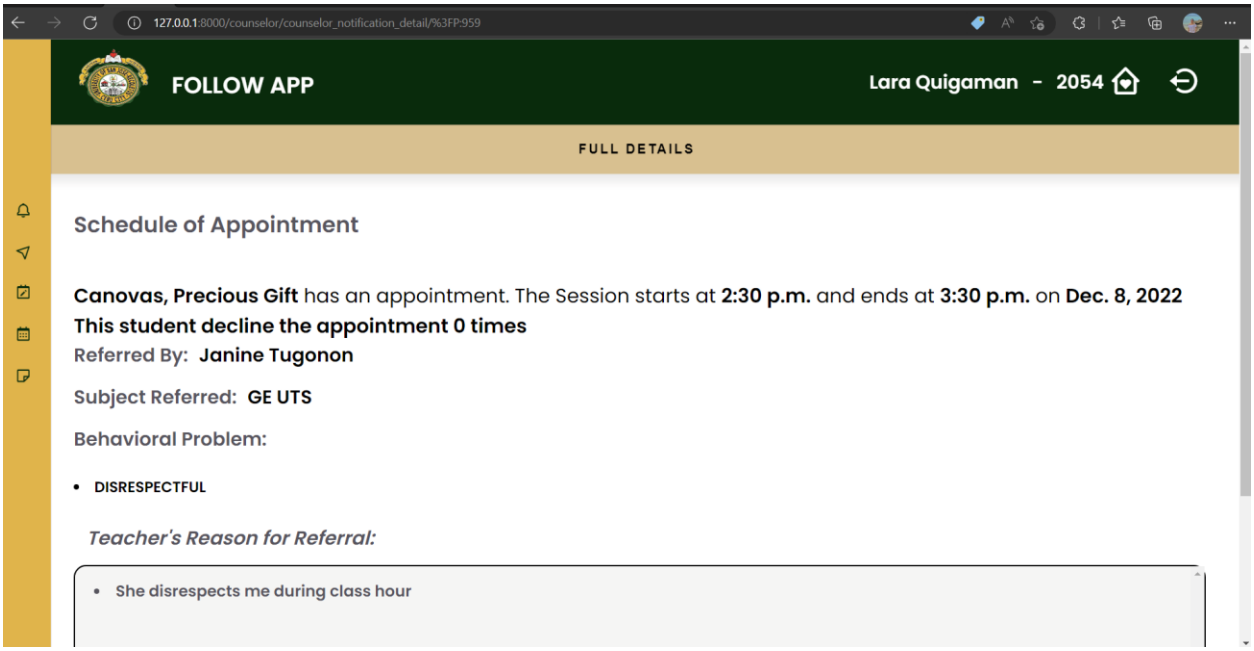

Figure 41: UI Design for Counselor Receives Referral from Teacher

Student Receives Referral

The student receives the counseling referral and upon viewing the information, the student may accept or decline the session. If the student chooses to Accept the session, he will be notified 10 minutes before the counseling, hence, if the student declines the session, the system will provide another schedule, as well as the counselor. When the student rejects the counseling three times in a row, Follow App will immediately contact the parents/guardian of the student through SMS.

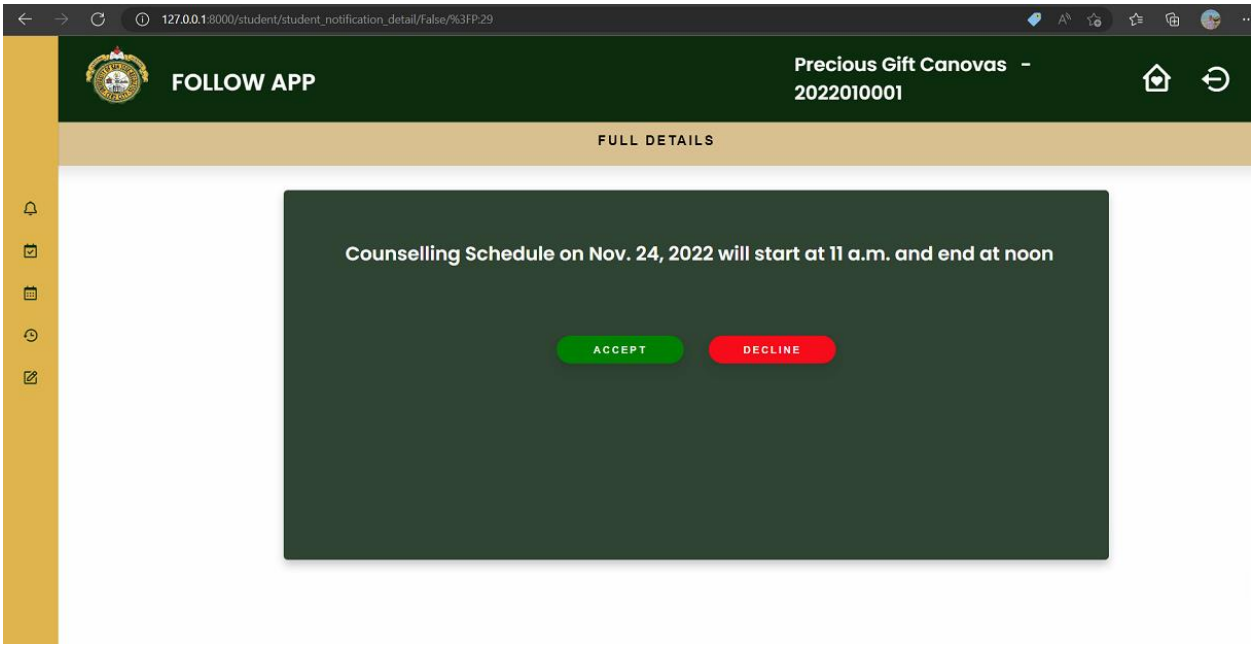

Figure 42: UI Design for Student Accepts/Decline Schedule

Joining the Session

Both users (counselor and student) are required to join the session. Upon tapping the, "Join" button, they will be redirected to the counseling session.

For any circumstances that both users (counselor and student) will not show up, they can just click the button provided by the Follow App, containing a text of, "Click here if student/counselor is not available". By clicking the button, both users (counselor and

student) will receive a notification of a new set of counseling schedule.

Counselor will join the meeting.

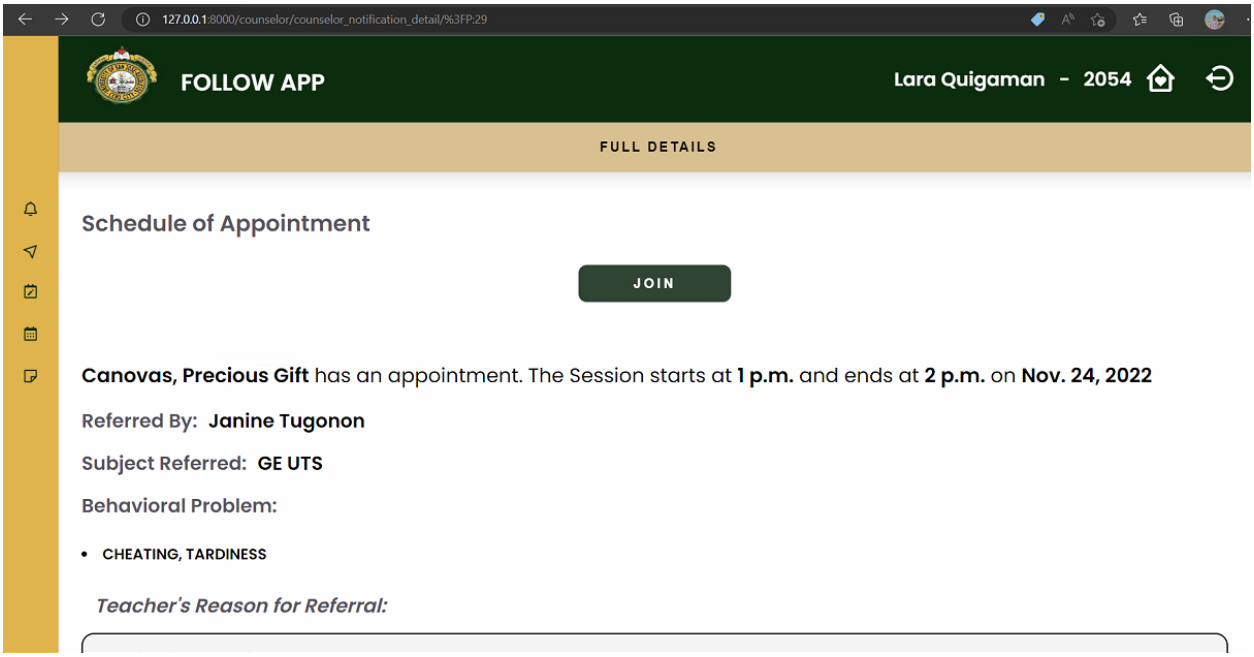

Figure 43: UI Design for Counselor Joins the Session

Student will join the meeting.

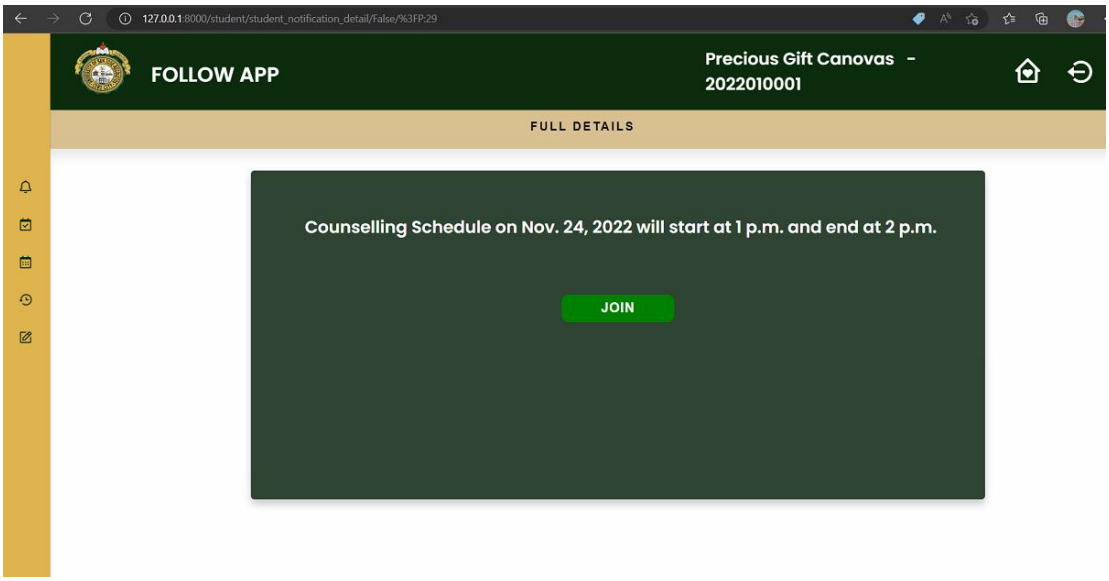

Figure 44: UI Design for Student Joins the Session

Counselor during the meeting.

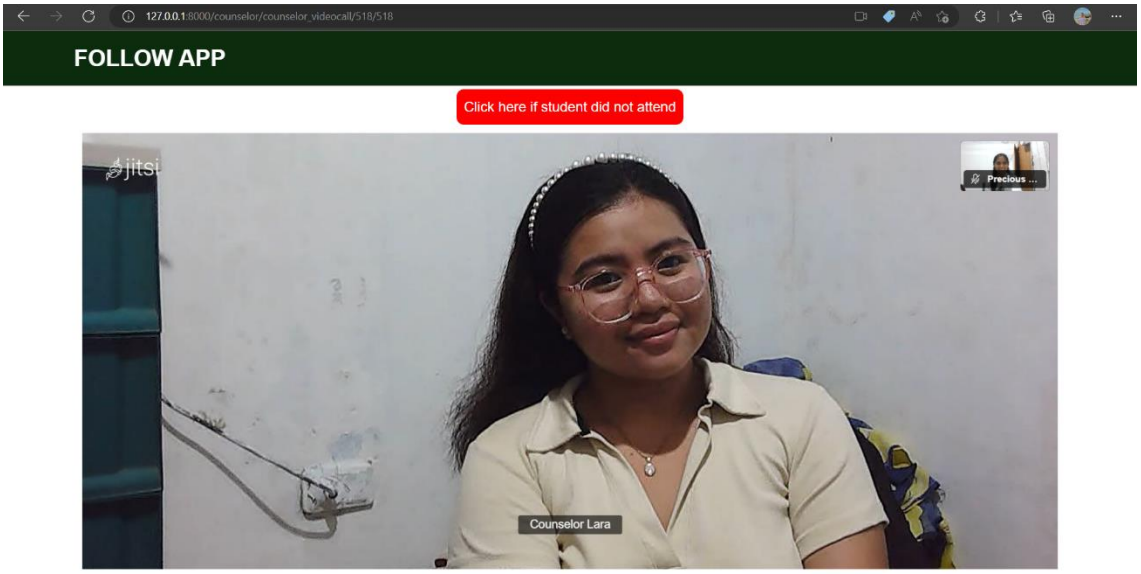

Figure 45: UI Design for Counselor Joins the Call

Student during the meeting.

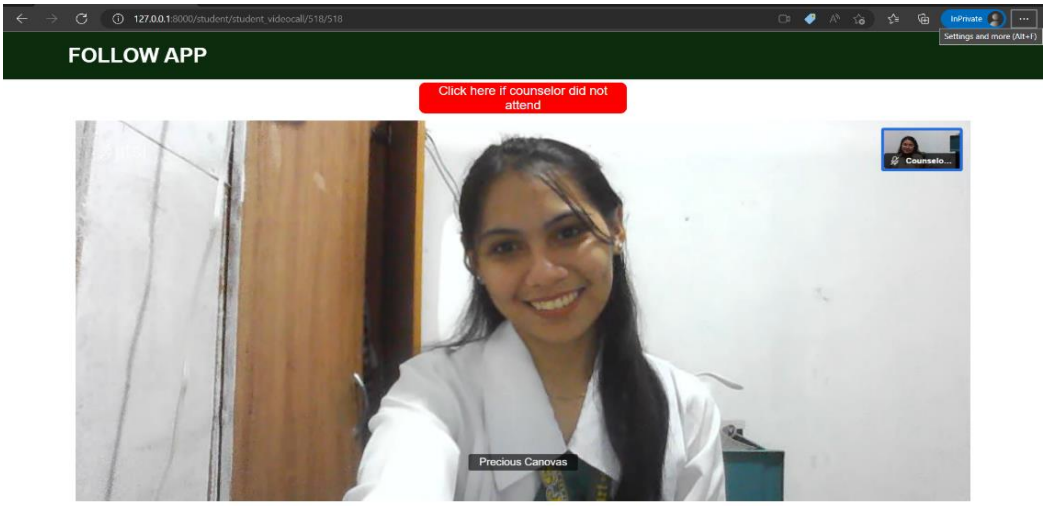

Figure 46: UI Design for Student Joins the Call

Counselor Gives Feedback

Counselor gives Feedback to the counseling session.

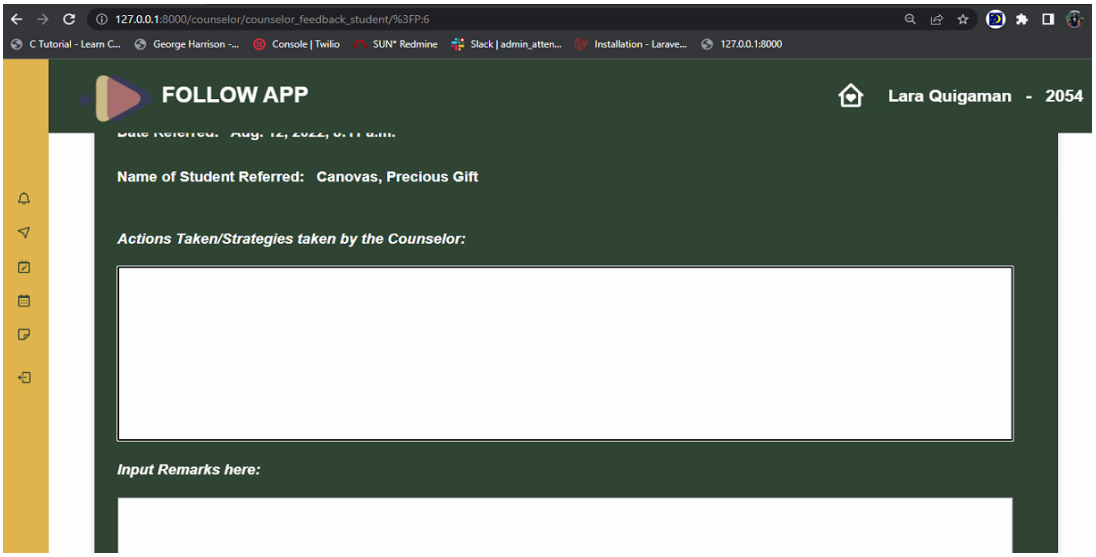

Figure 47: UI Design for Feedback

Teacher Receives the Feedback

Teacher receives the Feedback via notification, that the student is already assessed.

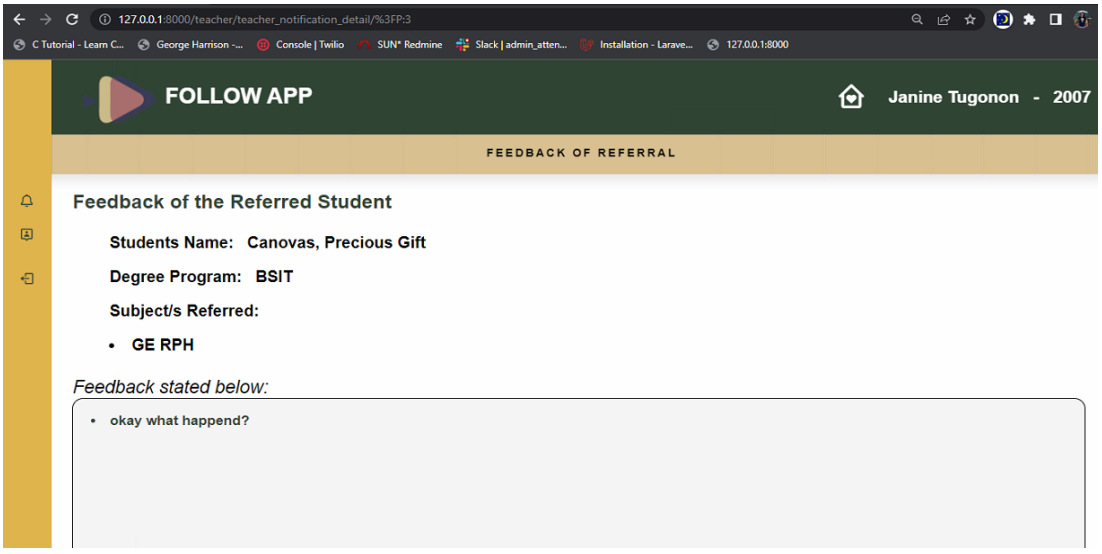

Figure 48: UI Design for Feedback of the Referred Student

Chapter III

Software Development and Testing

This chapter describes the implementation of the project in development, the various tools used to create and run the application. It contains sections for the Development and Testing Process.

DEVELOPMENT SOFTWARE PLATFORMS, DEVELOPMENT ENVIRONMENTS AND PROJECT MANAGEMENT TOOLS

'Follow App' is a web-based application developed in Windows operating system and Visual Studio Code as the tool in web development. Visual Studio Code is a powerful source code editor developed by Microsoft. The programming languages used in software development are Python with Django Framework. The application also uses Jitsi for video calls. Other software development tools and languages are HTML, CSS, GitHub, JavaScript, Twilio, Jitsi for video calls, and PostgreSQL.

The primary language used in the application is Python with Django Framework. Django framework is a free, open-source framework for Python web development written by Python programmers Adrian Holovaty and Simon Willison. Django Framework follows the model-template-views architectural pattern. FollowApp also utilizes Jitsi, a collection of Open-Source projects that offers video conferencing features that are safe, simple to use, and accessible to self-host.

Development Process

Scheduling System with Automatic Date and Time Generation for Counseling Session

The application includes a recommender system that will help counselors schedule student referrals. The system will suggest a date and time for a counseling session based on the available schedule of the student and the counselor. The counselor and student will either accept or decline the schedule. If both of them accept, the suggested schedule by the system will be finalized. If either of them declines, the time and date will be stored in the database as unavailable on the schedule of the user who declined.

```
get_object_counselor_assigned = DegreeProgram.objects.get(
    program_code=degree_program_student_referred)
counselor_assigned_id = get_object_counselor_assigned.faculty_id_id
```

Figure 49: Code Snippet of Getting the Counselor Assigned

Figure 49 shows the process of getting the counselor assigned to the student referred. First, the system gets the degree program of the student. The variable *degree_program_student_referred* holds the degree program of the student using it in finding the name of the counselor.

```
CounselorLoad = SubjectOfferings.objects.filter(
    faculty_id=counselor_assigned_id)
student_referred_load = Studentload.objects. filter(student_number=studentReferredId)
```

Figure 50: Code Snippet of Getting the Load of Counselor and Student

Figure 50 shows the process of getting the load of the counselor and student. All subjects of the counselor and student will be filtered using their id number.

```
for object in CounselorLoad:
    if day_name == 'Thu':
        check = bool(
            day_name[:-1].upper() in object.school_days)
    else:
        check = bool(
            day_name[0].upper() in object.school_days)
    if(check == True):
        classes_counselor.append(SubjectOfferings(offer_no=object.offer_no, subject_code=object.subject_code,
                                                  subject_title=object.subject_title, school_days=object.school_days,
                                                  school_time=object.school_time, sem_id=object.sem_id, academic_year=object.academic_year,
                                                  department_code=object.department_code, faculty_id=object.faculty_id))
for object in student_referred_load:
    sched= SubjectOfferings.objects.get(offer_no=object.offer_no.offer_no)
    if day_name == 'Thu':
        check = bool(
            day_name[:-1].upper() in sched.school_days)
    else:
        check = bool(
            day_name[0].upper() in sched.school_days)
    if(check == True):
        classes_student.append(SubjectOfferings(offer_no=sched.offer_no, subject_code=sched.subject_code,
                                                  subject_title=sched.subject_title, school_days=sched.school_days,
                                                  school_time=sched.school_time, sem_id=sched.sem_id, academic_year=sched.academic_year,
                                                  department_code=sched.department_code, faculty_id=sched.faculty_id))
```

Figure 51: Code Snippet of Getting the Classes of Counselor and Student on the Suggested Day of Counseling Session

Figure 51 gets the classes of the counselor and student on the suggested day for the counseling session by the system. The time of the subjects will be used to look for the available schedule of the counselor and student.

```
referral_list_byday = Referral.objects.filter(
    date=tomorrow, counselor_id=old_get_referral.counselor_id)
```

Figure 52: Code Snippet of Getting the Referral Schedule of the Counselor

Figure 52 shows the process of checking the referral schedule of the counselor on the suggested day of the system. The result of this will be used for checking the availability of the counselor.

```
notAvailableSched = SetScheduleCounselor.objects.filter(
    date=tomorrow, faculty_id=old_get_referral.counselor_id)
notAvailableStudentSched = NotAvailableStudent.objects.filter(
    date=tomorrow, student_number= old_get_referral.student_number)
```

Figure 53: Code Snippet of Getting the Unavailable Schedule of Counselor and Student

The figure above gets the unavailable schedule of the counselor and student on the day of the suggested day of the system for a counseling session. These unavailable schedules will be used to find the availability of both the student and counselor.

```
for x in range(len(timeArray)):
        if(timeArray[x] >= start and timeArray[x] < end):
            #first if -- checking the classes of counselor
            if(classes_counselor_check == True):
                    for object in classes_counselor:
                        get_time = object.school_time
                        classes_counselor_time = get_time.split(
                            '-')
                        classes_counselor_start_time = classes_counselor_time[0].upper(
                        ).replace(" ", "")
                        classes_counselor_end_time = classes_counselor_time[1].upper(
                        ).replace(" ", "")
                        cc_start =get_start_time(classes_counselor_start_time)
                        cc_end =get_end_time(classes_counselor_end_time)
                        cc_start_convert = datetime.strptime(
                            cc_start, '%H:%M').time()
                        cc_end_convert = datetime.strptime(
                            cc_end, '%H:%M').time()
                        if(timeArray[x+1] <= cc_end_convert and timeArray[x] >= cc_start_convert):
                            TimeTaken += 1
```

Figure 54: Checking the Classes of Counselor

Figure 54 shows the first process of checking the availability of the counselor and student. The system will go through all the classes, and if the suggested schedule has the same time as the classes of the counselor, a flag name “TimeTaken” will be incremented, and this will be used for checking the schedule.

```
            if(classes_student_check== True and TimeTaken == 0):
                        for object2 in classes_student:
                            get_time = object2.school_time
                            classes_student_time = get_time.split(
                                '-')
                            classes_student_start_time = classes_student_time[0].upper(
                            ).replace(" ", "")
                            classes_student_end_time = classes_student_time[1].upper(
                            ).replace(" ", "")
                            cs_start =get_start_time(classes_student_start_time)
                            cs_end =get_end_time(classes_student_end_time)
                            cs_start_convert = datetime.strptime(
                                cs_start, '%H:%M').time()
                            cs_end_convert = datetime.strptime(
                                cs_end, '%H:%M').time()
                            if(timeArray[x+1] <= cs_end_convert and timeArray[x] >= cs_start_convert):
                                TimeTaken += 1
```

Figure 55: Checking the Classes of Student

After checking all the classes of the counselor, and “TimeTaken” is still zero or not incremented, it means that the recommended schedule does not have the same time as the classes of a counselor. To this, the process continues to check the classes of the student. The method of checking is still the same with the counselor, and if there is the same class with the student, the “TimeTaken” will be incremented.

```
# 3rd if-- checking the referrals of counselors
if(referral_list_byday_check== True and TimeTaken == 0):
    for object3 in referral_list_byday:
        if(timeArray[x+1] <= object3.end_time and timeArray[x] >= object3.start_time):
            TimeTaken += 1
```

Figure 56: Checking the Counseling Schedule of Counselor

Figure 56 shows the following process for checking the availability of the counselor and student. Unavailable schedules of both the student and counselor will be checked, and if the recommended time has the same time as those counseling schedules of the specific counselor, “TimeTaken” will be incremented.

```
#3rd if -- checking the schedule of counselor which are not available
if(notAvailableSchedChecker== True and TimeTaken==0):
    for object4 in notAvailableSched:
        if(timeArray[x+1] <= object4.end_time and timeArray[x] >= object4.start_time):
            TimeTaken +=1
#4th if -- checking the schedule of student which are not available
if(notAvailableStudentSchedChecker == True and TimeTaken == 0):
    for object5 in notAvailableStudentSchedCheck:
        if(timeArray[x+1] <= object5.end_time and timeArray[x] >= object5.start_time):
            TimeTaken +=1
```

Figure 57: Checking the Unavailable Schedule of the Counselor and Student

After checking the classes of the student, counselor, and the counseling schedule of the counselor, if the recommended time still does not have the same time as those schedules, it will proceed to the last part of the checking, which is going through the unavailable schedule of both the counselor and student. The process of checking the unavailable schedule is still the same as the other.

```
if(TimeTaken == 0 and counter == 0):
        time1 = timeArray[x] #the start time of the counseling schedule
        counter = 1
        TimeTaken = 0
elif(TimeTaken == 0 and counter == 1):
        time2 = timeArray[x+1] #the end time of the counseling schedule
        finder = 1
        counter = 0
        TimeTaken = 0
        break
elif TimeTaken != 0:
        time1 = ''
        TimeTaken = 0
        counter = 0
```

Figure 58: Start Time and End Time Checking

Figure 58 shows the final checking of the availability of the counselor and student. The first condition is checking the first thirty minutes or the start time of the counseling session. The second condition is for the end time of the counseling session. And the last condition is for checking the “TimeTaken”. If the “TimeTaken” is still not incremented, meaning the recommended time does not have the same time with the classes of student and counselor, counseling schedule, and the unavailable schedule of counselor and student, the recommended schedule will be finalized.

Declining of Recommended Schedule by Student with Automatic Contacting of Parents trough SMS.

```python
@login_required(login_url='login')
def accept_or_not(request, id, status, user):
    get_notification = Notification.objects.get(id=id)
    referral = Referral.objects.get(id=id)
    if user == 'counselor':
        if status == 'accept':
            get_notification.is_approve_counselor = True
            get_notification.is_read_counselor = True
        else:
            SetScheduleCounselor.objects.create(faculty_id =referral.counselor_id,
            date=referral.date, start_time=referral.start_time, end_time=referral.end_time,choice='Not Available')
            get_notification.is_approve_counselor = False
    else:
        if status == 'accept':
            get_notification.is_approve_student = True
            get_notification.is_read_student = True
        else:
            NotAvailableStudent.objects.create(student_number=referral.student_number,
            date=referral.date, start_time=referral.start_time, end_time=referral.end_time,choice='Not Available')
            get_notification.is_approve_student = False
            get_notification.count_decline_student = int(get_notification.count_decline_student) + int(1)
            if get_notification.count_decline_student == 3:
                get_student_details = StudentAdditionalInformation.objects.get(student_number=referral.student_number)
                account_sid = 'AC0b6a57cf4cd8bef2d5400ec25ed496fd'
                auth_token = '82b3b98dc7393bc61182c6bbce950fe5'
                client = Client(account_sid, auth_token)
                body = "Good day parents! I would like to meet your son/daughter for a counselling sessions, but he/she always declines our schedule."
                message = client.messages.create(
                to=get_student_details.mother_contact_number, from_='+15139606294', body=body)
                print(message.sid)
                message = client.messages.create(
                to=get_student_details.father_contact_number, from_='+15139606294', body=body)
                print(message.sid)
                message = client.messages.create(
                to=get_student_details.guardian_contact_number, from_='+15139606294', body=body)
                print(message.sid)
```

Figure 59: Accept or Decline Recommended Counseling Schedule

Figure 59 shows how the student and counselor accept the recommended schedule. The first condition is to check if the user is a “counselor” or “student.” After checking, it will proceed to the status where it has the value of “accept” or “not.” If the user accepts the recommended schedule, it will be the final schedule; otherwise, it will set a not available one depending on the user who declined the schedule. If the number of student declines reaches three, the system will automatically send a message to the parents regarding this matter.

Student No Show on Counseling Session with Automatic Contacting of Parent through SMS

```
if contact_parents == 3:
    get_student_details = StudentAdditionalInformation.objects.get(student_number=old_get_referral.student_number)
    account_sid = 'AC0b6a57cf4cd8bef2d5400ec25ed496fd'
    auth_token = '82b3b98dc7393bc61182c6bbce950fe5'
    client = Client(account_sid, auth_token)
    body = 'Good day parents! I would like to inform you that your child did not attend 2 consecutive counselling sessions. Kindly inform your child to visit the of
    message = client.messages.create(
    to=get_student_details.mother_contact_number, from_='+15139606294', body=body)
    print(message.sid)
    message = client.messages.create(
    to=get_student_details.father_contact_number, from_='+15139606294', body=body)
    print(message.sid)
    message = client.messages.create(
    to=get_student_details.guardian_contact_number, from_='+15139606294', body=body)
    print(message.sid)
    messages.info(request, 'Parents are already informed!')
    contact_parents=0
    get_notif.is_read_student = True
    get_notif.is_read_counselor = True
    get_notif.is_approve_counselor = True
    get_notif.is_approve_student = True
    get_notif.is_counseled = True
    get_notif.save()
    return redirect('counselor_home_view')
```

Figure 60: Code Snippet of Student No Show and Contacting of Parents

Figure 60 shows the process when the student did not join the meeting for three consecutive sessions. If the student reaches several no-shows, the student's parent will be automatically contacted through SMS.

Automatic Referral of Student

The system also includes automatic referrals of students. The system will assess and find at-risk students to be referred to their respective counselors. The data to be considered are data from the SIS (School Information Service) and LMS (Learning Management System). The data from SIS includes grades from all the subjects the student is enrolled in that semester, while the data from LMS includes the number of absences, the content completed and content required, assignment completed and assignment required, quiz completed, and quiz required of each student in each subject.

```
for data in all_edp:
    edp_data = EDP.objects.filter(student_number=data,sem_id=Active_Sem,term=term,academic_year=Active_Year)
    for obj in edp_data:
        if obj.grades >= 3.0 or obj.grades >= 3:
            data_to_show.append({'student_number':  obj.student_number.student_number, 'offer_no': obj.offer_no.offer_no,'sem_id': obj.sem_id,
            'academic_year': obj.academic_year, 'term': obj.term})
            sched = automation_reffer(obj.student_number.student_number,obj.offer_no.offer_no,obj.sem_id,
            'Student is referred for having a grade of equal or greater than 3.0' )
        else:
            lms = LMS.objects.get(student_number=data,sem_id=Active_Sem,term=term,academic_year=Active_Year, offer_no=obj.offer_no)
            if lms.absences >= 3:
                data_to_show.append({'student_number':  lms.student_number.student_number, 'offer_no': lms.offer_no.offer_no,'sem_id': lms.sem_id,
                'academic_year': lms.academic_year, 'term': lms.term})
                sched = automation_reffer(lms.student_number.student_number,lms.offer_no.offer_no,lms.sem_id,
                'Student is referred because of having 3 or more absences')
            elif lms.total <= 50.0 or lms.total <= 50:
                data_to_show.append({'student_number':  lms.student_number.student_number, 'offer_no': lms.offer_no.offer_no,'sem_id': lms.sem_id,
                'academic_year': lms.academic_year, 'term': lms.term})
                sched = automation_reffer(lms.student_number.student_number,lms.offer_no.offer_no,lms.sem_id,
                'Student is referred because of his/her academic performance' )
```

Figure 61: Automatic Referral of Student Based on Grade and Absences

If the student has a grade of 3.0 or above in any subject, he will be automatically referred. Likewise, if a specific student accumulates three or more absences, he will be automatically referred.

```
check_day = SubjectOfferings.objects.get(offer_no=data[1],sem_id =data[2])
if check_day.school_days == 'MWF':
    limit = 9
else:
    limit = 7
weight_absences = round((1-(data[5]/limit))*100*absences,2)
weight_content =  round((data[6]/data[7])*100*content,2)
weight_assignment = round((data[8]/data[9])*100*assignment,2)
weight_quiz =  round((data[10]/data[11])*100*quiz,2)
```

Figure 62: Calculating the Weight of the Student's Data from LMS

Figure 62 shows how the weight in each column was computed. The weight signifies the impact on the referral of the student. The columns are computed using the following formulas.

Absence Weight = (1- (Number of Absences /Maximum Number of Absences) 100 * Impact Percentage*

*Content Weight = (Content Completed / Content Required) *100 * Impact Percentage*

*Assignment Weight = Assignment Completed/ Assignment Required * 100 * Impact Percentage*

Quiz Weight = (Quiz Completed / Quiz Required) 100 * Impact Percentage*

```
lms = LMS.objects.get(student_number=data,sem_id=Active_Sem,term=term,academic_year=Active_Year, offer_no=obj.offer_no)
if lms.total <= 50.0 or lms.total <= 50:
    data_to_show.append({'student_number':  lms.student_number.student_number, 'offer_no': lms.offer_no.offer_no,'sem_id': lms.sem_id,
    'academic_year': lms.academic_year, 'term': lms.term})
    sched = automation_reffer(lms.student_number.student_number,lms.offer_no.offer_no,lms.sem_id,
    'Student is referred because of his/her academic performance' )
```

Figure 63: Automatic Referral of Student Based on Student's Data from LMS

If the total computed weights of the columns of the student are lower than 50%, they will be automatically referred. The table shows a simulation to see when and how the student is referred to visually.

Data from SIS

Student No. 1

Subject	Grade
Subject No. 1	1.3
Subject No. 2	3.4
Subject No. 3	2.0

Table 15: Grade of Student No. 1 from SIS

Status: **Referred**

Table 15 shows the grades of student 1 in all subjects. It shows that Student1 has a grade of 3.0 above which means he will be automatically referred.

Student No. 2

Subject	Grade
Subject No. 1	1.3
Subject No. 2	2.5
Subject No. 3	2.0

Table 16: Grade of Student No. 2 from SIS

Status: **Not Referred**

Table 16 shows all the grades of Student No. 2 in all subjects. It shows no grade above 3.0 which means the student will not be referred.

Data from LMS Computation

The data from the LMS will be queried. The weight from each column will be computed. The weight signifies the impact of the referral. The higher the percentage, the more significant impact it has on the student's referral. The table below shows how the percentage was achieved in automatically referring the student.

Student No. 1

Absences (30 %)		
No. of Absences	**Maximum No. of Absences**	**Weight**
(1-(No. of Absences/ Maximum No. of Absences))*100*0.30		
6	9	9.9 %

Table 17: Computing the Absences Weight of Student No. 1

Table 17 shows the computation of the weight of the absences using the given formula and sample data.

Content (15 %)		
Content Completed	**Content Required**	**Weight**
(Content Completed / Content Required)*100*0.15		
5	10	7.5%

Table 18: Computing the Content Weight of Student No.1

Table 18 shows the computation of the weight of content using the given formula and sample data.

Assignment (15 %)		
Assignment Completed	**Assignment Required**	**Weight**
(Assignment Completed / Assignment Required)*100*0.15		
10	20	7.5%

Table 19: Computing the Assignment Weight of Student No. 1

Table 19 shows the computation of the weight of assignment using the given formula and sample data.

Quiz (40 %)		
Quiz Completed	**Quiz Required**	**Weight**
(Quiz Completed / Quiz Required)*100*0.40		
2	10	8 %

Table 20: Computing the Quiz Weight of Student No. 1

Table 20 shows the computation of the weight of assignment using the given formula and sample data.

Total= Weight of Absences+ Weight of Content + Weight of Assignment + Weight of Quiz

Total = 9.9 % + 7.5 % +7.5 % + 8% = 32.99% (less than 50 %) Referred

Since the threshold is 50% and student No. 1 did not reach 50%, the student is referred directly to the counselor assigned on the degree program of the student.

Student No. 2

Absences (30 %)		
No. of Absences	**Maximum No. of Absences**	**Weight**
(No. of Absences/ Maximum No. of Absences)*100*0.30		
1	9	26.7 %

Table 21: Computing the Absences Weight of Student No. 1

Table 21 shows the computation of the weight of absences of Student No. 2 using the given formula and sample data.

Content (15 %)		
Content Completed	**Content Required**	**Weight**
(Content Completed / Content Required)*100*0.15		

9	10	13.5%

Table 22: Computing the Absences Weight of Student No. 1

Table 22 shows the computation of the weight of content of Student No. 2 using the given formula and sample data.

Assignment (15 %)		
Assignment Completed	**Assignment Required**	**Weight**
(Assignment Completed / Assignment Required)*100*0.15		
15	20	11.25%

Table 23: Computing the Absences Weight of Student No. 1

Table 23 shows the computation of the weight of assignment of Student No. 2 using the given formula and sample data.

Quiz (40 %)		
Quiz Completed	**Quiz Required**	**Weight**
(Quiz Completed / Quiz Required)*100*0.40		
8	10	32 %

Table 24: Computing the Absences Weight of Student No. 1

Table 234 shows the computation of the weight of quiz of Student No. 2 using the given formula and sample data.

Total= Weight of Absences+ Weight of Content + Weight of Assignment + Weight of Quiz

Total = 26.7 % + 13.5 % +11.25 % + 32% = 83.45% (greater than 50 %) Not Referred

Since the threshold is 50% and student No. 2 reached 50% and above, the student is not referred.

TESTING PROCESS

Development Testing

The developers testing segment depicts that all the features of the application are working correctly.

Test Cases

TEST CASE 1: ADMIN

Test Module	Test Scenario	Expected Result	Actual Result	Status
Upload Data	Admin uploads the data with correct semester and year	Must display the data being uploaded	Data are displayed and are stored in the database	**PASSED**
View Data	Admin will view the data that are uploaded	He can choose what data to be viewed and display	System displays the data selected	**PASSED**

Table 25: Test Case of Admin

This testing process shows where the Admin uploads data to populate the database and also view the data that is uploaded.

TEST CASE 2: DIRECTOR

Test Module	Test Scenario	Expected Result	Actual Result	Status
Assigns Counselor	Director will assign the counselor for each degree program	Degree program will have assigned counselor	Degree program have assigned counselor	**PASSED**
View Statistics	Views Statistics by department	Display total number of students counseled by department	Display total number of students counseled by department	**PASSED**

	Views statistics by counselors	Display total number of students counseled by counselor	Display total number of students counseled by counselor	**PASSED**

Table 26: Test Cases of Director

This testing process shows how the Director assigns a Counselor per degree program, how to view the statistics per degree program and counselor, and set a filter on what data is to be displayed.

TEST CASE 3: TEACHER

Test Module	**Test Scenario**	**Expected Result**	**Actual Result**	**Status**
Refer Student	Input the information about the student being referred	A recommended schedule will be suggested to the counselor and if both accepted by the student and counselor, the referral will be stored to the database.	A schedule was recommended to the counselor and student; and if both confirm, the referral and the schedule will be stored in the database.	**PASSED**
	Fields with (*) are not filled in by the teacher	A message “Please fill out this field” will be displayed	A message “Please fill out this field” showed.	**PASSED**
Receives Feedback	A feedback should be sent by the counselor	A feedback from the counselor on the referred student will display.	A feedback from the counselor on the referred student is displayed	**PASSED**

Table 27: Test Cases of Teacher

This testing process shows the Teacher manually referring a student because of having a behavioral problem and after the counseling session of the student and counselor. The Teacher will then receive a notification of the feedback.

TEST CASE 4: COUNSELOR

Test Module	**Test Scenario**	**Expected Result**	**Actual Result**	**Status**
Accept and Decline Schedule	Accepts recommended schedule	Navigate to home page	Navigate to home page	**PASSED**
	Declines recommended schedule	Navigate to home page and will notify another schedule	Navigate to home page and will notify another schedule	**PASSED**
Join Video meeting	Counselor joins the meeting	Video call UI will display then counselor will join the meeting	Video call UI will display then counselor will join the meeting	**PASSED**
	Student did not join the meeting	Counselor will click the button , “Click here if student/counselor is not available” if the student did not join the meeting and will notify for the new schedule	Counselor will click the button if the student did not join the meeting and will notify for the new schedule	**PASSED**
Give Feedback	Counselor gives feedback after the counseling session	Teacher will be notified and receives the feedback of the counselor	Teacher will be notified and receives the feedback of the counselor	**PASSED**

Table 28: Test Cases of Counselor

This testing process shows how the Counselor accepts and declines the recommended schedule, how the Counselor joins the meeting for the counseling session and can click

a button to reschedule if the student does not attend, and the process of giving feedback after their session.

TEST CASE 5: STUDENT

Test Module	**Test Scenario**	**Expected Result**	**Actual Result**	**Status**
Sets Appointment	Student enters the information in the form.	Done button will enable after inputting all the details	Done button will enable after inputting all the details	**PASSED**
	Incomplete form	Done button is disabled if details are incomplete	Done button is disabled if details are incomplete	**PASSED**
Join Video Meeting	Student joins the video meeting	Video call UI will display then student will join the meeting	Video call UI will display then student will join the meeting	**PASSED**
Accept and Decline Appointment	Accepts recommended schedule	Navigate to home page	Navigate to home page	**PASSED**
	Declines Recommended Schedule	Navigate to home page and will notify another schedule	Navigate to home page and will notify another schedule	**PASSED**

Table 29: Test Cases of Student

This testing process shows how the Student accepts and declines the recommended schedule, how the Student joins the meeting for the counseling session and can click a button to reschedule if the counselor does not attend, and the process of setting a schedule for counseling.

Usability Testing

This test is intended to identify and evaluate the user experience while utilizing the app in its completed form. This test includes all functions of the app, including implemented system features. The table below shows the satisfaction rating of 20 users who have used the application. The ratings used in the survey were as follows:

5–Strongly Agree, 4–Agree, 3–Neutral, 2 –Disagree, 1 –Strongly Disagree

Questions	Average Rating by User
The application's features work.	4.13
The application doesn't crash.	3.93
It is easy to navigate between different pages.	4.13
The application was not interrupted when using other apps and vice-versa.	3.93
I was able to complete tasks and scenarios using this system.	4.2
The system gave error messages that clearly told me how to fix problems.	4.067
The interface of this system is pleasant.	4.067
It was easy to use this system.	4.2
How likely are you to recommend the Follow App application to a friend	4.2

Table 30: Result of Usability Testing

Summary of Testing

Upon undergoing the testing process stated above, the researchers' project objectives were achieved. The features of the application were evaluated, and they have passed the testing phase. To attain a positive result, the researchers evaluated using different test cases. With all these, the system requirements were fully realized.

CHAPTER IV

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

This chapter presents the summary of the study, conclusions, drawn from the findings, and the recommendations offered by the researchers.

Summary of Findings

In manual referral, the system was able to match the availability of both counselor and student. It is found that the time complexity is just enough, given the referral is scheduled one at a time. However, during the automatic referral of students, it is found that the larger the data from the LMS and SIS are, the longer the processing time since the system will go through each data from SIS and LMS. The time complexity notation used in the referral part is N2 log N.

The table below shows the number of data being uploaded to the system and the corresponding time it can finish processing the automatic referral.

Number of Rows in the CSV file	Time
200	6 seconds
400	18 seconds
600	42 seconds
800	1 minute
1000	2 minutes

Table 31: Table of the Processing Time for Automatic Referral

In the video call, it is also found out that the developers found it hard to manipulate some functions of the JITSI library prompting developers to create buttons to navigate through the system.

Conclusion

The study has achieved its purpose of designing a web application to help improve the scheduling of referrals for counselors and teachers. The counselor is provided with an easy and hassle-free way of handling student referrals. The recommender system will automatically suggest possible schedules of counseling based on the available schedule of both the student and counselor. The automatic referral system also helps teachers to refer at-risk students based on their data from SIS and LMS. The result shows that Follow App is an effective system that helps counselors and teachers manage their referrals.

Recommendation

To further improve Follow App, future researchers should integrate transcripts on video calls to help counselors remember everything that has been discussed during the call or meeting. It will also serve as a future reference for counselors if in case the same student is referred again.

To also improve the performance of the automatic referral system, future researchers are encouraged to find another algorithm to optimize the processes, or they can restructure the database so that they can query data faster. The researchers can also integrate machine learning to help counselors by giving suggestions depending on the problem of the students after counseling.

APPENDICES

BIBLIOGRAPHY

[1] Megan Hrivnak, Don Coble and Rebekah Byrd, “Cybercounseling: Legal and Ethical Considerations”, Spring 2015, [Online].

Available: https://dc.etsu.edu/cgi/viewcontent.cgi?article=1910&context=etsu-works [Accessed: July 2022]

[2] Tawonda M. Hunter, Liberty University. “Virtual School Counseling Amid The Covid-19 Pandemic”, 2020. [Online]

Available: https://digitalcommons.liberty.edu/cgi/viewcontent.cgi?article=4037&context=doctoral [Accessed: July 2022]

[3] Sharon Evans, “The practicalities and benefits of online counseling”, May 2, 2019. [Online]

Available: https://www.counselling-directory.org.uk/memberarticles/the-practicalities-and-benefits-of-online-counselling [Accessed: July 2022]

[4] Serife Gonca Zeren, Seher Merve Erus, Yagmur Amanvermez, Arzu Buyruk Genc, Mihriban Betul Yilmaz, and Baki Duy, “The Effectiveness of Online Counseling for University Students in Turkey, Aprill 11, 2020. [Online]

Available: https://pdf.eu-jer.com/EU-JER_9_2_825.pdf [Accessed: July 2022]

[5] Introduction | Jitsi Meet [Online]

[Accessed: October 2022]

[6] Console | Twilio [Online]

[Accessed: May 2022]

[7] Django documentation | Django documentation | Django (djangoproject.com) [Online]

[Accessed: June 2021]

[8] PostgreSQL: Documentation [Online]

[Accessed: June 2021]

[9] Django ArrayField - Working with Arrays - ZeroToByte [Online]

[Accessed: July 2021]

[10] django-multiselectfield · PyPI [Online]

[Accessed: July 2021]

[11] Django and Openpyxl: Extracting and Sending Django model data as excel file (.xlsx), [Online].

Available: Django and Openpyxl: Extracting and Sending Django model data as excel file (.xlsx) - DEV Community

[Accessed: August 2021]

[12] User authentication in Django, [Online].

Available: User authentication in Django | Django documentation | Django (djangoproject.com)

[Accessed: July 2021]

[13] Paginator, [Online].

Available: Paginator | Django documentation | Django (djangoproject.com)

[Accessed: June 2022]

[14] Quickstart — Tablib 3.2.1 documentation [Online]

[Accessed: August 2021]

[15] Django Forms, [Online].

Available: Django Forms | How to Create a Django Form with Examples (educba.com)

[Accessed: July 2021]

[16] 2022 Talkspace Review: Features, Benefits, Costs, and More [Online].

Available: Talkspace Review 2022: Features, Benefits, and More (healthline.com)

[Accessed: December 2022]

[17] 18 Best Therapy & Counseling Apps and Therapist Platforms [Online].

Available: 18 Best Therapy & Counseling Apps and Therapist Platforms (quenza.com)

[Accessed: December 2022]

[18] BetterHelp is the world's largest therapy platform [Online].

Available: About Us - The Largest Online Therapy Provider | BetterHelp

[Accessed: December 2022]

[19] Teen Counseling [Online]

Available: Teen Counseling

[Accessed: December 2022]

Curriculum Vitae

PERSONAL INFORMATION

Name	: **Precious Gift A. Canovas**
Nickname	: **Gift**
Date of Birth	: **November 10, 1999**
Address	: **Brgy. 2, San Francisco, Agusan Del Sur**
Status	: **Single**
Citizenship	: **Filipino**
Religion	: **Born Again**
Email Address	: **canovasprecious@gmail.com**
Contact Number	: **09109248762**

ACADEMIC INFORMATION

Tertiary	: **University of San Jose Recoletos**	**(2018 - 2022)**
Secondary	: **Agusan Del Sur National High School**	**(2012 - 2018)**
Primary	: **San Francisco Pilot Elementary School**	**(2007 - 2012)**

PROFESSIONAL SKILLS

Language	: **C, Java, HTML, CSS, JavaScript, Python, Flutter**
Software	: **MS Office, Visual Studio, VSCode, Netbeans**
OS	: **Windows**

PERSONAL INFORMATION

Name	: **Rebecca Lander T. Evangelista**
Nickname	: **Becs**
Date of Birth	: **February 27, 2000**
Address	: **Brgy. Poblacion, Talibon, Bohol**
Status	: **Single**
Citizenship	: **Filipino**
Religion	: **Roman Catholic**
Email Address	: **rebeccalanderevans@gmail.com**
Contact Number	: **09212119350**

ACADEMIC INFORMATION

Tertiary	: **University of San Jose Recoletos**	**(2018 - 2022)**
Secondary	: **Blessed Trinity College**	**(2012 - 2018)**
Primary	: **Talibon Central Elementary School**	**(2007 - 2012)**

PROFESSIONAL SKILLS

Language	: **C, Java, HTML, CSS, JavaScript, Python, Flutter**
Software	: **MS Office, Visual Studio, VSCode, Netbeans**
OS	: **Windows**

PERSONAL INFORMATION

Name	: **Epifanio U. Villaro Jr.**
Nickname	: **Pif**
Date of Birth	: **January 25, 1999**
Address	: **Brgy. Tagnate, Hilongos, Leyte**
Status	: **Single**
Citizenship	: **Filipino**
Religion	: **Roman Catholic**
Email Address	: **villarop@gmail.com**
Contact Number	: **09263044506**

ACADEMIC INFORMATION

Tertiary	: **University of San Jose Recoletos**	**(2018 - 2023)**
Secondary	: **Hilongos National Vocational School**	**(2012 - 2018)**
Primary	: **Tagnate Elementary School**	**(2007 - 2012)**

PROFESSIONAL SKILLS

Language	: **C, Java, HTML, CSS, JavaScript, Python, Flutter**
Software	: **MS Office, Visual Studio, VSCode, Netbeans**
OS	: **Windows**

University of San Jose - Recoletos
Corner P. Lopez & Magallanes Streets, 6000 Cebu City, Philippines
CENTER FOR POLICY, RESEARCH AND DEVELOPMENT STUDIES

CERTIFICATION

The manuscript entitled **"FOLLOW APP: A WEB APPLICATION FOR SCHOOL COUNSELING"** has undergone Similarity tests under Turnitin and Grammarly softwares.

AUTHOR/s: **EPIFANIO U. VILLARO JR.**
PRECIOUS GIFT A. CANOVAS
REBECCA LANDER T. EVANGELISTA

TURNITIN RESULT**: 8%**

GRAMMARLY RESULT: **91/100**

This is to certify further that the manuscript has registered an originality grade of **92%** and technical writing quality of **91%** which includes grammar, spelling, and punctuations, among others. Given this 9th day of December, 2022 at the Quality Assurance Unit of the Center for Policy, Research and Development Studies, University of San Jose-Recoletos, Cebu City.

MR. ENGLEVERT P. REYES, LPT, M.Ed, PhD(c)
CPRDS Director