



Faculty of Computers and Information Technology (FCIT)

**A Prediction Model-Based Application to forecast the Attainment of
the Programs' Learning Outcomes (PLOs) at the University of
Tabuk (UoT)**

Author:

Saleh Mohammed Alanazi SID: 421004033

Mousa Mohammed Bariqi SID: 421002998

Sultan Yahya Alawaji SID: 381013469

Esam Muftin Alotaibi SID: 381008495

Supervisor: Dr. Awad M. Awadelkarim

Submitted in partial fulfilment of the requirements for the Degree of Bachelor of Information Technology

Academic Year: 2023-2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at University of Tabuk or other institutions.

Name	ID No.	Signature
Saleh Mohammed Alanazi	421004033	
Mousa Mohammed Bariqi	421002998	
Sultan Yahya Alawaji	381013469	
Esam Muftin Alotaibi	381008495	

Date : 10/05/2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled "**A Prediction Model-Based Application to forecast the Attainment of the Programs' Learning Outcomes (PLOs) at the University of Tabuk (UoT)**" was prepared by **Saleh Mohammed Alanazi, Mousa Mohammed Bariqi, Sultan Yahya Alawaji, Esam Mavtn Alotaibi**, has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Information Technology at University of Tabuk (UoT).

Approved by,

Signature : _____

Supervisor: **Dr. Awad M. Awadelkarim**

Date : _____

The copyright of this report belongs to the author and is protected under the intellectual property right laws and conventions. It can only be considered/used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University of Tabuk.

© 2023-2024, A Prediction Model-Based Applications (PLOs). All right reserved.

Specially dedicated to
my beloved grandmother, mother and father

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, **Dr. Awad M. Awadelkarim** for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement.

A Prediction Model-Based Application to forecast the Attainment of the Programs' Learning Outcomes (PLOs) at the University of Tabuk (UoT)

ABSTRACT

Enhancing the academic achievements of university students represents a crucial and fundamental objective. Educational establishments diligently endeavor to equip their graduates with the requisite qualifications for the labor market. This project aims to develop a web-based predictive model for evaluating the Program Learning Outcomes (PLOs) to forecast the Attainment at University of Tabuk (UoT). The proposed system utilizes predictive analytics to identify students at risk of poor performance in achieving learning outcomes (academic hurdles). The project aims to identify students at risk of academic challenges at an early stage. The web-based system development follows the Waterfall methodology, and the prediction model builds using the standard methodology known for building predictive models in data science and artificial intelligence. Provided that these achievements should work to increase and enhance the academic achievement via the attainment of Program Learning Outcomes (PLOs) at University of Tabuk (UoT), and accordingly graduates are more suitable for the labor market.

TABLE OF CONTENTS

DECLARATION	ii
APPROVAL FOR SUBMISSION	iii
ACKNOWLEDGEMENTS	vi
ABSTRACT	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	x
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xix

CHAPTER

1	INTRODUCTION	1
1.1	Background	1
1.2	Problem overview	2
1.3	Aims and Objectives	2
1.4	The Importance of the Project	2
1.5	The Scope of the Project	3
1.6	Report's Layout	3
1.7	Project Methodology	4
1.7.1	Prediction model	4
1.7.2	Waterfall Model	4
2	THE REQUIREMENTS COLLECTION AND SYSTEM ANALYSIS	6
2.1	Introduction	6
2.2	Background	6

2.2.1	Program Learning Outcome's (PLOs)	6
2.2.2	Predictive Models	8
2.3	Related Works	8
2.3.1	Course Signals at Purdue University	8
2.3.2	Grade Predictor	11
2.3.3	UG Partner	12
2.4	System Description	13
2.5	The System Requirements	13
2.5.1	Functional Requirements of the System	14
2.5.2	Non- Functional Requirements of the System	16
2.6	System Analysis	18
2.6.1	Use Case Diagrams	18
2.6.2	Use Case Scenario	23
2.6.3	Activity Diagram	47
2.6.4	Sequence Diagram	72
2.7	Summary	88
3	DATABASE DESIGN	89
3.1	Introduction	89
3.2	Entity Relationship Diagram (ERD)	89
3.2.1	Identify entities	89
3.2.1	Determine Entity Relationships, Multiplicities, and Verify	92
3.3	logical Relational Model	103
3.4	Summary	109
4	THE SYSTEM IMPLEMENTATION AND TESTING	110
4.1	Introduction	110
4.2	Prediction Model	110
4.2.1	Program-level prediction	110
4.2.2	Subject-level prediction	113
4.3	System User Interface	117
4.4	Summary	132

5	CONCLUSION AND RECOMMENDATIONS	133
5.1	Conclusion	133
5.2	Recommendations	133
5.3	Difficulties	133
REFERENCES		135
APPENDICES		136

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Log in (User Scenario)	23
2.2	Manage User Profile (User Scenario)	23
2.3	Sign out (User Scenario)	24
2.4	Manage the basic system data (Admin Scenario)	25
2.5	Create reports (Admin Scenario)	27
2.6	Manage system data (Admin Scenario)	28
2.7	Monitor and generate the reports (President of the University Scenario)	30
2.8	Monitor and generate the reports (Dean of the College Scenario)	31

2.9	Manage program basic information (Program Coordinator Scenario)	32
2.10	Manage program learning outcomes (Program Coordinator Scenario)	34
2.11	Receiving reports (Program Coordinator Scenario)	36
2.12	Generate reports (Program Coordinator Scenario)	37
2.13	Monitor of students (Academic advisor Scenario)	38
2.14	View student profiles (Academic advisor Scenario)	38
2.15	Communicate with students (Academic advisor Scenario)	39
2.16	Identify students at risk (Academic advisor Scenario)	40
2.17	Generate reports (Academic advisor Scenario)	40
2.18	Monitor of students (Faculty Member Scenario)	41
2.19	Manage grades (Faculty Member Scenario)	42
2.20	Communicate with students (Faculty Member Scenario)	43
2.21	Identify students at risk (Faculty Member Scenario)	44
2.22	View subject information (Student Scenario)	44
2.23	view the predicted academic performance (Student Scenario)	45
2.24	View previous academic performance (Student Scenario)	45
2.25	Identify subjects requiring improvement (Student Scenario)	46
2.26	Communicate with the Academic Advisor or faculty member (Student Scenario)	47

LIST OF FIGURES

FIGURE	TITLE	PAGE
1.1	Waterfall Model	5
2.1	Course Signals	10
2.2	Grade Predictor	11
2.3	UG Partner	12
2.4	User use case diagram	18
2.5	Admin use case diagram	19
2.6	President of university use case diagram	19
2.7	Vice President for Academic Affairs use case diagram	20
2.8	Dean of college use case diagram	20
2.9	Program Coordinator use case diagram	21
2.10	Academic Advisor use case diagram	21
2.11	Faculty Member use case diagram	22
2.12	Student use case diagram	22
2.13	Login activity diagram	48
2.14	manage user profile activity diagram	49
2.15	sign out activity diagram	50
2.16	manage members accounts activity diagram	51
2.17	Create reports activity diagram	52
2.18	Manage system data activity diagram	53

2.19	Monitor and generate the reports activity diagram	54
2.20	Monitor and generate the reports activity diagram	55
2.21	Manage program basic information activity diagram	56
2.22	Manage program learning outcomes activity diagram	57
2.23	Receiving reports activity diagram	58
2.24	Generate reports activity diagram	59
2.25	Monitor of students activity diagram	60
2.26	View student profiles activity diagram	60
2.27	Communicate with students activity diagram	61
2.28	Identify students at risk activity diagram	62
2.29	Generate reports activity diagram	63
2.30	Monitor of students activity diagram	64
2.31	Manage grades activity diagram	65
2.32	Communicate with students activity diagram	66
2.33	Identify students at risk activity diagram	67
2.34	view subject information activity diagram	67
2.35	view the predicted academic performance activity diagram	68
2.36	View previous academic performance activity diagram	69
2.37	Identify subjects requiring improvement activity diagram	70
2.38	Communicate with the Academic Advisor or faculty member activity diagram.	71
2.39	Log in Sequence diagram	72

2.40	Manage User Profile Sequence diagram	72
2.41	Sign out Sequence diagram	73
2.42	Manage members accounts Sequence diagram	74
2.43	Create reports Sequence diagram	75
2.44	Manage the basic system data Sequence diagram	76
2.45	Monitor and generate reports Sequence diagram	77
2.46	Monitor and generate reports Sequence diagram	78
2.47	Manage program basic information Sequence diagram	79
2.48	Manage program learning outcomes Sequence diagram	80
2.49	Receiving reports Sequence diagram	81
2.50	Generate reports Sequence diagram	81
2.51	Monitor of students Sequence diagram	81
2.52	View student profiles Sequence diagram	82
2.53	Communicate with students Sequence diagram	82
2.54	Identify students at risk Sequence diagram	82
2.55	Generate reports Sequence diagram	83
2.56	Monitor of students Sequence diagram	83
2.57	Manage grades Sequence diagram	84
2.58	Communicate with students Sequence diagram	85
2.59	Identify students at risk Sequence diagram	85
2.60	view subject information Sequence diagram	86
2.61	view the predicted academic performance Sequence diagram	86

2.62	View previous academic performance Sequence diagram	87
2.63	Identify subjects requiring improvement Sequence diagram	87
2.64	Communicate with the Academic Advisor or faculty member Sequence diagram	88
3.1	Admin Entity	90
3.2	Account Entity	90
3.3	Report Entity	90
3.4	President_of_University Entity	90
3.5	Vice_President Entity	90
3.6	Dean_of_College Entity	90
3.7	College Entity	91
3.8	Department Entity	91
3.9	Program Entity	91
3.10	Faculty_Member Entity	91
3.11	Program_Coordinator Entity	91
3.12	Academic_Advisor Entity	91
3.13	Subject Entity	92
3.14	Student Entity	92
3.15	Coursework Entity	92
3.16	Admin and Account Relationship	93
3.17	Admin and Report Relationship	94
3.18	President_of_University, Faculty_Member and Report Relationship	94
3.19	Vice_President, Faculty Member and Report Relationship	95

3.20	Dean_of_College, Faculty_Member and Report Relationship	95
3.21	College and Department Relationship	96
3.22	Department and Program Relationship	97
3.23	Program_Coordinator, Faculty Member, Program and Report Relationship	98
3.24	Academic_Advisor, Faculty Member and Report Relationship	99
3.25	Faculty_Member and Account Relationship	99
3.26	Program and Subject Relationship	100
3.27	Student, Account, Program and Academic_Advisor relationship	101
3.28	Faculty_Member, Coursework , Student and Subject Relationship	102
3.29	ERD	103
3.30	Admins Table	104
3.31	Accounts Table	104
3.32	Position Table	104
3.33	Status Table	104
3.34	Colleges Table	104
3.35	Departments Table	105
3.36	Programs Table	105
3.37	Subjects Table	105
3.38	Current semester Table	105
3.39	Coursework Table	105
3.40	Coursework type Table	106
3.41	Grades Table	106

3.42	Academic Record Table	106
3.43	Student GPA Table	106
3.44	Prediction Table	106
3.45	Prediction grade type Table	107
3.46	University President Table	107
3.47	Vice-President Table	107
3.48	College Dean Table	107
3.49	Program Coordinator Table	107
3.50	Academic Advisor Table	108
3.51	Academic Advisor for Student Table	108
3.52	Faculty Member Table	108
3.53	Student Table	108
3.54	School type Table	109
4.1	Second-year students Dataset	111
4.2	Second-year students Data Visualization	112
4.3	Second-year students test results	112
4.4	Third-year students Dataset	113
4.5	Third-year students Data Visualization	113
4.6	Third-year students test results	113
4.7	Prediction in the seventh week Dataset	115
4.8	Prediction in the seventh week Data Visualization	115
4.9	Prediction in the seventh week test results	116
4.10	Prediction in the fourteenth week Dataset	116
4.11	Prediction in the fourteenth week Data Visualization	117

4.12	Prediction in the fourteenth week test results	117
4.13	subjects interface	118
4.14	Forgot Password interface	118
4.15	Forgot Password interface	119
4.16	Accounts interface	120
4.17	Reports interface	120
4.18	Reports interface	121
4.19	Reports interface	122
4.20	Reports interface	123
4.21	Program interface	124
4.22	Reports interface	124
4.23	Students interface	125
4.24	Reports interface	126
4.25	subjects interface	127
4.26	Communication interface	127
4.27	Registered Subjects interface	128
4.28	Current Subjects interface	129
4.29	Subject prediction interface	129
4.30	Graduation prediction interface	130
4.31	Academic record interface	131
4.32	Communication interface	131
5.1	Appendices api page	136

LIST OF ABBREVIATIONS

PLOs	Programs Learning Outcomes
UoT	University of Tabuk
IT	Information Technology
FCIT	Faculty of Computers and Information Technology
UML	Unified Modelling Language
ACM	Association for Computing Machinery
IEEE	Institute of Electrical and Electronics Engineers
LMS	Learning Management System
SSA	Student Success Algorithm
GPA	Grade Point Average
GUI	Graphical User Interface
etc	et cetera
ERD	Entity Relationship Diagram

CHAPTER 1

INTRODUCTION

1.1 Background

Achieving better educational outcomes for university students is an important and critical goal. Educational institutions strive to prepare graduates who are qualified for the job market and capable of integrating into society. If graduates' skills are achieved acceptably and align with the requirements of the job market, this will have a positive impact on employment opportunities and, consequently, contribute to the support of the national economy. Each university program includes Program Learning Outcomes (PLOs) that encompass three domains: knowledge, skills, and values that students should acquire upon completing their university education. Enhancing predictive models holds significant importance across various fields. They are instrumental in forecasting outcomes in areas such as medicine, predicting disease occurrences, and in meteorology for weather forecasting, and other domains. These models depend on the data and information they contain to generate precise predictions. The project development is based on the predictive model to measure the attainment of academic learning outcomes of students at University of Tabuk (UoT) in a refined and thoughtful manner. Achieving learning outcomes at the university and enhancing predictive models in a variety of fields represent important aspects of besting the quality and performance of our societies and economies.

1.2 Problem overview

Predicting student failure is extremely challenging, and there are difficulties in both limiting and preventing student failures. Furthermore, achieving a higher and optimal percentage of learning outcomes that align with the job market and prepare students for it is also a challenge.

1.3 Aims and Objectives

The main aim of this project is to develop A Prediction Model-Based Application for Assessing the Programs Learning Outcomes (PLOs) Attainment at University of Tabuk (UoT).

Sub-Goals:

- Develop a prediction model: including:
 - Determine the desired Dataset
 - Training the Model
 - Testing and verifying the developed Model
- In order to develop the intended system application, might including:
 - Requirements collection and system analysis.
 - System and database design.
 - System implementation and testing.

1.4 The Importance of the Project

A web-based application aimed at predicting students' academic performance to prevent failures can have a positive influence on their learning outcomes and better prepare them for the job market. Such an application could utilize various data-driven methods, including machine learning and predictive analytics, to identify students at risk of academic challenges at an early stage. This early identification enables educators and institutions to offer timely interventions and support. Ultimately, this can lead to enhanced educational

achievements and a more skilled workforce, benefiting both individuals and the nation's economy.

1.5 The Scope of the Project

In this project, a dataset was used related to programming subjects within the Information Technology (IT) program students at the Faculty of Computers and Information Technology (FCIT) at University Tabuk (UoT), covering the period from 1429/1430 to 1441/1442. This dataset includes academic data of students from their admission date, encompassing previous information such as the type of school they attended, their cumulative high school GPA, their scores in the general aptitude test, and other fundamental student information. It also covers their grades in programming courses such as Programming 1 and Programming 2, etc. The data was gathered for the purpose of training and testing a predictive model to ensure the creation of an accurate and reliable model. During the program development phase, in the best scenario, the model is applied to the data of both current and new students. however, it will be applied and improved using another dataset, which will be similar to the one the model was trained on, albeit with slight variations in additional personal information about the students, such as social status, type of residence, etc.

1.6 Report's Layout

The first chapter serves as an introduction to the report. In addition to the current chapter, the report consists of five chapters in total. These chapters are as follows: The second chapter explores related works, presents a collection of functional and non-functional requirements, and conducts an analysis of the model and system. The third chapter focuses on system and database design, providing insights into their structures and architectures. Chapter 4 describes the implementation and testing of the system, showcasing the practical aspects of its development and ensuring its functionality. Finally, in Chapter 5, the report presents conclusions drawn from the research

and provides recommendations for future actions or improvements based on the study's outcomes.

1.7 Project Methodology

1.7.1 Prediction model

The prediction model is built using the standard methodology known for building predictive models in data science and artificial intelligence.

1.7.2 Waterfall Model

There are many methods used in the systems development lifecycle, but in this project will follow the waterfall model. The waterfall model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap. The figure below shows Waterfall Model[1].

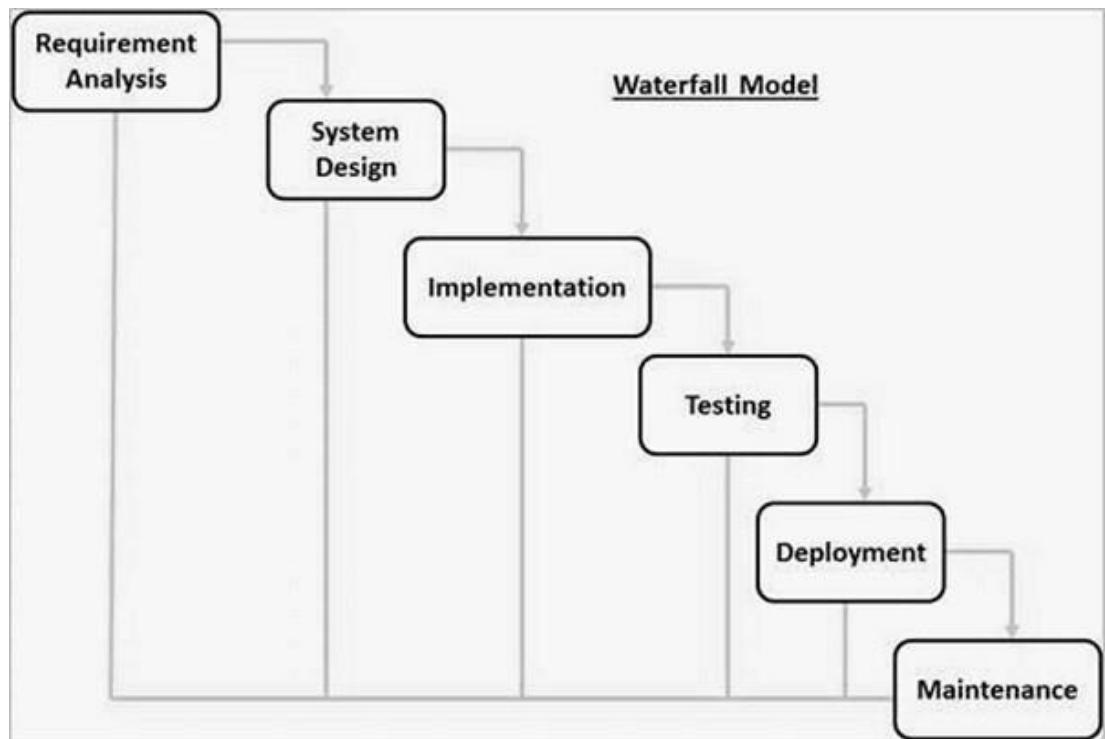


Figure 1.1: Waterfall Model

CHAPTER 2

THE REQUIREMENTS COLLECTION AND SYSTEM ANALYSIS

2.1 Introduction

This chapter explores and reviews related works, presents a collection of functional and non-functional requirements, and conducts an analysis of the model and system by using UML diagrams, such as use case diagrams.

2.2 Background

This section presents an introduction to program learning outcomes and the utilization of a prediction model within an academic context.

2.2.1 Program Learning Outcome's (PLOs)

Program Learning Outcomes (PLOs) are general descriptions of the knowledge, skills, values, and overall abilities that students are expected to have acquired upon completing a specific degree in the program. These outcomes are designed to provide a clear and measurable statement of what students should be able to accomplish at the end of their educational journey. Program Learning Outcomes (PLOs) are instrumental in setting educational goals, aligning the curriculum, assessing student progress, and continuously improving educational programs. They serve as a roadmap for both educators and students, ensuring that the program delivers on its intended outcomes. They focus on the results of student learning and are typically outlined at the program level. Each program has its specific learning outcomes, and within each program, there are multiple subjects, each with its own learning outcomes. The design of the curriculum should ensure that the learning outcomes of these individual subjects align with and contribute to the achievement of at least one

program-level learning outcome. This alignment is crucial to ensure that the curriculum effectively addresses student outcomes [2]. Computer Science Graduates of the Program will have the ability to:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Communicate effectively in a variety of professional contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

Proficiency domains (Competency Domains) within the specialized program (cognitive, skill-based, and professional) are identified, drawing from the guidance of local and global academic and professional institutions relevant to the discipline. The ensuing list outlines the essential competency domains (Core Areas) that must be incorporated into the final assessment of the Bachelor of Computer Science program. These domains are derived from the ACM & IEEE Computing Curricula 2020 (CC2020), aligning with the previously mentioned matrix of alignment:

- Programming
- Algorithms and Complexity
- Theory of Computation
- Discrete Structures
- Operating Systems
- Architecture and Organization
- Computer Graphics and Visualization
- Software Engineering

- Database systems and Data Modeling
- Human-computer Interaction
- Intelligent Systems,
- Social and Professional issues
- Information Assurance and Security

2.2.2 Predictive Models

Machine learning predicts outcomes and enhances decision-making in fields like education and business using data analysis techniques such as decision trees and neural networks. It's valuable for forecasting student success, relying on data quality and supervised learning for predictions. algorithms used with machine learning, M5P machine learning algorithm using decision trees to predict numeric values. It partitions data and builds models using simple mathematical functions. Useful for forecasting student performance without bias or superficial understanding, and Random forests, a collection of decision trees, offer accurate predictions and mitigate overfitting, making them useful for forecasting high school student's academic performance [2]

2.3 Related Works

This section presents a critical review and evaluation of a sample of the previous related work. Despite the availability of numerous research papers, there is a clear scarcity in implementing this research into system applications.

2.3.1 Course Signals at Purdue University

Course Signals is a predictive learning analytics system that was originally developed at Purdue University in the United States. This system leverages student data to proactively identify individuals who may be at risk of not successfully completing their courses. By utilizing predictive modelling of student data and their interactions within the learning management system

(LMS), each student is categorized into one of three 'risk groups,' represented by the colours of a traffic signal: red, yellow, or green [3]. As a result, when the approximately 11,000 students enrolled in introductory courses access the course website or use their mobile devices, they encounter these colored signals which provide insight into their progress. Furthermore, these signals are accompanied by messages from instructors, offering specific recommendations to enhance their academic performance, such as attending additional help sessions or reading supplementary materials [5]. Student Success Algorithm (SSA). This algorithm carefully analyses more than 20 data points to evaluate and categorize each student. The SSA comprises four primary components:

1. Performance: This factor is determined by assessing the percentage of points earned by the student in the course up to the present point.
2. Effort: Effort is measured through the student's interactions with Blackboard Vista and Purdue's learning management system, in comparison to their peers.
3. Prior Academic History: This component takes into consideration the student's academic background, including their academic preparation, high school GPA, and standardized test scores.
4. Student Characteristics: Variables such as residency, age, and marital status are factored in to further refine the predictions. Weighted Calculation: Each of the four components is assigned specific weights, which are then incorporated into the proprietary algorithm.

The algorithm subsequently computes a result for each student based on these weighted inputs. Outcome Signals: Based on the results of the SSA, a corresponding-colored signal is displayed on each student's course homepage:

- Red Signal: This indicates a high likelihood of the student being unsuccessful in the course.
 - Yellow Signal: It suggests a potential problem in succeeding.
 - Green Signal: This demonstrates a high likelihood of success in the course.
- This proactive system allows instructors and students to monitor progress and address potential issues promptly, thereby enhancing overall student success and retention rates [5].

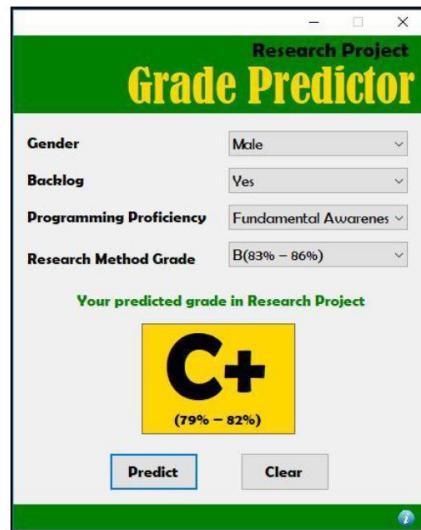


Figure 2.1: Course Signals

The proposed application shares similarities in its predictive capabilities to evaluate learning outcomes through prediction-based analysis with the aim of improving student performance. They are similar using different colours to represent students' progress. In addition, specific recommendations are made to enhance academic performance, such as attending additional help sessions and using predictive modelling techniques to analyse student data and provide results-based recommendations. However, there are differences between the two applications. The Prediction Model-Based Application is associated with a system that allows students to predict their grades in specific subjects and provides a solution to improve academic performance. In contrast, Course Signals discusses a system designed to identify students at risk of not successfully completing their courses. The Prediction Model-Based Application also mentions using colours to display performance predictions, while Course Signals uses colours to classify students into different risk groups. Finally, The Prediction Model-Based Application does not provide specific information about the factors used in predictive analysis, while Course Signals identifies four primary factors considered when evaluating students.

2.3.2 Grade Predictor

The Grade Predictor Software predicts research project grades based on specific conditions, assisting Teachers or those responsible for students in counselling students with low predicted grades. It was developed using a card-based design approach, enhancing usability with a user-centred GUI. The application was created using Visual C#, after 10-fold cross-validation, the Random Tree decision tree algorithm was chosen to extract classification rules. These rules, generated from leaf nodes along the path of the root node, represent predicted grades. A total of 28 rules were extracted from table 111, which correspond to specific predicted grades and can determine if a student needs assistance in the research class based on their predicted grade[6].



The Grade Predictor Software.

Figure 2.2 : Grade Predictor

The proposed application shares similarities in methods, aiming to support educators or responsible individuals in advising students facing anticipated lower grades. However, the Grade Predictor application differs from the Prediction Model-Based Application in that it was developed using a card-based design approach. Furthermore, the Random Tree decision tree algorithm was selected for extracting classification rules within the Grade Predictor application.

2.3.3 UG Partner

An application that helps students at the University of Ghana predict the grades they need in the upcoming semesters to graduate with their desired class. The app also includes the added feature of providing students with past exam questions. You can also access test templates from the app [7].



Figure 2.3 : UG Partner

The proposed application share similarities in their predictive capabilities for assessing learning outcomes. However, the UG Partner application distinguishes itself from the Prediction Model-Based Application by calculating cumulative GPAs and offering students access to past exam questions and test templates.

2.4 System Description

This section will explain how the system works in detail. The process starts when the student registers his account and completes his personal information. The student will find options to register Programming related subjects. If he has studied any of these topics, he must register the grade he received. If it has not been studied yet, the system helps with the expected grade. This grade is based on his school GPA, previous test scores such as academic achievement and general abilities. When the information is entered, the system sends this data to a predictive model. This model shows its predictions based on the information provided and the previous information it has stored. The result is then sent to the system, where the student can see the expected grade for upcoming courses. A colour will be announced to show how well the expectations match the student's actual performance. The meaning of each colour and the focus points that students should consider for bettering their performance are also clarified. The system provides a solution for students who are having difficulty in bettering their grades, such as communicating with an academic advisor or suggesting a list of private lessons to enhance the student's excellence. In addition, the system allows students to estimate their current academic status by adding their courses and entering current test scores. The system will be able to provide predictions for their performance in the same way as for programming courses. To assist students in working to better their academic performance effectively and early, mitigating the risk of failure.

2.5 The System Requirements

The collection of system requirements starts with the understanding of user requirements, which can represent many viewpoints and impose many constraints on the system. In this chapter, in addition to understanding the functions performed by the user.

2.5.1 Functional Requirements of the System

Functional requirements emphasize what the system should do to perform its functions. It is the study of the required tasks that the system should do, these tasks are known as the set of functional requirements of the system, and it also defines how the system behaves. Shall define all the system requirements and make them clear, and easy to understand. The system functional requirements are as follows:

A. User Functional Requirements:

The following Functional Requirements are common and general across all major users views (Actors).

1. User Registration.
2. Log in.
3. Notifications.
4. Messaging. (SMS, Emails)
5. Help and Support.
6. Manage User Profile. (View, Add, and Update).
7. Sign out.

B. Admin functional requirements:

1. Manage members accounts. (View, Add, Update, and Delete).
2. Create performance reports for students, subjects, etc.
3. Manage the basic system data. (View, Add, Update, and Delete).

In an ideal scenario, the system requires multiple views of the university database such as information about academic programs, study plans, faculty data, and student data. However, for the purpose of the graduation project, the system will be developing and handling issues by adding simple additional data, which is only used for testing.

C. President of the University functional requirements:

1. Monitor and generate the performance reports and other associated reports for all academic programs, students, subjects at all departments associated to all colleges at the University.

D. Vice President for Academic Affairs functional requirements:

1. Monitor and generate the performance reports and other associated reports for all academic programs, students, subjects at all departments associated to all colleges at the University.

E. Dean of the College functional requirements:

1. Monitor and generate the performance reports and other associated reports for all academic programs, students, subjects at all departments associated to the College.

F. Program Coordinator functional requirements:

1. Manage program information (View, Add, Update, and Delete).
 - a. Manage program basic information (View, Add, Update, and Delete).
 - b. Manage program learning outcomes (View, Add, Update, and Delete).
 - c. Manage program subjects information (View, Add, Update, and Delete).
2. Monitor program learning outcomes for each program within the term.
 - a. Receiving periodic and detailed reports on the achievement of program learning outcomes and other associated reports.

G. Academic Advisor functional requirements:

1. Monitor of students' performance progress for whom he is responsible as an academic advisor.
2. View student profiles and their academic records.
3. Monitor and Generate performance reports for students, subjects, etc.
4. Communicate with students for whom he is responsible as an academic advisor.
5. Identify students at risk for whom he is responsible as an academic advisor.

H. Faculty Member functional requirements:

1. Monitor students' performance progress in specific subjects taught by a faculty member for a given academic semester.
2. Manage grades for subject-related work within the semester. (View, Add, Update, and Delete).
3. Communicate with students in specific subjects taught by a faculty member for a given academic semester.
4. Identify students at risk in specific subjects taught by a faculty member for a given academic semester.

I. Student Functional Requirements:

1. View subjects information, such as subject schedules, subject grade, etc.
2. View the predicted performance for each subject organized per semester.
3. View previous academic performance.
4. Identify subjects requiring improvement.
5. Communicate with the Academic Advisor or faculty member.

2.5.2 Non- Functional Requirements of the System

Non-functional requirements are the measurement of the performance of the functional requirements, it describes how the system will do its functions, not only what it will do. A system user interface, performance, design constraints, and quality are examples of non-functional requirements. Non-functional requirements are sometimes difficult to be tested, so they are usually evaluated individually, Non-functional requirements may include:

- **Availability**

It means that at any time the user accesses the website, it must be available.

- **Reliability**

The system or service performs its intended function appropriately and operates in a specified environment without failure.

- **Simplicity**

The system that prioritizes ease of use, straightforwardness, and minimal complexity in its design and operation.

- **Performance**

The performance of the system must be acceptable in its operations, the faster the system responds to the user's query, the better performance it has.

- **Security**

This non-functional requirement is very important, where the system does not accept invalid data, and administrators have the authority to modify, add, etc.

- **Usability**

The system's user interface should be intuitive and user-friendly.

- **Compliance**

The system should comply with relevant data protection regulations and industry standards to ensure legal and ethical operation.

- **Logging and Monitoring**

System logs and performance metrics should be collected and monitored to identify and address issues.

2.6 System Analysis

This section employs the Unified Modeling Language (UML) for system analysis due to its substantial assistance in the analytical process.

2.6.1 Use Case Diagrams

A use case diagram is one of the Unified Modelling Language (UML), used to summarize the system details and its users (actors) and their tasks within the system. Use case contains special symbols and notations such as:

- Use cases: scenarios where the actor interacts with the system.
- Actors: the people who are going to interact with the system.

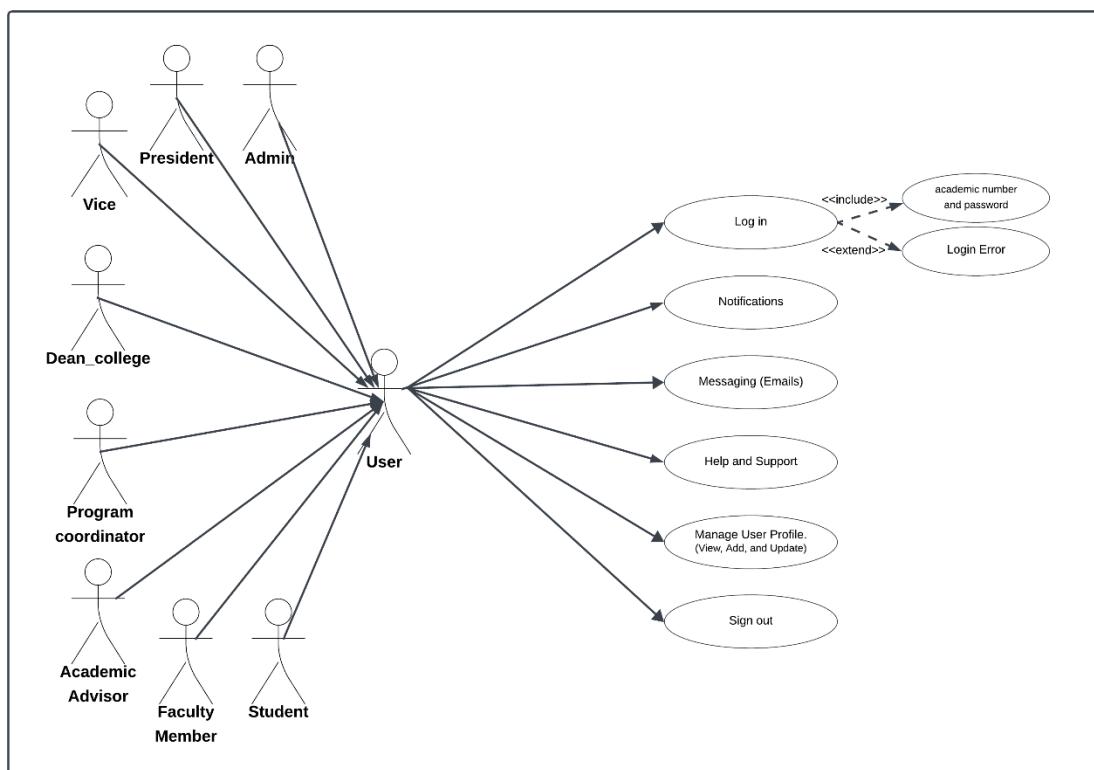


Figure 2.4: User use case diagram

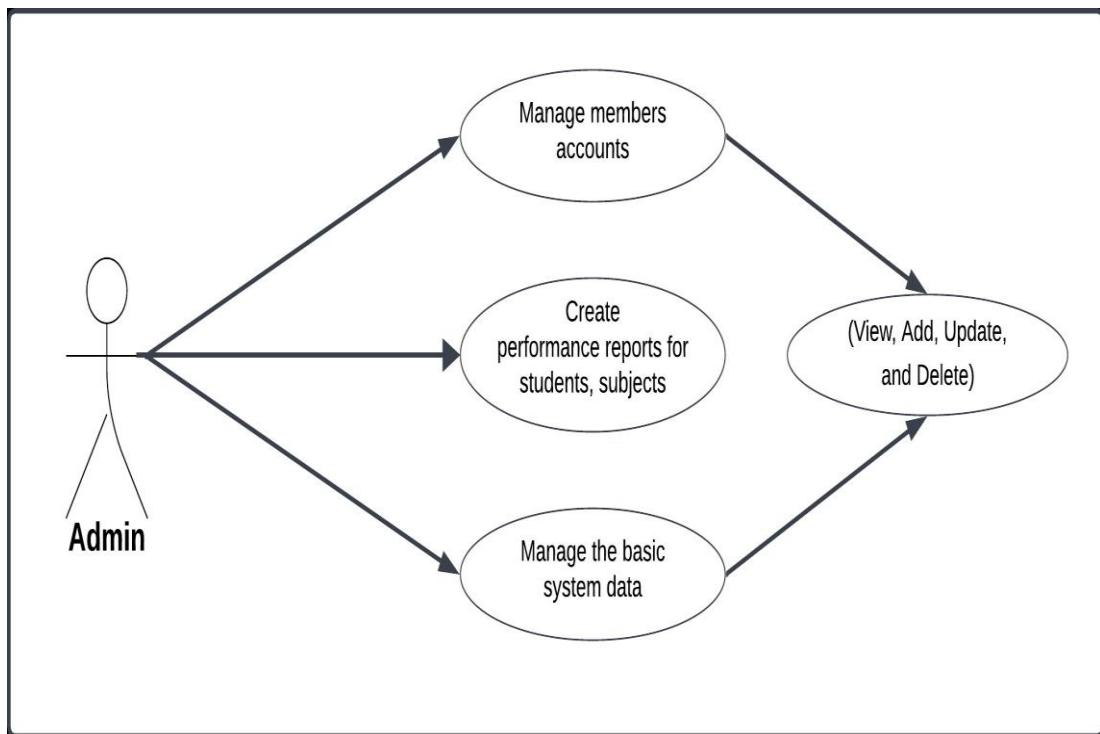


Figure 2.5: Admin use case diagram

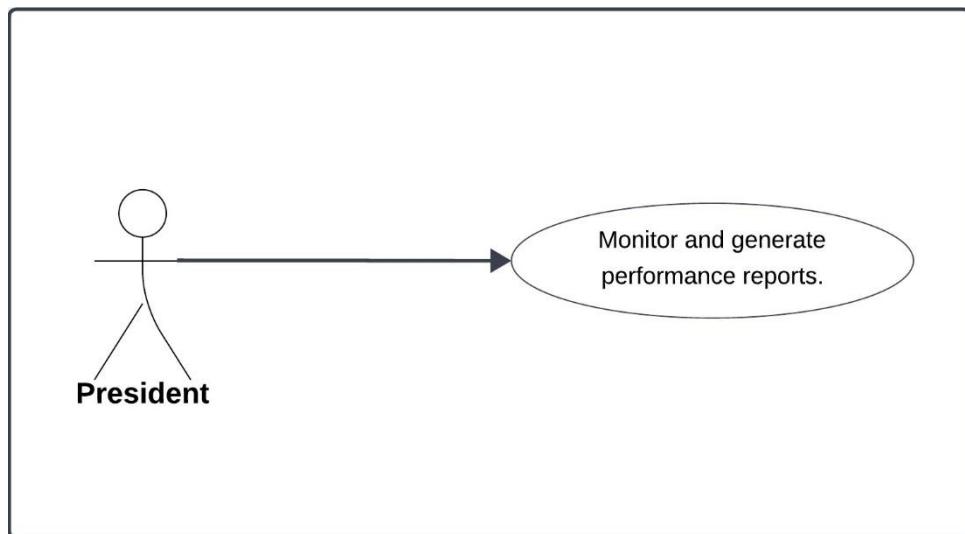


Figure 2.6: President of university use case diagram

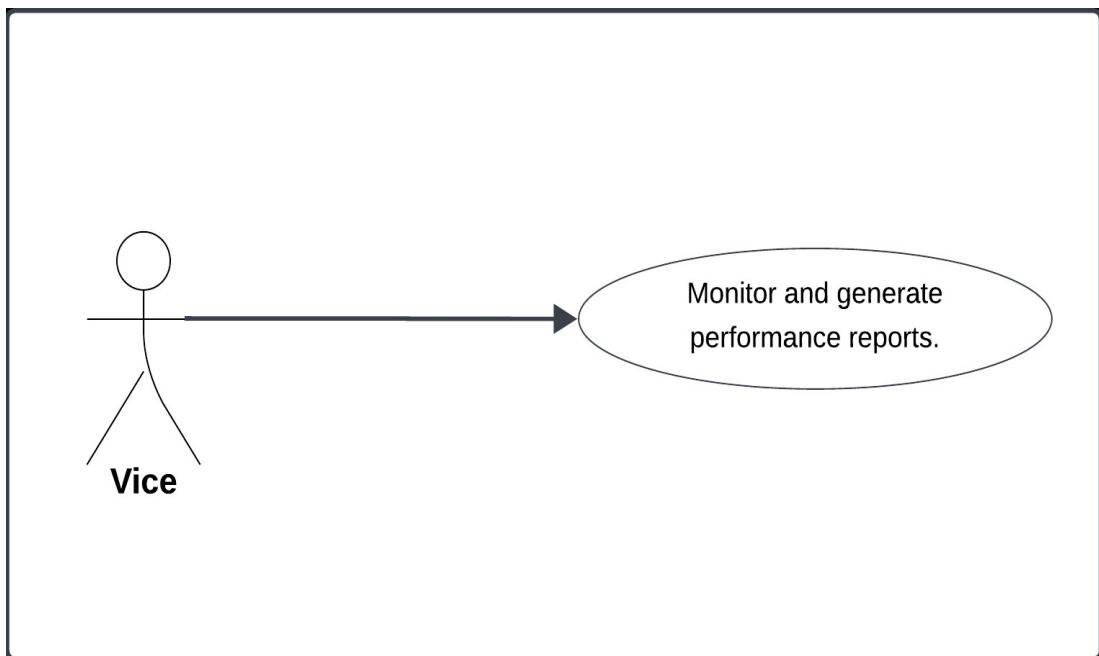


Figure 2.7: Vice President for Academic Affairs use case diagram

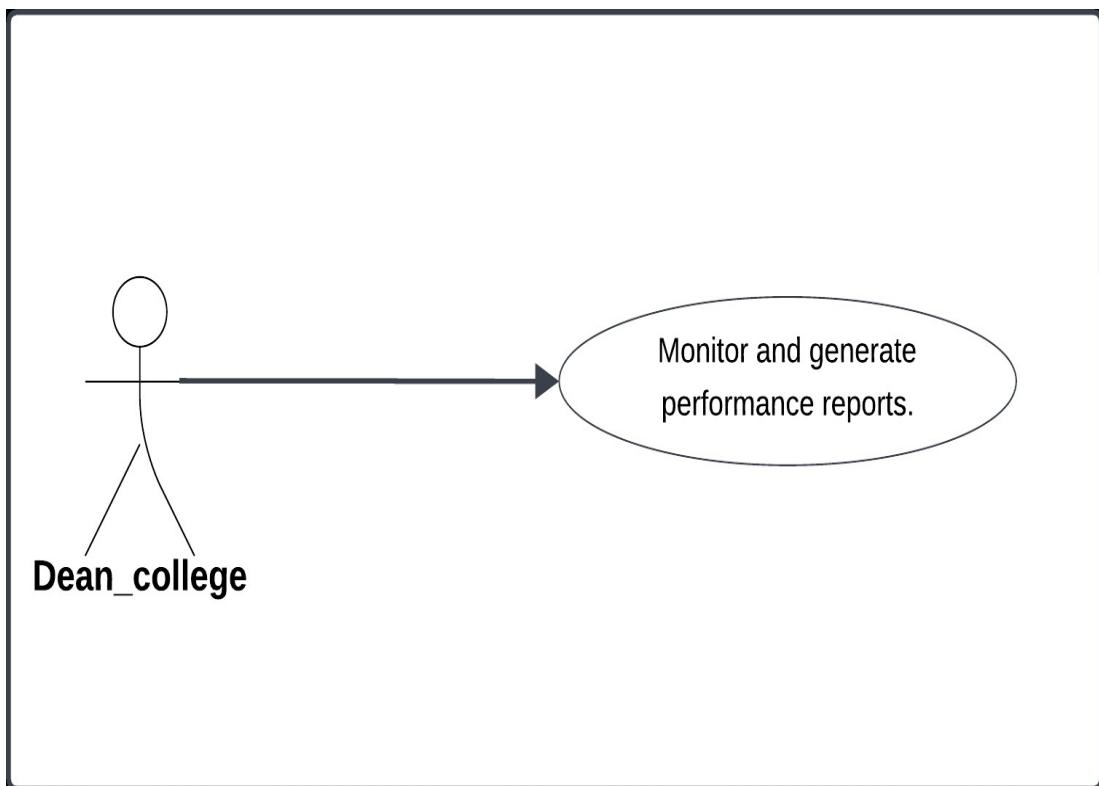


Figure 2.8: Dean of college use case diagram

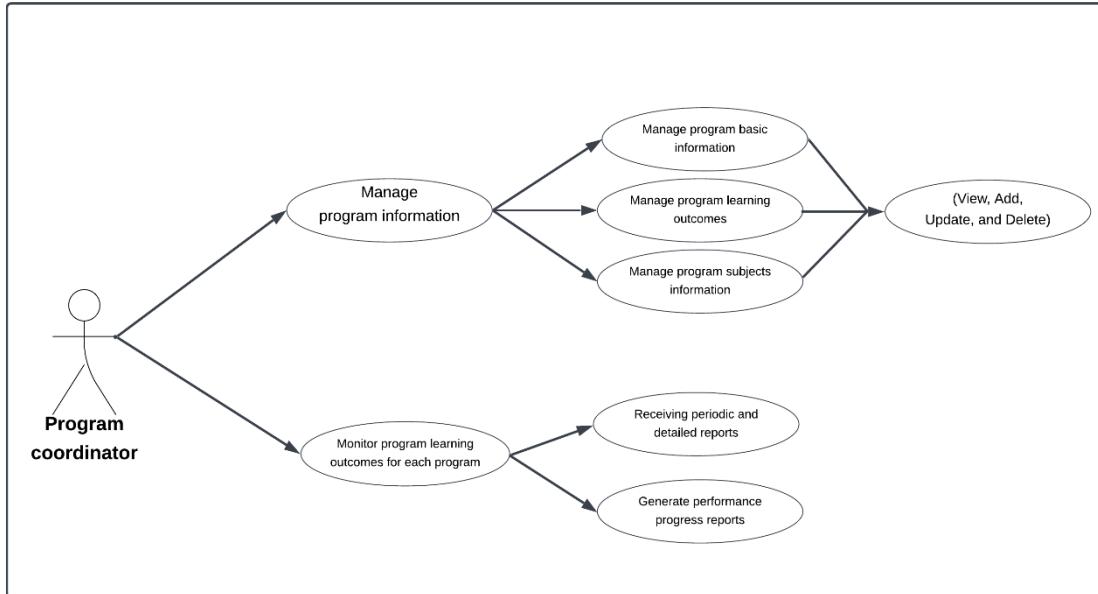


Figure 2.9: Program Coordinator use case diagram

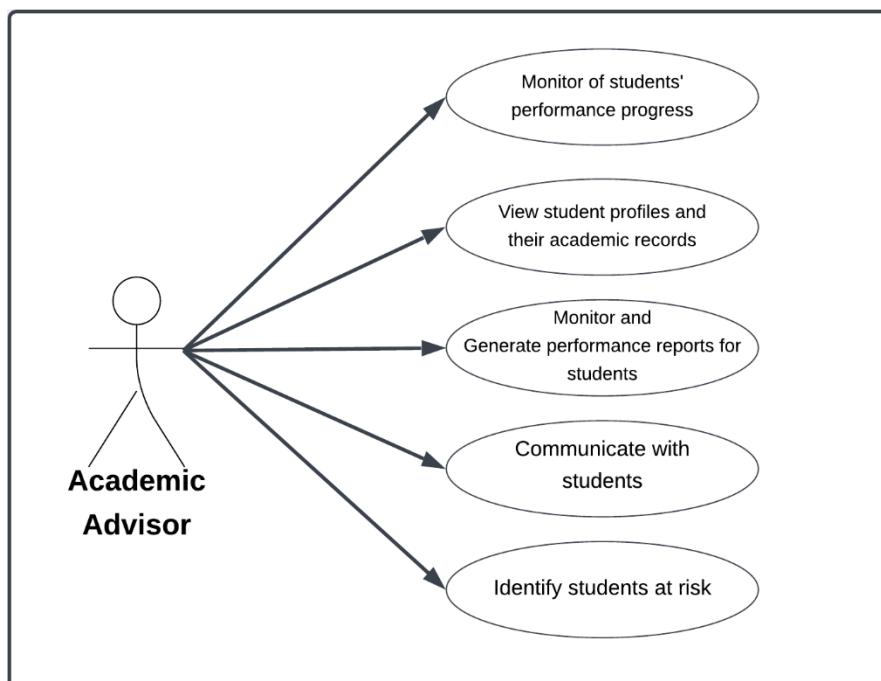


Figure 2.10: Academic Advisor use case diagram

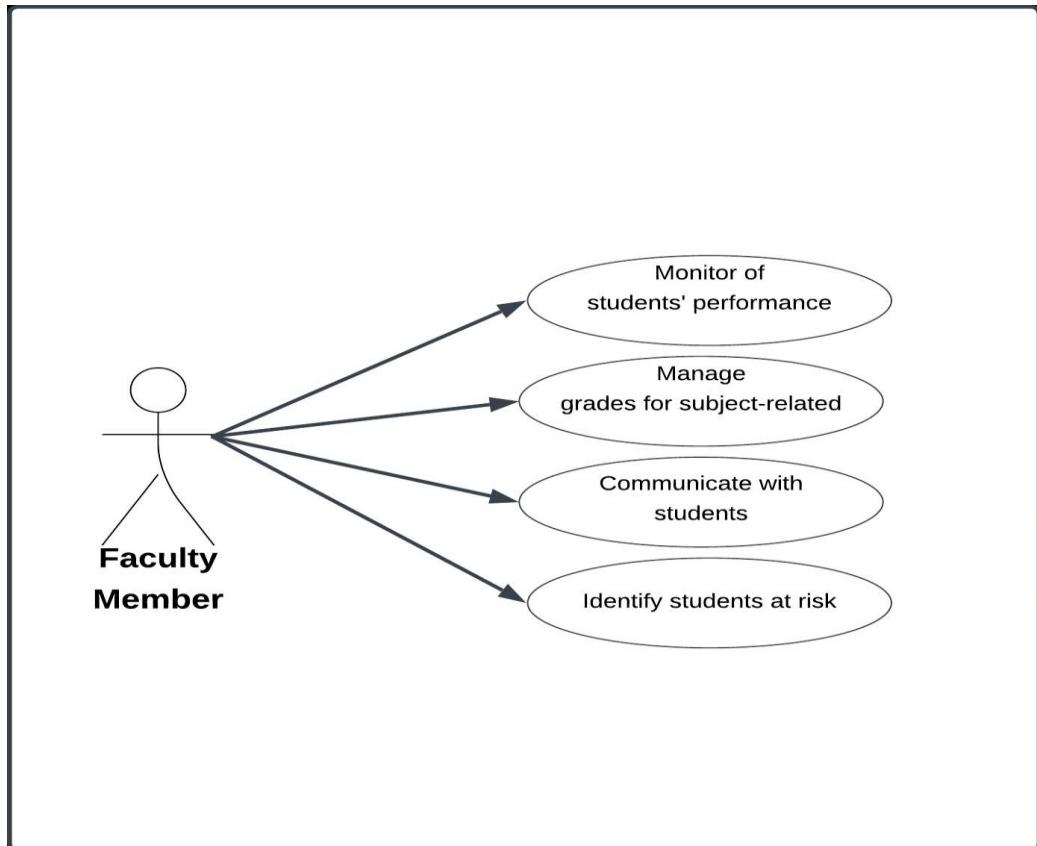


Figure 2.11: Faculty Member use case diagram

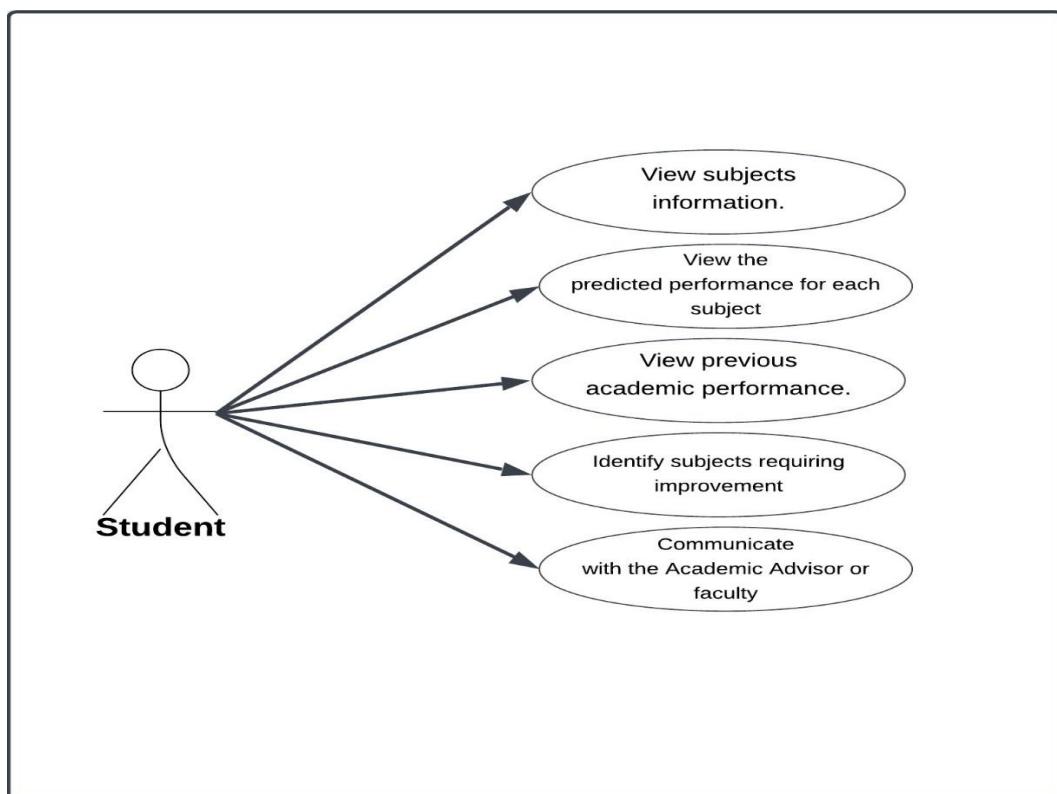


Figure 2.12 : Student use case diagram

2.6.2 Use Case Scenario

is a specific representation illustrating how a system interacts with its users (actors) to accomplish a particular goal or function. It portrays a sequential series of steps or actions performed by the user and the system to complete a specific task or process within the system. Use case scenarios are employed to describe and understand the interactions and functionalities of a system in real-life situations or specific scenarios.

Table 2.1: Log in (User Scenario)

Operation ID	1
Operation Name	Log in
Actor	Admin - Dean of the University - Vice President for Academic Affairs - Dean of the College - Program Coordinator - Academic Advisor - Faculty Member - Student
Description	This page is used to log into the system using your academic number and password
Exception	If the actor is not registered, the actor cannot use this page
Operation Flow	<p>1) Actor: Enter your academic number and password</p> <p>2) System: validate the data in the database</p> <p> 2.1) If it is correct:</p> <p> System: displays “main” page.</p> <p> 2.2) If it is wrong:</p> <p> 2.2.1) displays a message that “the academic number or password is incorrect”.</p> <p> 2.2.2) Enter your information correctly.</p> <p> 2.2.3) Cleaning textbox tools</p>

Table 2.2: Manage User Profile (User Scenario)

Operation ID	2
Operation Name	Manage User Profile.
Actor	Admin - Dean of the University - Vice President for Academic Affairs - Dean of the College - Program Coordinator - Academic Advisor - Faculty Member - Student
Description	This page is used to manage user profile information within the system, for example, uploading a new profile picture, and more.
Exception	None.
Operation Flow	<ol style="list-style-type: none"> 1) Actor: Location is on the main page, clicks on his “Profile icon”. 2) System: display a dropdown menu with various options including “Profile information”. 3) Actor: select from options “Profile information”. 4) System: display form page of “Profile information”. 5) Actor: Update or add personal information, which could include contact details, profile picture, etc. 6) Actor: After making desired changes, save the updates. 7) System: save the updates in the database and display the Profile information.

Table 2.3: Sign out (User Scenario)

Operation ID	3
Operation Name	Sign out.
Actor	Admin - Dean of the University - Vice President for Academic Affairs - Dean of the College - Program Coordinator - Academic Advisor - Faculty Member - Student

Description	This page is used to Sign out of the system.
Exception	None.
Operation Flow	<p>1) Actor: Location is on the main page, clicks on his “Profile icon”.</p> <p>2) System: display a dropdown menu with various options including “Sign out”.</p> <p>3) Actor: selects the “Sign out” option.</p> <p>3) System: displays a confirmation pop-up asking, “Are you sure you want to sign out?”</p> <p>4) Actor: confirms by clicking “Sign Out” or “No” within the pop-up dialog box.</p> <p>4.1) If actor click No:</p> <p> 4.1.1) System: displays “main” page.</p> <p>4.2) If actor click Sign Out:</p> <p> 4.2.1) System: logs actor out of his account and redirects him to the “login page”.</p>

Table 2.4: Manage the basic system data (Admin Scenario)

Operation ID	4
Operation Name	Manage the basic system data.
Actor	Admin
Description	This page is used to manage members accounts such as, permissions, usernames, etc. on the system.
Exception	None.
Operation Flow	<p>1) Actor: click on “members accounts” on the “Main” page, or click “members accounts” from the navigation bar.</p> <p>2) System: display page of “members accounts”.</p> <p>3) Actor: View the “members accounts” page and can choose between the four options: View, Add, Update, and Delete.</p> <p> 3.1) if actor choose View:</p>

	<p>3.1.1) System: display page of “members accounts”.</p> <p>3.2) if actor choose Add:</p> <p>3.2.1) System: display form page of add “Manage members accounts”.</p> <p>3.2.2) Actor: Fill the form with required data and save.</p> <p>3.2.3) System: validates the data in the database:</p> <p>3.2.3.1) If the data is found: an error message will be displayed, stating that “the data added for member accounts cannot be added because it already exists.”.</p> <p>3.2.3.2) If the data is not found: Saves the new members accounts in the database, and displays page of “members accounts”.</p> <p>3.3) if actor select Update:</p> <p>3.3.1) System: display page of “members accounts”.</p> <p>3.3.2) Actor: Select the specific member accounts that the actor wants to update.</p> <p>3.3.3) System: display form page of “Update members accounts”.</p> <p>3.3.4) Actor: Fill the form with updated information and save.</p> <p>3.3.5) System: Saves the updated program basic information in the database, and display page of “members accounts”.</p> <p>3.4) if actor select Delete:</p> <p>3.4.1) System: display page of “members accounts”.</p> <p>3.4.2) Actor: Select the specific member accounts that the actor wants to delete.</p> <p>3.4.3) System: displays a confirmation pop-up asking, "Are you sure you want to delete the selected member accounts from database?"</p>
--	---

	<p>3.4.4) Actor: confirms by clicking "Yes" or "No" within the pop-up dialog box.</p> <p>3.4.4.1) if actor clicking "Yes":</p> <p>3.4.4.1.1) System: Delete the selected member accounts from members accounts in the database, and display page of “members accounts”.</p> <p>3.4.4.2) if actor clicking "No":</p> <p>3.4.4.1.2) System: display a message “The command has been canceled”, and display page of “members accounts”.</p>
--	--

Table 2.5: Create reports (Admin Scenario)

Operation ID	5
Operation Name	Create reports.
Actor	Admin
Description	This page is used to Create performance reports for students, subjects, etc.
Exception	None.
Operation Flow	<p>1) Actor: click on “Create reports” on the “Main” page, or select “Reports” from the navigation bar, the system will display a dropdown menu with various options, including “Create reports”.</p> <p>2) System: display form page of “Create reports”.</p> <p>3) Actor: Fill the form with data and save it, then select the recipient's name or their email address for the report.</p> <p>4) System: validate the data in the database</p> <p>4.1) If the data is found:</p> <p>4.1.1) System: send the report to select the recipient's name or their email address and display a message "The report has been sent successfully.".</p> <p>4.2) If the data isn't found:</p>

	4.2.1) displays a message that "No person with this name or similar email has been found, please Enter the address correctly.".
--	---

Table 2.6: Manage system data (Admin Scenario)

Operation ID	6
Operation Name	Manage system data
Actor	Admin
Description	This page is used to Manage the basic system data, such as Data Maintenance, Data Entry, etc. on the system.
Exception	None.
Operation Flow	<p>1) Actor: click on “Manage system data” on the “Main” page, or click “Manage system data” from the navigation bar.</p> <p>2) System: display page of “Manage system data”.</p> <p>3) Actor: View the “Manage system data” page and can choose between the four options: View, Add, Update, and Delete.</p> <p>3.1) if actor choose View:</p> <p>3.1.1) System: display page of “Manage system data”.</p> <p>3.2) if actor choose Add:</p> <p>3.2.1) System: display form page of add “Manage system data”.</p> <p>3.2.2) Actor: Fill the form with required data and save.</p> <p>3.2.3) System: validates the data in the database:</p> <p>3.2.3.1) If the data is found: an error message will be displayed, stating that “the data added for system cannot be added because it already exists.”.</p>

	<p>3.2.3.2) If the data is not found: Saves the new system data in the database, and displays page of “Manage system data”.</p> <p>3.3) if actor select Update:</p> <p>3.3.1) System: display page of “Manage system data”.</p> <p>3.3.2) Actor: Select the specific system data that the actor wants to update.</p> <p>3.3.3) System: display form page of “Update system data”.</p> <p>3.3.4) Actor: Fill the form with updated data and save.</p> <p>3.3.5) System: Saves the updated Manage system data in the database, and display page of “Manage system data”.</p> <p>3.4) if actor select Delete:</p> <p>3.4.1) System: display page of “Manage system data”.</p> <p>3.4.2) Actor: Select the specific system data that the actor wants to delete.</p> <p>3.4.3) System: displays a confirmation pop-up asking, "Are you sure you want to delete the selected system data from database?"</p> <p>3.4.4) Actor: confirms by clicking "Yes" or "No" within the pop-up dialog box.</p> <p>3.4.4.1) if actor clicking "Yes":</p> <p>3.4.4.1.1) System: Delete the selected system data from system data in the database, and display page of “Manage system data”.</p> <p>3.4.4.2) if actor clicking "No":</p> <p>3.4.4.1.2) System: display a message “The command has been canceled”, and display page of “Manage system data”.</p>
--	--

Table 2.7: Monitor and generate the reports (President of the University Scenario)

Operation ID	7
Operation Name	Monitor and generate the reports.
Actor	President of the University
Description	Monitor and generate the performance reports and other associated reports for all colleges, and programs, etc. in the system.
Exception	None.
Operation Flow	<p>1) Actor: click on “Monitor and generate reports” on the “Main” page, or select “Reports” from the navigation bar, the system will display a dropdown menu with various options, including “Monitor and generate reports”.</p> <p>2) System: display page of “monitor and generate reports”.</p> <p>3) Actor: Views the “monitor and generate reports” page and can select from the 2 options (Monitor, and Generate).</p> <p>3.1) if actor select Monitor:</p> <ul style="list-style-type: none"> 3.1.1) System: display page of “Monitor reports”. 3.1.2) Actor: view receiving the performance reports and other associated reports for all colleges, and programs, etc. <p>3.2) if actor select Generate:</p> <ul style="list-style-type: none"> 3.2.1) System: display form page of “Generate reports”. 3.2.2) Actor: Fill the form with data and save it, then select the recipient's name or their email address for the report. 3.2.3) System: Validates the address send to in the database. 3.2.3.1) If the data is found:

	<p>3.2.3.1.1) System: send the report to select the recipient's name or their email address.</p> <p>3.2.3.2) If the data isn't found:</p> <p>3.2.3.2.1) displays a message that "No person with this name or similar email has been found, please Enter the address correctly.".</p>
--	--

Table 2.8: Monitor and generate the reports (Dean of the College Scenario)

Operation ID	8
Operation Name	Monitor and generate the reports.
Actor	Dean of the University
Description	Monitor and generate the performance reports and other associated reports for all programs, students, subjects, etc., within the department and the programs in the faculty for which he is responsible as its dean.
Exception	None.
Operation Flow	<p>1) Actor: click on “Monitor and generate reports” on the “Main” page, or select “Reports” from the navigation bar, the system will display a dropdown menu with various options, including “Monitor and generate reports”.</p> <p>2) System: display page of “monitor and generate reports”.</p> <p>3) Actor: Views the “monitor and generate reports” page and can select from the 2 options (Monitor, and Generate).</p> <p>3.1) if actor select Monitor:</p> <p>3.1.1) System: display page of “Monitor reports”.</p> <p>3.1.2) Actor: view receiving the performance reports and other associated reports for all colleges, and programs, etc.</p> <p>3.2) if actor select Generate:</p>

	<p>3.2.1) System: display form page of “Generate reports”.</p> <p>3.2.2) Actor: Fill the form with data and save it, then select the recipient's name or their email address for the report.</p> <p>3.2.3) System: Validates the address send to in the database.</p> <p>3.2.3.1) If the data is found:</p> <p>3.2.3.1.1) System: send the report to select the recipient's name or their email address.</p> <p>3.2.3.2) If the data isn't found:</p> <p>3.2.3.2.1) displays a message that "No person with this name or similar email has been found, please Enter the address correctly.".</p>
--	--

Table 2.9: Manage program basic information (Program Coordinator Scenario)

Operation ID	9
Operation Name	Manage program basic information.
Actor	Program Coordinator.
Description	This page is used to manage program basic information on the system.
Exception	None.
Operation Flow	<p>1) Actor: click on “program basic information” on the “Main” page, or select “manage program information” from the navigation bar, the system will display a dropdown menu with various options, including “program basic information”.</p> <p>2) System: display page of “Program basic information”.</p>

	<p>3) Actor: View the “Program Basic Information” page and can choose between the four options: View, Add, Update, and Delete.</p> <p>3.1) if actor choose View:</p> <p>3.1.1) System: display page of “Program basic information”.</p> <p>3.2) if actor choose Add:</p> <p>3.2.1) System: display form page of add “Program basic information”.</p> <p>3.2.2) Actor: Fill the form with information and save.</p> <p>3.2.3) System: validates the information in the database:</p> <p>3.2.3.1) If the information is found: an error message will be displayed, stating that “the information added for program basic information cannot be added because it already exists.”.</p> <p>3.2.3.2) If the information is not found: Saves the new program basic information in the database, and displays page of “program basic information”.</p> <p>3.3) if actor select Update:</p> <p>3.3.1) System: display page of “program basic information”.</p> <p>3.3.2) Actor: Select the specific programs basic information that the actor wants to update.</p> <p>3.3.3) System: display form page of “Update program basic information”.</p> <p>3.3.4) Actor: Fill the form with updated information and save.</p> <p>3.3.5) System: Saves the updated program basic information in the database, and display page of “program basic information”.</p>
--	---

	<p>3.4) if actor select Delete:</p> <p>3.4.1) System: display page of “program basic information”.</p> <p>3.4.2) Actor: Select the specific programs basic information that the actor wants to delete.</p> <p>3.4.3) System: displays a confirmation pop-up asking, "Are you sure you want to delete the selected information from the program's basic information?"</p> <p>3.4.4) Actor: confirms by clicking "Yes" or "No" within the pop-up dialog box.</p> <p>3.4.4.1) if actor clicking "Yes":</p> <p>3.4.4.1.1) System: Delete the selected information from programs basic information in the database, and display page of “Program basic information”.</p> <p>3.4.4.2) if actor clicking "No":</p> <p>3.4.4.1.2) System: display a message “The command has been canceled”, and display page of “Program basic information”.</p>
--	---

Table 2.10: Manage program learning outcomes (Program Coordinator Scenario)

Operation ID	10
Operation Name	Manage program learning outcomes.
Actor	Program Coordinator.
Description	This page is used to manage program learning outcomes on the system.
Exception	None.
Operation Flow	1) Actor: click on “program learning outcomes” on the “Main” page, or select “manage program information” from the navigation bar, the system will display a dropdown menu

	<p>with various options, including “manage program learning outcomes”.</p> <p>2) System: display page of “manage program learning outcomes”.</p> <p>3) Actor: View the “manage program learning outcomes” page and can choose between the four options: View, Add, Update, and Delete.</p> <p>3.1) if actor choose View:</p> <p>3.1.1) System: display page of “Program learning outcomes”.</p> <p>3.2) if actor choose Add:</p> <p>3.2.1) System: display form page of add “Program learning outcomes”.</p> <p>3.2.2) Actor: Fill the form with information and save.</p> <p>3.2.3) System: validates the information in the database:</p> <p>3.2.3.1) If the information is found: an error message will be displayed, stating that “the information added for program learning outcomes cannot be added because it already exists.”.</p> <p>3.2.3.2) If the information is not found: Saves the new program learning outcomes in the database, and displays page of “program learning outcomes”.</p> <p>3.3) if actor select Update:</p> <p>3.3.1) System: display page of “program learning outcomes”.</p> <p>3.3.2) Actor: Select the specific programs learning outcomes that the actor wants to update.</p> <p>3.3.3) System: display form page of “Update program learning outcomes”.</p>
--	---

	<p>3.3.4) Actor: Fill the form with updated information and save.</p> <p>3.3.5) System: Saves the updated program basic information in the database, and display page of “program learning outcomes”.</p> <p>3.4) if actor select Delete:</p> <p>3.4.1) System: display page of “program learning outcomes”.</p> <p>3.4.2) Actor: Select the specific learning outcomes of the programs that the actor want to delete.</p> <p>3.4.3) System: displays a confirmation pop-up asking, "Are you sure you want to delete the selected information from the program's learning outcomes?"</p> <p>3.4.4) Actor: confirms by clicking "Yes" or "No" within the pop-up dialog box.</p> <p>3.4.4.1) if actor clicking "Yes":</p> <p>3.4.4.1.1) System: Delete the selected information from programs learning outcomes in the database, and display page of “Program learning outcomes”.</p> <p>3.4.4.2) if actor clicking "No":</p> <p>3.4.4.1.2) System: display a massage “The command has been canceled”, and display page of “Program learning outcomes”.</p>
--	---

Table 2.11: Receiving reports (Program Coordinator Scenario)

Operation ID	11
Operation Name	Receiving reports.
Actor	Program Coordinator
Description	This page is used to receiving periodic and detailed reports on the achievement of program learning outcomes and other associated reports in the system.

Exception	None.
Operation Flow	<p>1) Actor: click on “Receiving reports” on the “Main” page, or select “Reports” from the navigation bar, the system will display a dropdown menu with various options, including “Receiving reports”.</p> <p>2) System: display page of “Receiving reports”.</p> <p>3) Actor: view the receiving periodic and detailed reports page.</p>

Table 2.12: Generate reports (Program Coordinator Scenario)

Operation ID	12
Operation Name	Generate reports.
Actor	Program Coordinator
Description	This page is used to generate performance progress reports for program learning outcomes. in the system.
Exception	None.
Operation Flow	<p>1) Actor: click on “Generate reports” on the “Main” page, or select “Reports” from the navigation bar, the system will display a dropdown menu with various options, including “Generate reports”.</p> <p>2) System: display form page of “Generate reports”.</p> <p>3) Actor: Fill the form with data and save it, then select the recipient's name or their email address for the report.</p> <p>4) System: validate the data in the database</p> <p>4.1) If the data is found:</p> <p>4.1.1) System: send the report to select the recipient's name or their email address and display a message "The report has been sent successfully.".</p> <p>4.2) If the data isn't found:</p>

	4.2.1) displays a message that "No person with this name or similar email has been found, please Enter the address correctly.".
--	---

Table 2.13: Monitor of students (Academic advisor Scenario)

Operation ID	13
Operation Name	Monitor of students.
Actor	Academic advisor
Description	This page will display monitor of students' performance progress for whom he is responsible as an academic advisor.
Exception	None.
Operation Flow	<ol style="list-style-type: none"> 1) Actor: click on “Students Performance progress” on the “Main” page, or select “Students” from the navigation bar. The system will display a dropdown menu with various options, including “students performance progress”. 2) System: Display the page of “students performance progress”. 3) Actor: view students performance progress through their grades while showing the color of each grade.

Table 2.14: View student profiles (Academic advisor Scenario)

Operation ID	14
Operation Name	View student profiles.
Actor	Academic advisor

Description	This page will display the student profiles for actor responsible as an academic advisor.
Exception	None.
Operation Flow	<p>1) Actor: click on “Students information” on the “Main” page, or select “Students” from the navigation bar. The system will display a dropdown menu with various options, including “Students information”.</p> <p>2) System: Display the page of “Students information”.</p> <p>3) Actor: view “Students information”, such as student profiles and their academic records, etc.</p>

Table 2.15: Communicate with students (Academic advisor Scenario)

Operation ID	15
Operation Name	Communicate with students.
Actor	Academic Advisor
Description	This page displays communicate with students for whom he is responsible as an academic advisor.
Exception	None.
Operation Flow	<p>1) Actor: click on “main menu” on the “Main” page.</p> <p>2) System: the system will display a menu with various options including “Communicate with the Student”.</p> <p>3) Actor: select from options “Communicate with students”.</p> <p>4) System: display page of the “Communicate with students”.</p> <p>5) Actor: select student to Communicate with.</p> <p>6) System: Displays the form page of “Communicate with students”.</p>

	<p>7) Actor: Fills out the form with data, saves it, then selects the student's name as the recipient.</p> <p>8) System: Sends the message.</p>
--	---

Table 2.16: Identify students at risk (Academic advisor Scenario)

Operation ID	16
Operation Name	Identify students at risk.
Actor	Academic advisor
Description	This page will Identify students at risk for whom he is responsible as an academic advisor.
Exception	None.
Operation Flow	<p>1) Actor : click on “Students at Risk” on the “Main” page, or select "Students" from the navigation bar. The system will display a dropdown menu with various options, including "Students at Risk".</p> <p>2) System: view students at risk through their grades while showing the color of each grade.</p>

Table 2.1: Generate reports (Academic advisor Scenario)

Operation ID	17
Operation Name	Generate reports.
Actor	Academic advisor
Description	This page is used to generate performance reports for students and subjects.
Exception	None.
Operation Flow	<p>1) Actor: click on “Generate reports” on the “Main” page, or select "Reports" from the navigation bar. The system</p>

	<p>will display a dropdown menu with various options, including “Generate reports”.</p> <p>2) System: Displays the form page for “Generating reports”.</p> <p>3) Actor: Fills out the form with data, saves it, then selects the recipient's name or email address for the report.</p> <p>4) System: Validates the address send to in the database.</p> <p>4.1) If the data is found:</p> <p>4.1.1) System: send the report to select the recipient's name or their email.</p> <p>4.2) If the data is not found:</p> <p>4.2.1) System: Displays a message that "No person with this name or similar email has been found, please Enter the address correctly.".</p>
--	---

Table 2.18: Monitor of students (Faculty Member Scenario)

Operation ID	18
Operation Name	Monitor of students.
Actor	Faculty Member
Description	This page will display the monitoring of students' performance progress in specific subjects taught by a faculty member for a given academic semester.
Exception	None.
Operation Flow	<p>1) Actor: click on “Students performance progress” on the “Main” page, or select “Students” from the navigation bar. The system will display a dropdown menu with various options, including “students performance progress”.</p> <p>2) System: Display the page of “students performance progress”.</p>

	3) Actor: view students performance progress through their grades while showing the color of each grade.
--	--

Table 2.19: Manage grades (Faculty Member Scenario)

Operation ID	19
Operation Name	Manage grades.
Actor	Faculty Member
Description	This page will be used to Manage grades for subject-related work within the semester in the system.
Exception	None.
Operation Flow	<p>1) Actor: click on “Grades” on the “Main” page or select “Grades” from the navigation bar.</p> <p>2) System: Display the page of “Grades”.</p> <p>3) Actor: View the “Grades” page and can choose between the four options: View, Add, Update, and Delete.</p> <p>3.1) if actor choose View:</p> <p>3.1.1) System: display page of “Grades”.</p> <p>3.2) if actor choose Add:</p> <p>3.2.1) System: display page of “Grades”.</p> <p>3.2.2) Actor: Fill the form with grades and save.</p> <p>3.2.3) System: Saves the added grades in the database, and displays page of “Grades”.</p> <p>3.3) if actor select Update:</p> <p>3.3.1) System: display page of “Grades”.</p> <p>3.3.2) Actor: Select the specific grade that the actor wants to update.</p> <p>3.3.3) System: display form page of “Update grades”.</p> <p>3.3.4) Actor: Fill the form with updated grades and save.</p>

	<p>3.3.5) System: Saves the updated grades in the database, and display page of “Grades”.</p> <p>3.4) if actor select Delete:</p> <p>3.4.1) System: display page of “Grades”.</p> <p>3.4.2) Actor: Select the specific grade that the actor wants to delete.</p> <p>3.4.3) System: displays a confirmation pop-up asking, "Are you sure you want to delete the selected grade?"</p> <p>3.4.4) Actor: confirms by clicking "Yes" or "No" within the pop-up dialog box.</p> <p>3.4.4.1) if actor clicking "Yes":</p> <p>3.4.4.1.1) System: Delete the selected Grade from Grades in the database.</p> <p>3.4.4.2) if actor clicking "No":</p> <p>3.4.4.1.2) System: display page of “Grades”.</p>
--	---

Table 2.20: Communicate with students (Faculty Member Scenario)

Operation ID	20
Operation Name	Communicate with students.
Actor	Faculty Member
Description	This page displays Communicate with students in specific subjects taught by a faculty member for a given academic semester.
Exception	None.
Operation Flow	<p>1) Actor: click on “main menu” on the “Main” page.</p> <p>2) System: the system will display a menu with various options including “Communicate with the Student”.</p>

	<p>3) Actor: select from options “Communicate with students”.</p> <p>4) System: display page of “Communicate with students”.</p> <p>5) Actor: select student to Communicate with.</p> <p>6) System: Displays the form page of “Communicate with students”.</p> <p>7) Actor: Fills out the form with data, saves it, then selects the student's name as the recipient.</p> <p>8) System: Sends the message.</p>
--	--

Table 2.21: Identify students at risk (Faculty Member Scenario)

Operation ID	21
Operation Name	Identify students at risk.
Actor	Faculty Member
Description	This page will Identify students at risk in specific subjects taught by a faculty member for a given academic semester.
Exception	None.
Operation Flow	<p>1) Actor : click on “Students at Risk” on the “Main” page, or select "Students information" from the navigation bar. The system will display a dropdown menu with various options, including "Students at Risk".</p> <p>2) System: view students at risk through their grades while showing the color of each grade.</p>

Table 2.22: View subject information (Student Scenario)

Operation ID	22
Operation Name	view subject information.
Actor	Student

Description	This page is used to view subject information, such as subject schedules, subject grade, etc.
Exception	None.
Operation Flow	<p>1) Actor: click on specified “Subject” on the “Main” page, or select “Subject” from the navigation bar, the system will display a dropdown menu with various options including the specified “Subject”.</p> <p>2) System: display page of the selected “Subject”.</p> <p>3) Actor: view information of subject and, such as subject schedules, subject grade, etc.</p>

Table 2.23: view the predicted academic performance (Student Scenario)

Operation ID	23
Operation Name	view the predicted academic performance.
Actor	Student
Description	This page is used to view predicted performance for each subject while indicating the color associated with each grade.
Exception	None.
Operation Flow	<p>1) Actor: click on specified “Subject” on the “Main” page, or select “Subject” from the navigation bar, the system will display a dropdown menu with various options including the specified “Subject”.</p> <p>2) System: display page of the selected “Subject”.</p> <p>3) Actor: view information of subject and, such as predicted performance for a subject, subject grade, etc.</p>

Table 2.24: View previous academic performance (Student Scenario)

Operation ID	24
--------------	----

Operation Name	View previous academic performance.
Actor	Student
Description	This page will view previous academic performance for Student.
Exception	None.
Operation Flow	<p>1) Actor: click on “main menu” on the “Main” page.</p> <p>2) System: the system will display a menu with various options including “previous academic performance”.</p> <p>3) Actor: select from options “previous academic performance”.</p> <p>4) System: display page of the “previous academic performance”.</p> <p>5) Actor: view previous academic performance.</p>

Table 2.25: Identify subjects requiring improvement (Student Scenario)

Operation ID	25
Operation Name	Identify subjects requiring improvement.
Actor	Student
Description	This page will Identify subject requiring improvement for Student.
Exception	None.
Operation Flow	<p>1) Actor: click on “Subjects requiring improvement” on the “Main” page, or select “Main Menu” the system will display a menu with various options including “Subjects requiring improvement”.</p> <p>2) System: display page of the “Subjects requiring improvement”.</p> <p>3) Actor: view the subjects requiring improvement while indicating the color associated with each subject grade.</p>

**Table 2.26 : Communicate with the Academic Advisor or faculty member
(Student Scenario)**

Operation ID	26
Operation Name	Communicate with the Academic Advisor or faculty member.
Actor	Student
Description	This page displays Communicate with the Academic Advisor or faculty member.
Exception	None.
Operation Flow	<ol style="list-style-type: none"> 1) Actor: click on “main menu” on the “Main” page. 2) System: display a menu with various options including “Communicate the Academic Advisor or faculty member”. 3) Actor: select from options “Communicate the Academic Advisor or faculty member”. 4) System: display page of the “Communicate the Academic Advisor or faculty member”. 5) Actor: select one to Communicate with. 6) System: Displays the form page of “Communicate the Academic Advisor or faculty member”. 7) Actor: Fills out the form with data, saves it, then selects the recipient name. 8) System: Sends the message.

2.6.3 Activity Diagram

is a graphical representation used in the Unified Modeling Language (UML) to illustrate the sequence of activities and processes within a specific system. It displays the sequence of activities and the flow of control among them, including decision points and concurrent activities, to elucidate the processes and flows within the system.

- Figure 2.14 shows activity diagram steps for a process of login by User Actor.

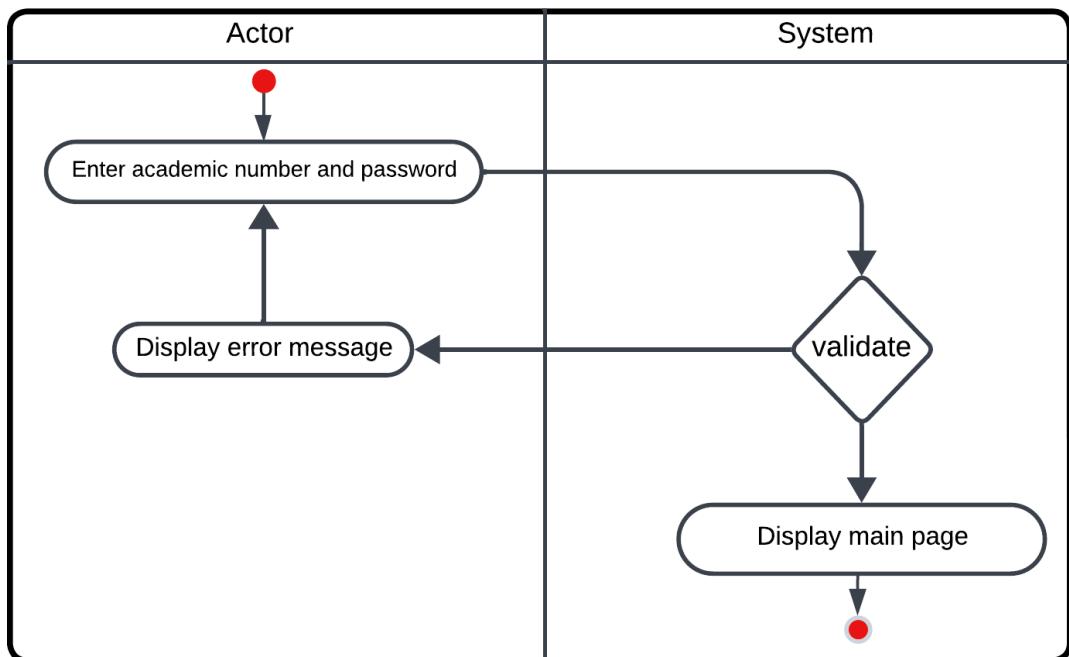


Figure 2.13: Login activity diagram

- Figure 2.15 shows activity diagram steps for a process of manage user profile by User Actor

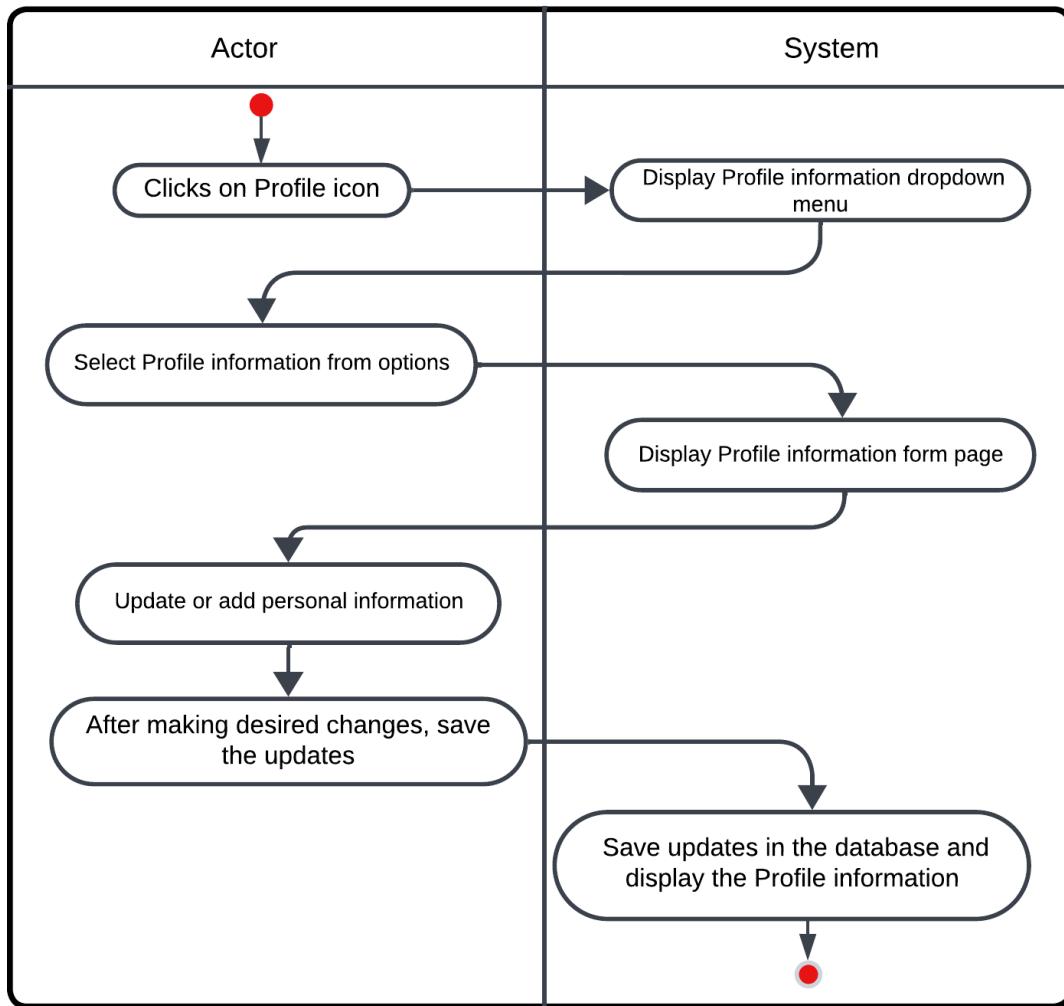


Figure 2.14: manage user profile activity diagram

- Figure 2.16 shows activity diagram steps for a process of sign out by User Actor

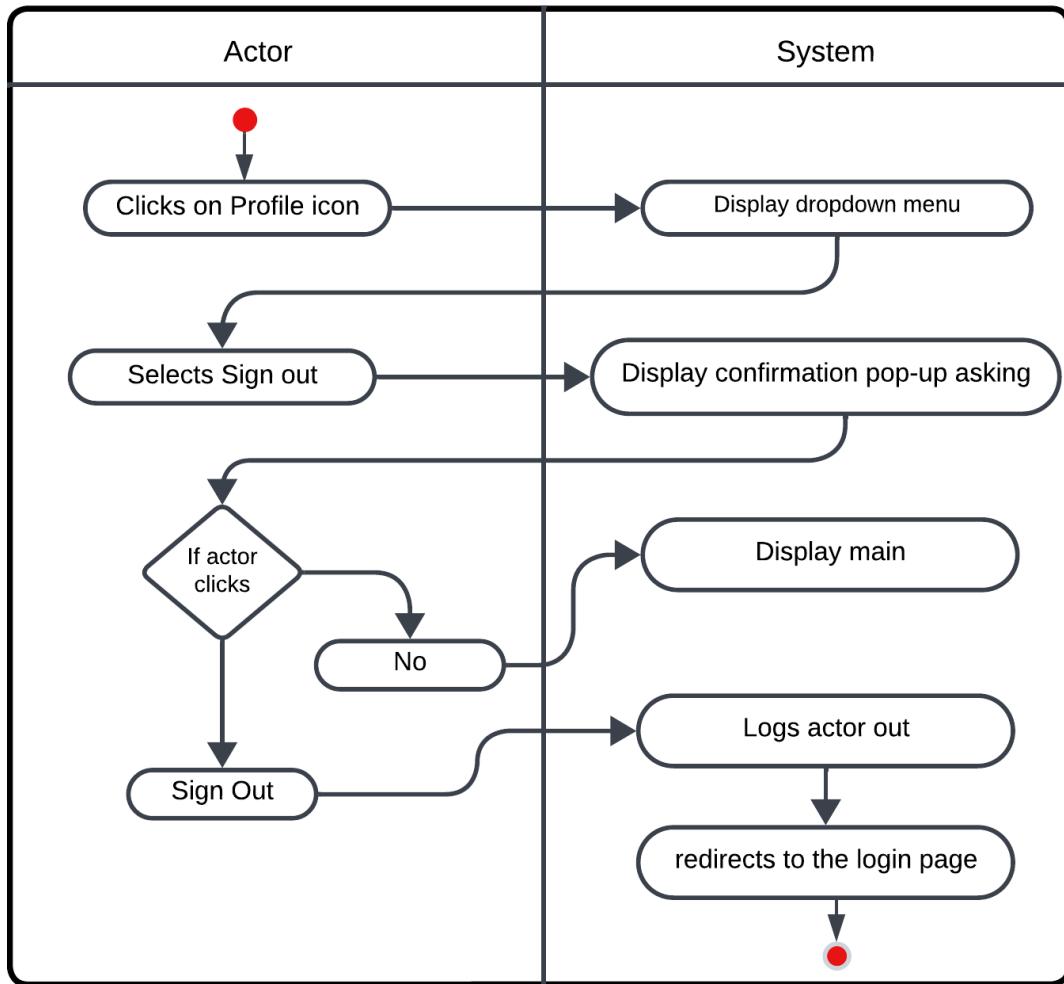


Figure2.15 : sign out activity diagram

- Figure 2.17 shows activity diagram steps for a process of manage members accounts by Admin Actor

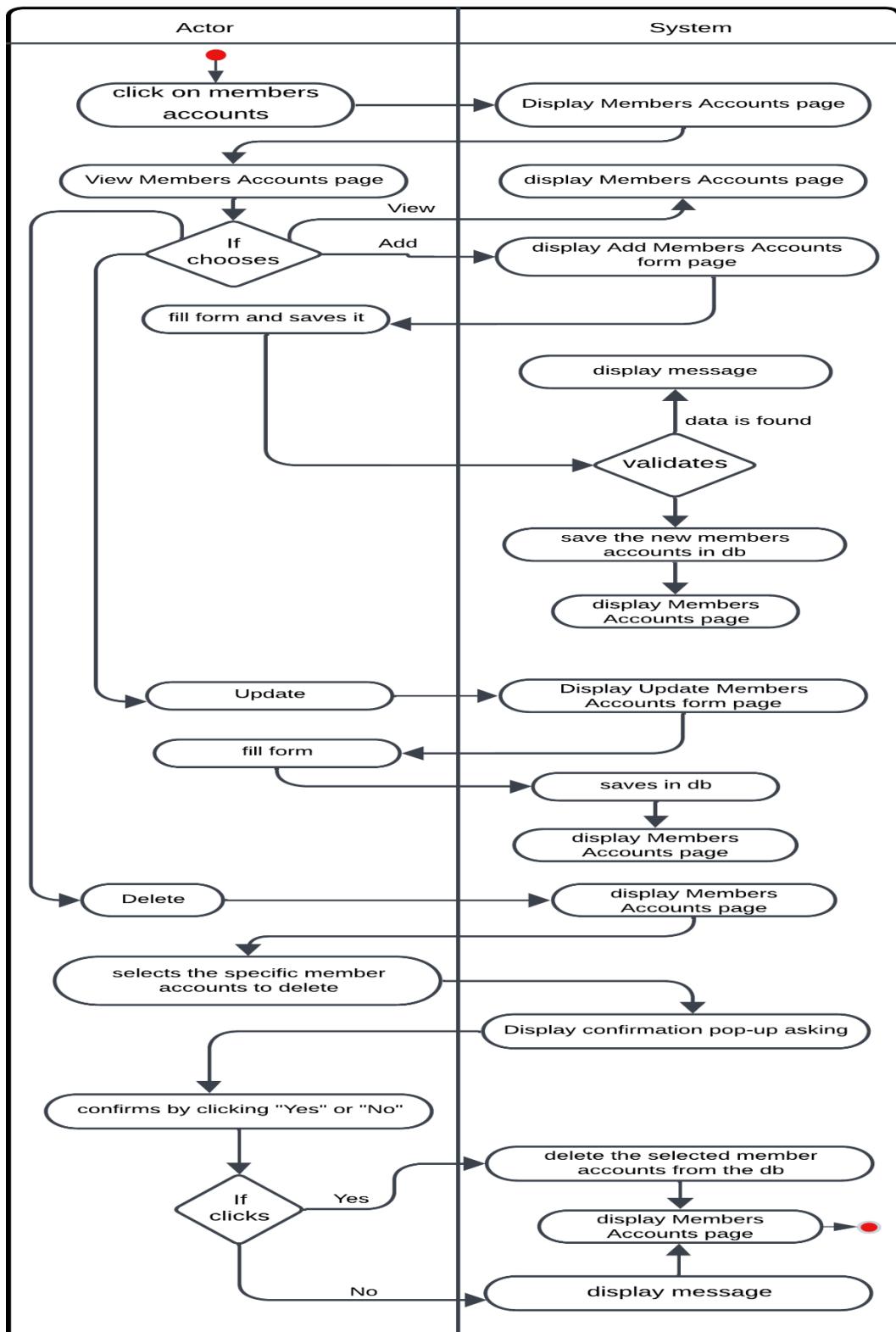


Figure 2.16: manage members accounts activity diagram

- Figure 2.18 shows activity diagram steps for a process of create reports by Admin Actor

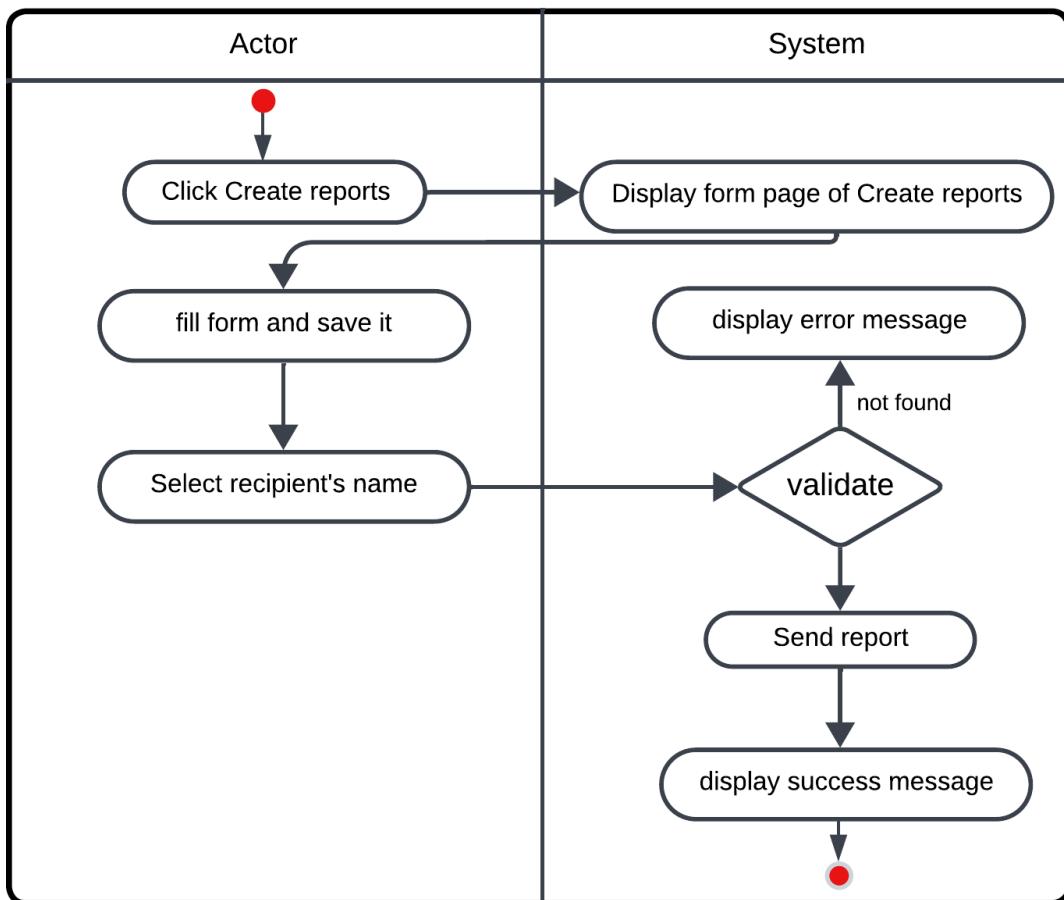


Figure 2.17: Create reports activity diagram

- Figure 2.19 shows activity diagram steps for a process of manage system data by Admin Actor

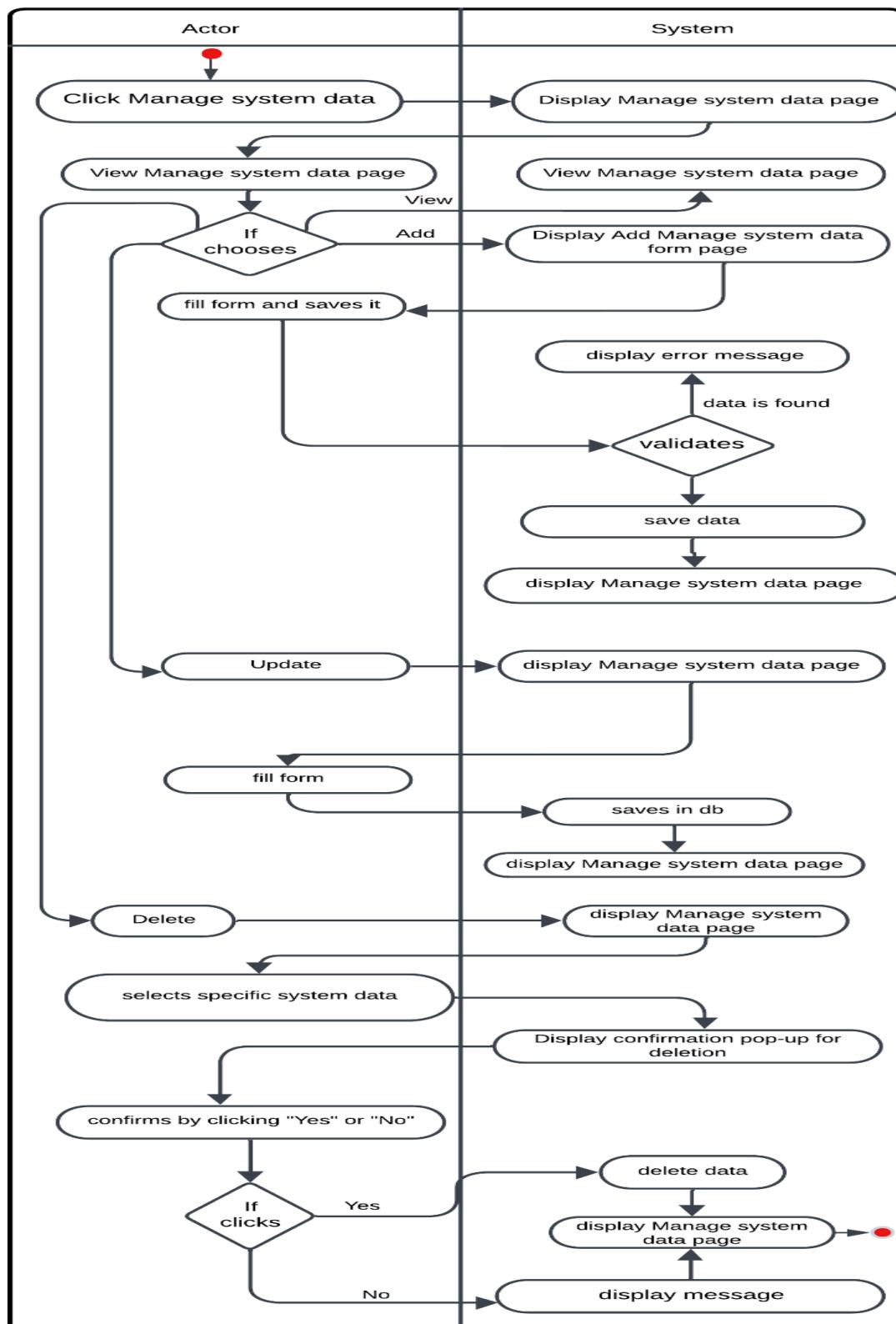


Figure 2.18: Manage system data activity diagram

- Figure 2.20 shows activity diagram steps for a process of monitor and generate the reports by President of the University Actor

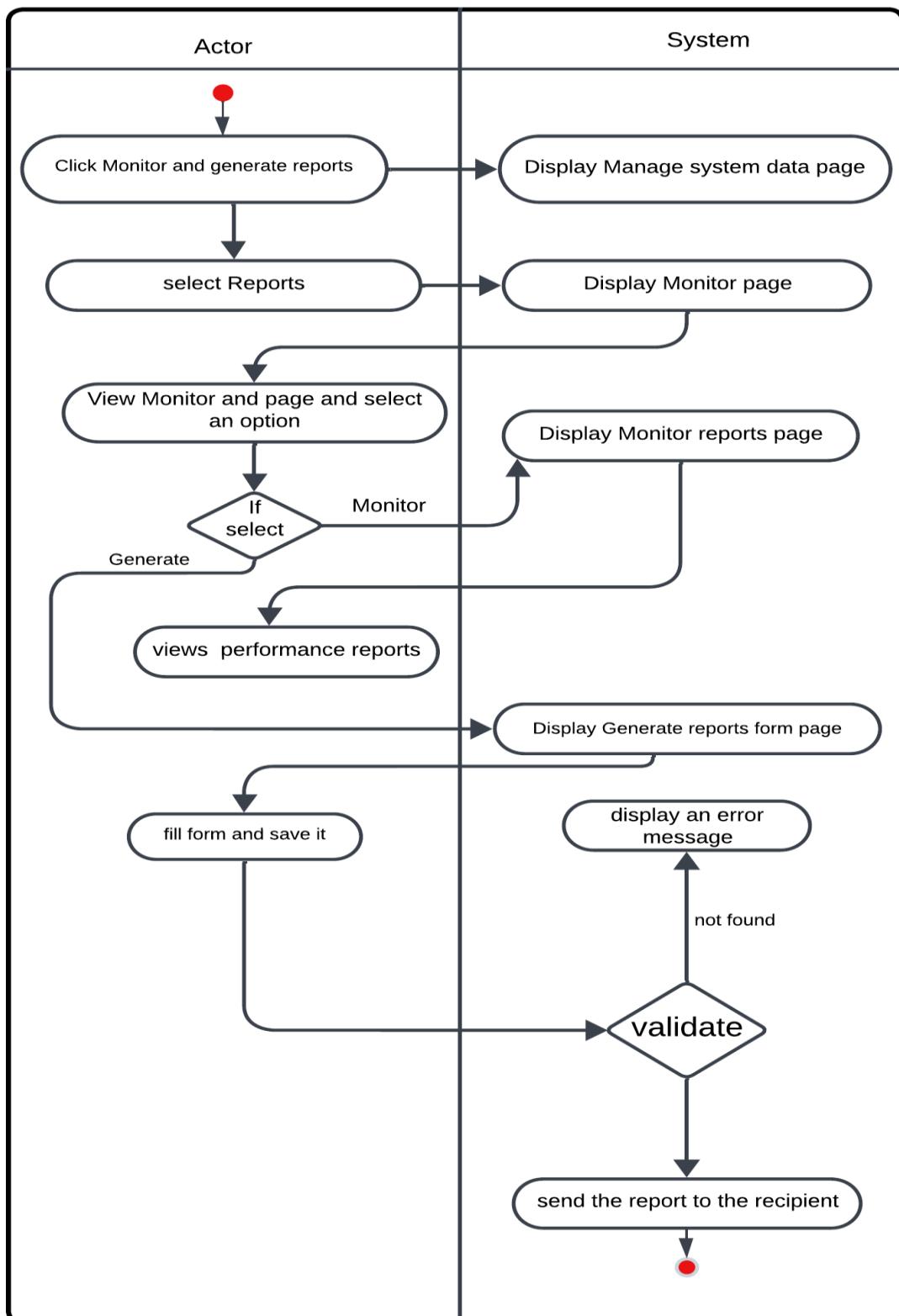


Figure 2.19: Monitor and generate the reports activity diagram

- Figure 2.21 shows activity diagram steps for a process of monitor and generate the reports by Dean of the College Actor

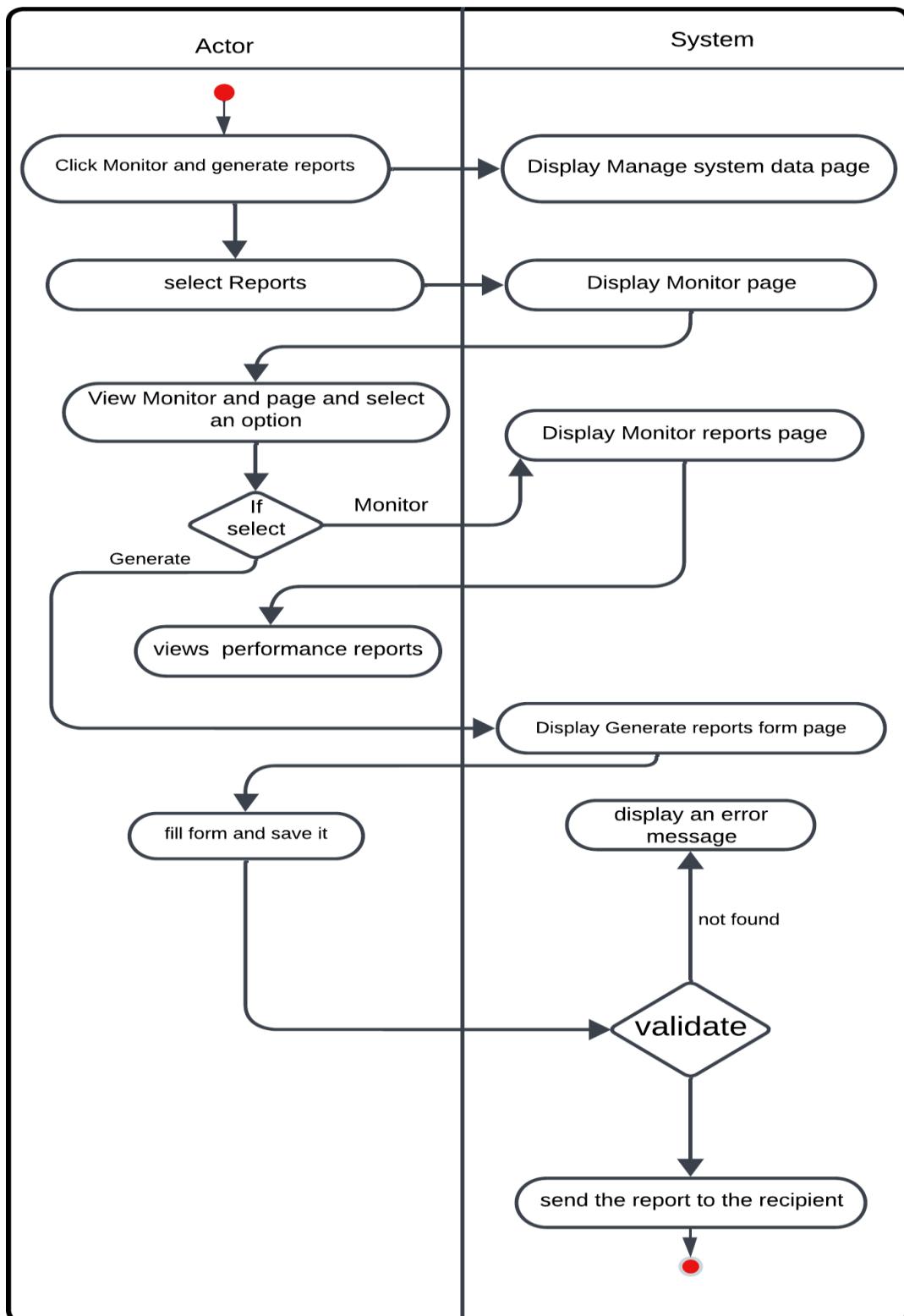


Figure 2.20: Monitor and generate the reports activity diagram

- Figure 2.22 shows activity diagram steps for a process of manage program basic information by Program Coordinator Actor

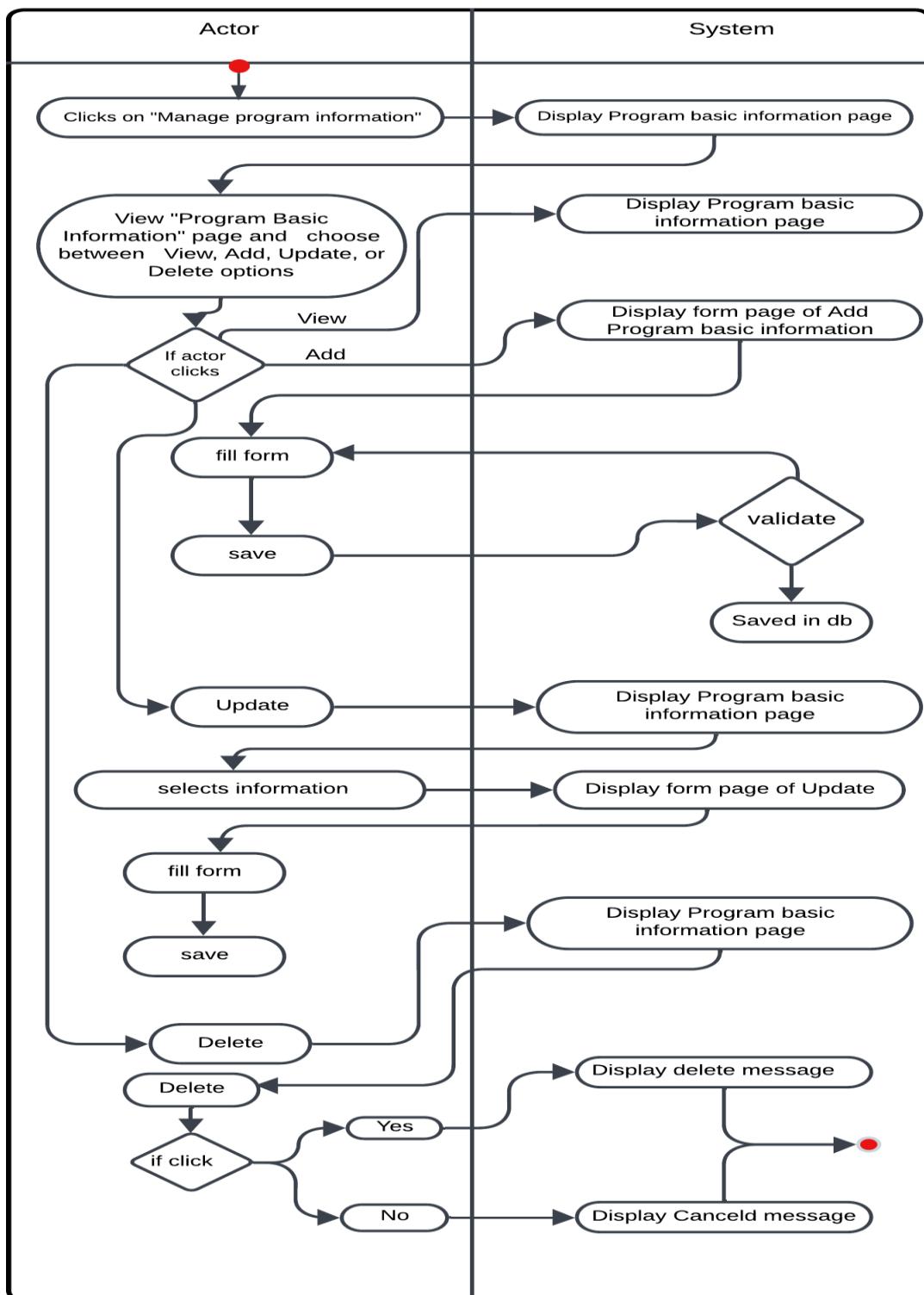


Figure 2.21: Manage program basic information activity diagram

- Figure 2.23 shows activity diagram steps for a process of manage program learning outcomes by Program Coordinator Actor

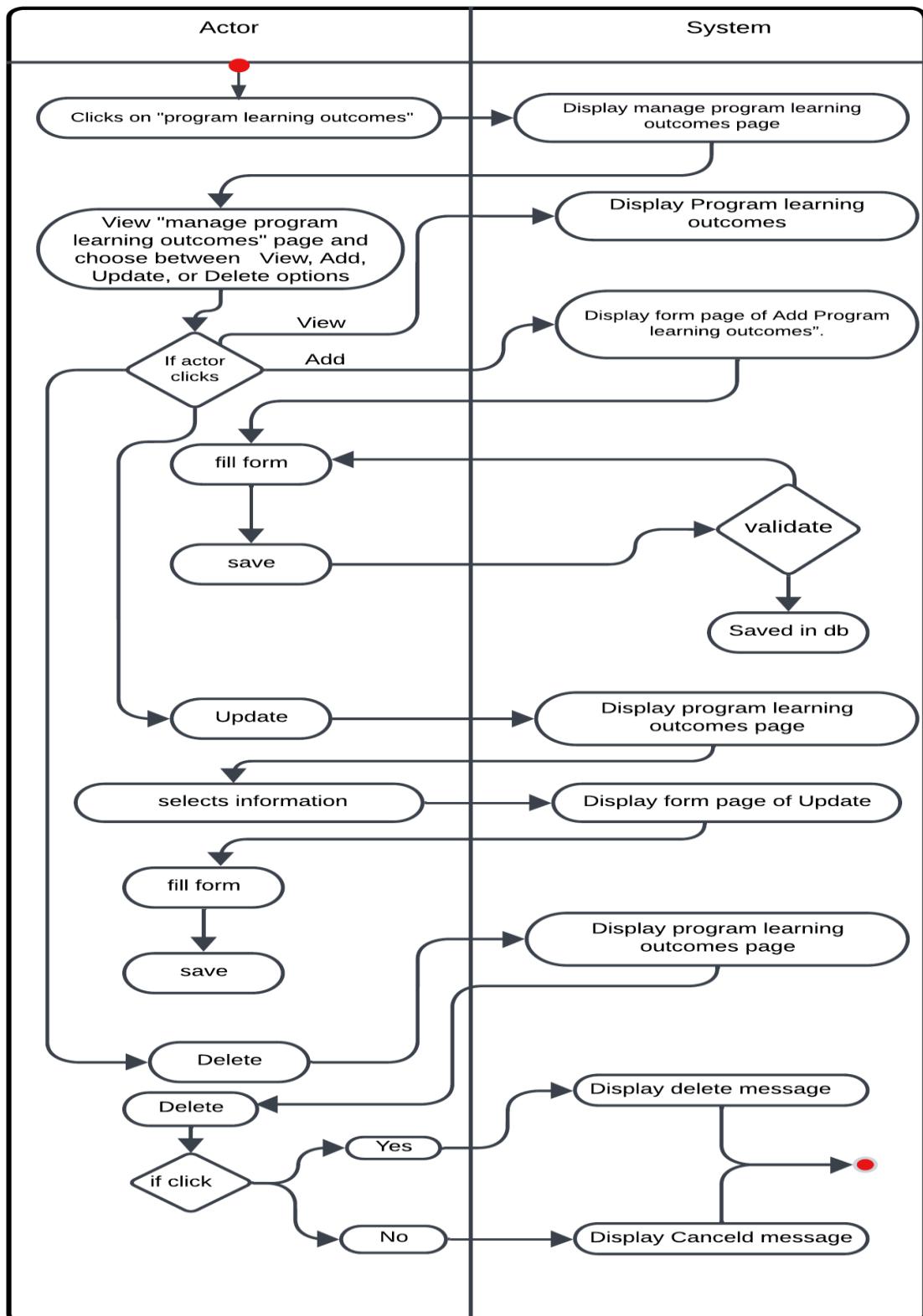


Figure 2.22: Manage program learning outcomes activity diagram

- Figure 2.24 shows activity diagram steps for a process of receiving reports by Program Coordinator Actor

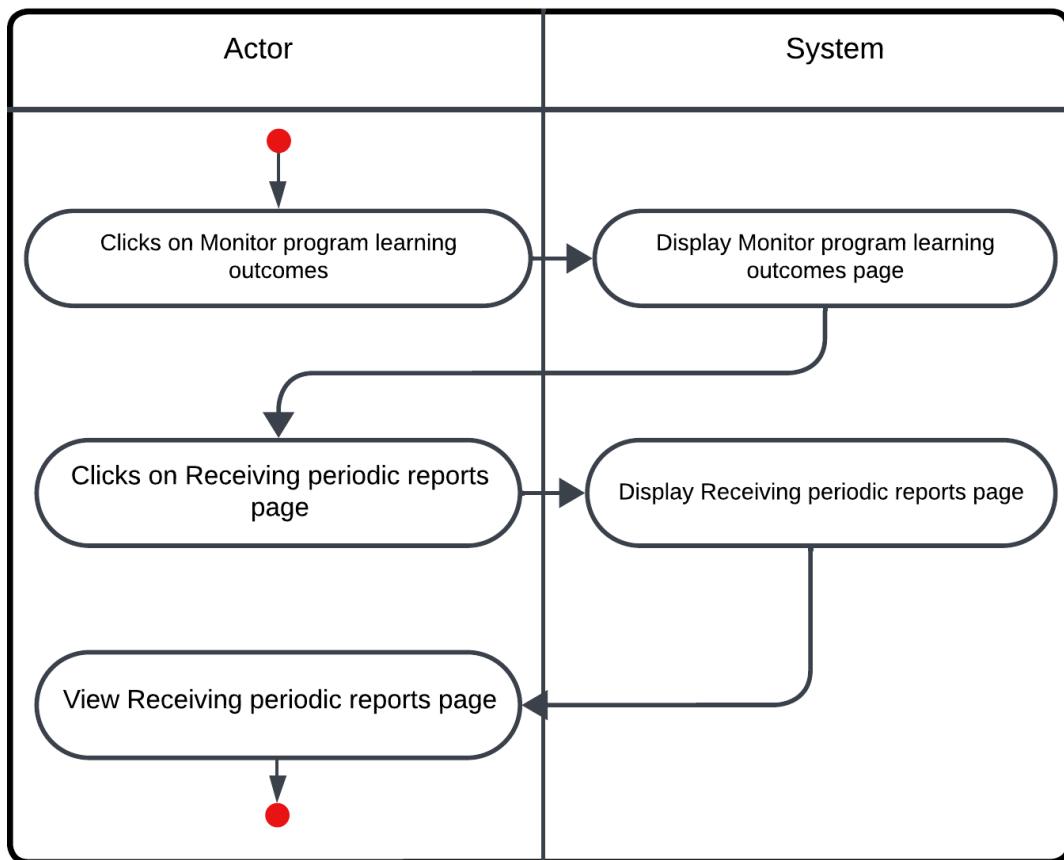


Figure 2.23: Receiving reports activity diagram

- Figure 2.25 shows activity diagram steps for a process of generate reports by Program Coordinator Actor

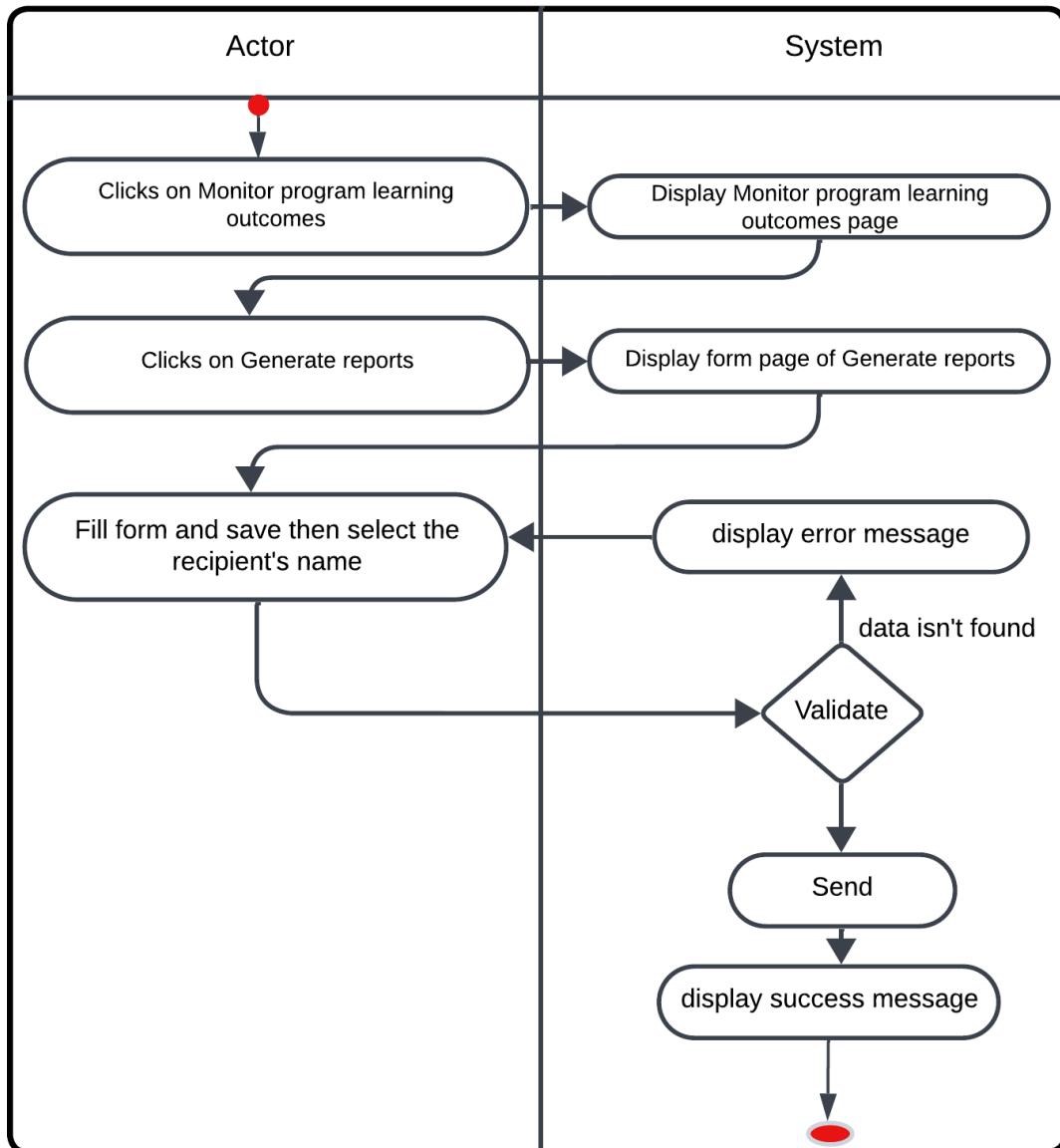


Figure 2.24: Generate reports activity diagram

- Figure 2.26 shows activity diagram steps for a process of monitor of students by Academic advisor Actor

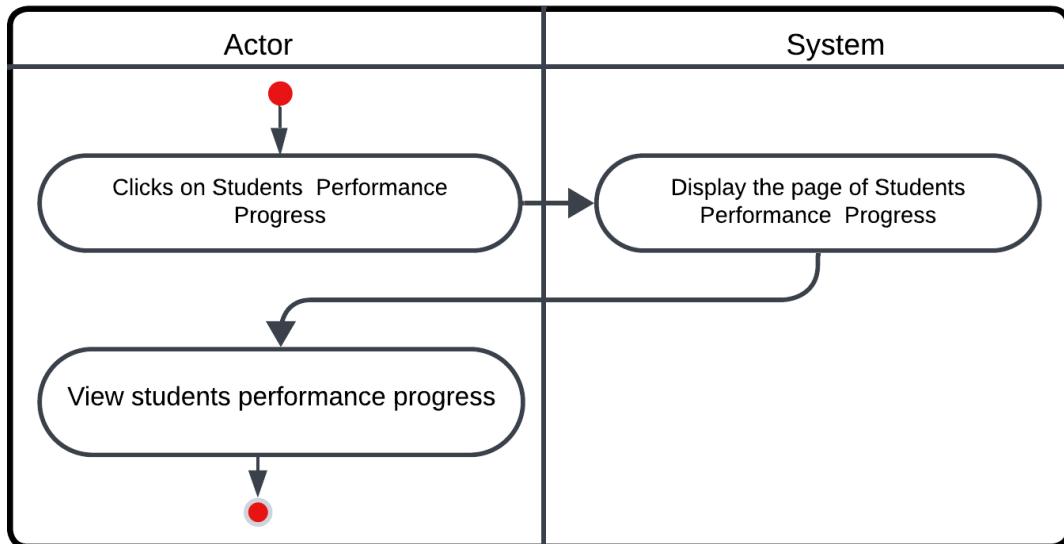


Figure 2.25: Monitor of students activity diagram

- Figure 2.27 shows activity diagram steps for a process of view student profiles by Academic advisor Actor

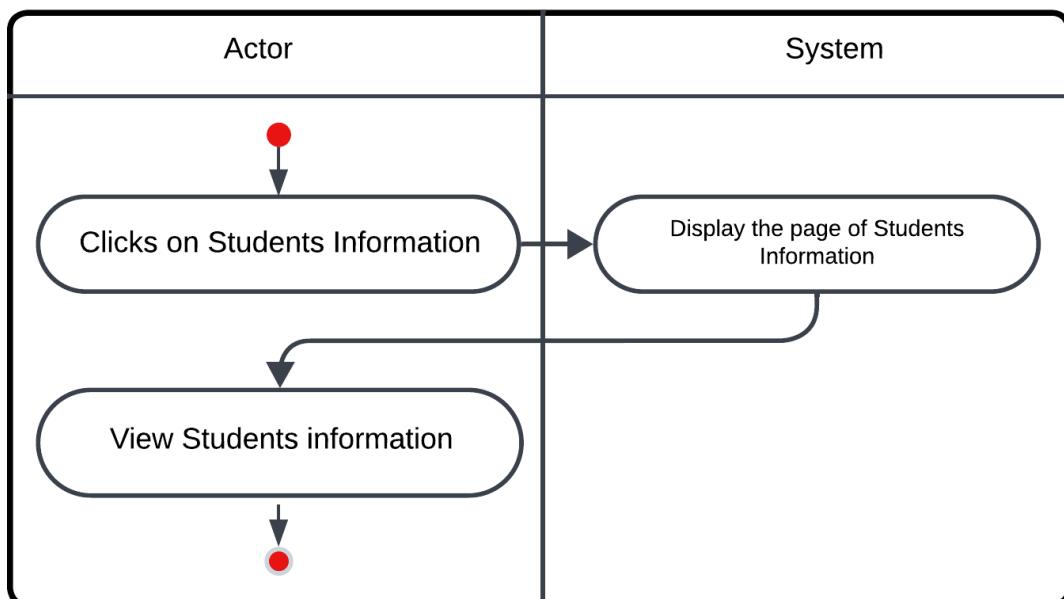


Figure 2.26: View student profiles activity diagram

- Figure 2.28 shows activity diagram steps for a process of communicate with students by Academic advisor Actor

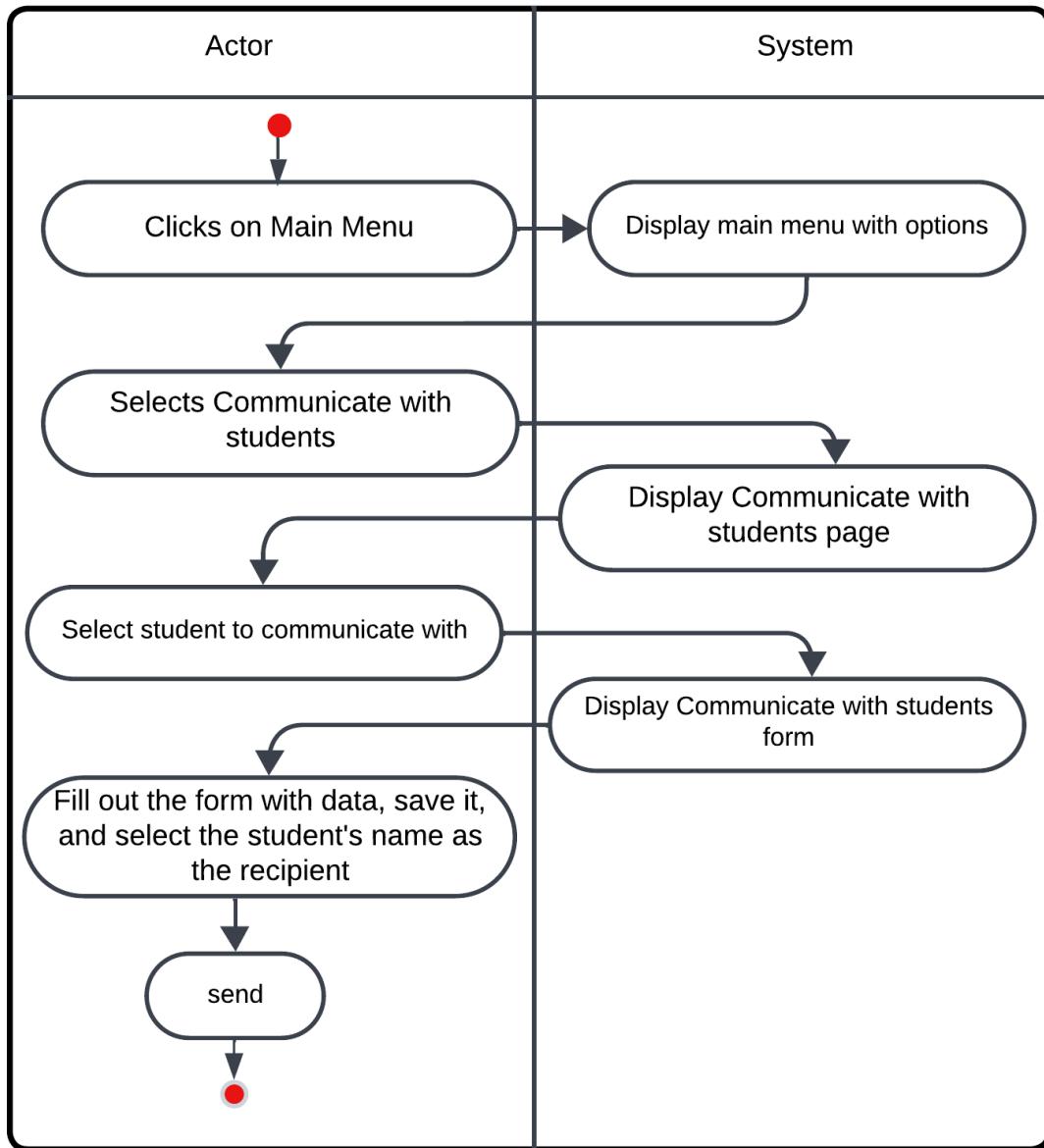


Figure 2.27: Communicate with students activity diagram

- Figure 2.29 shows activity diagram steps for a process of identify students at risk by Academic advisor Actor

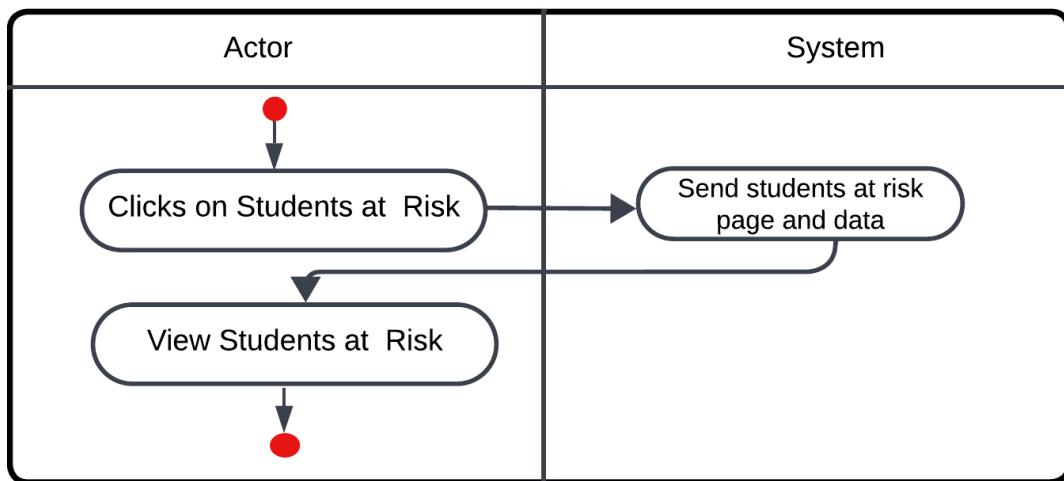


Figure 2.28: Identify students at risk activity diagram

- Figure 2.30 shows activity diagram steps for a process of generate reports by Academic advisor Actor

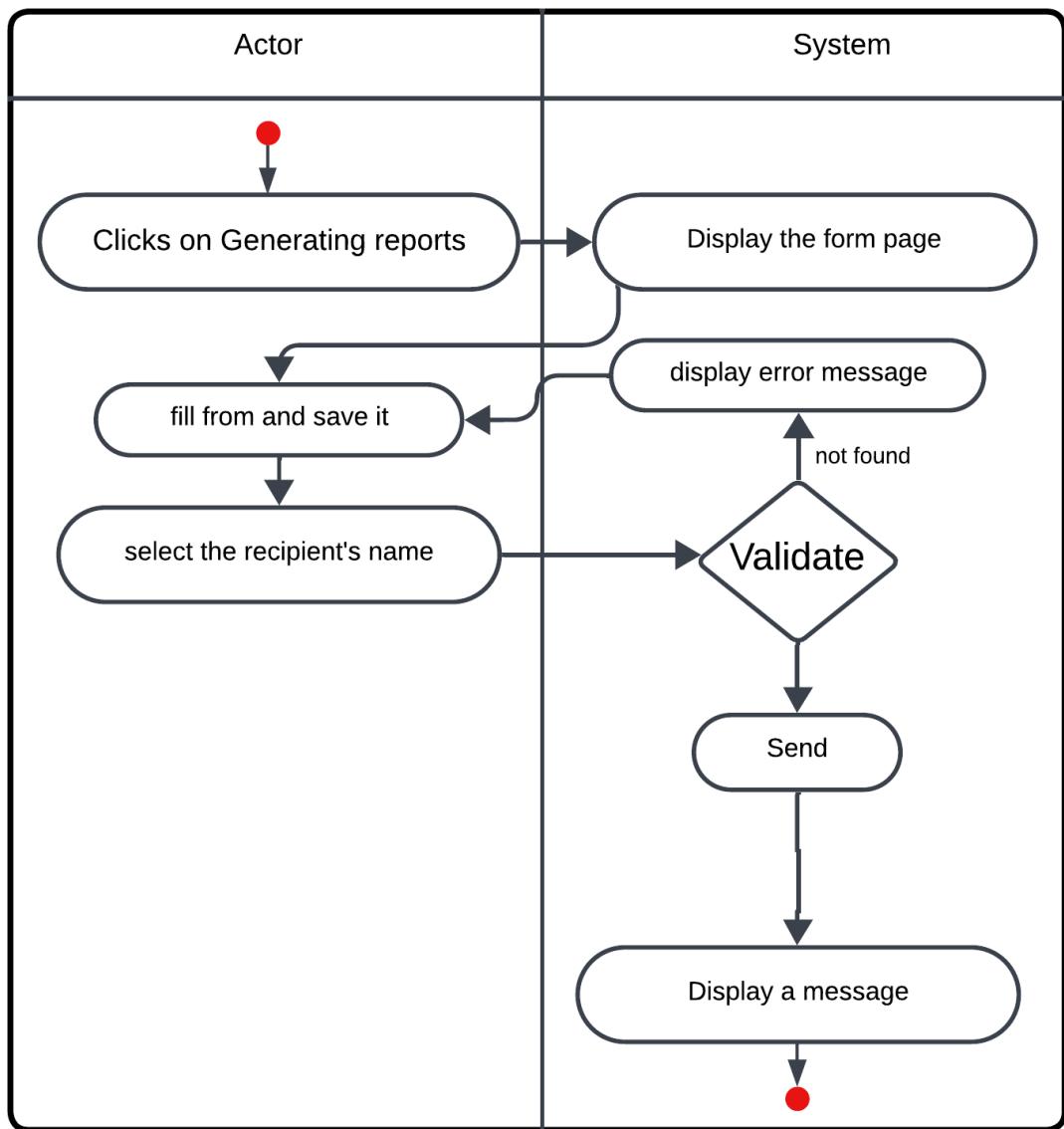


Figure 2.29: Generate reports activity diagram

- Figure 2.31 shows activity diagram steps for a process of monitor of students by Faculty Member Actor

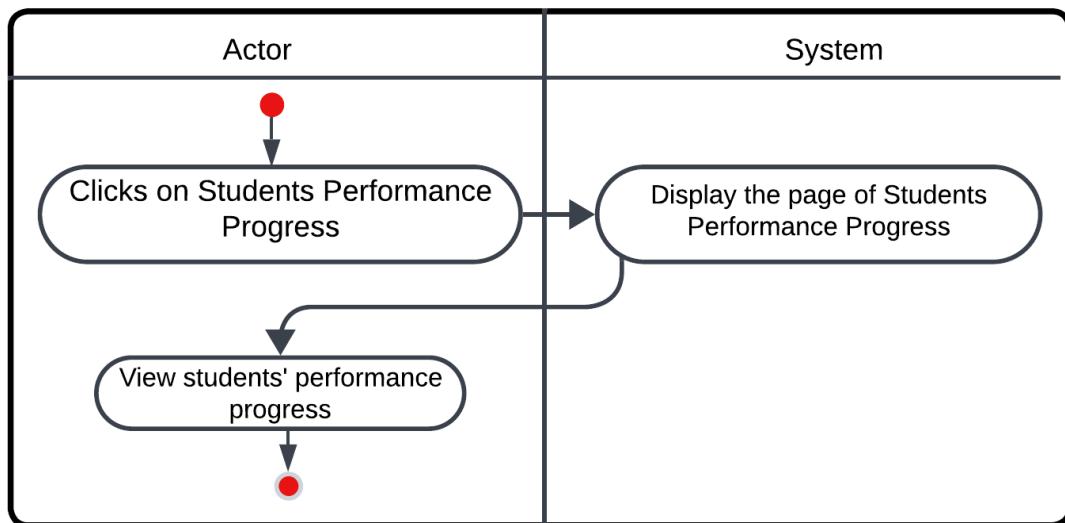


Figure 2.30: Monitor of students activity diagram

- Figure 2.32 shows activity diagram steps for a process of manage grades by Faculty Member Actor

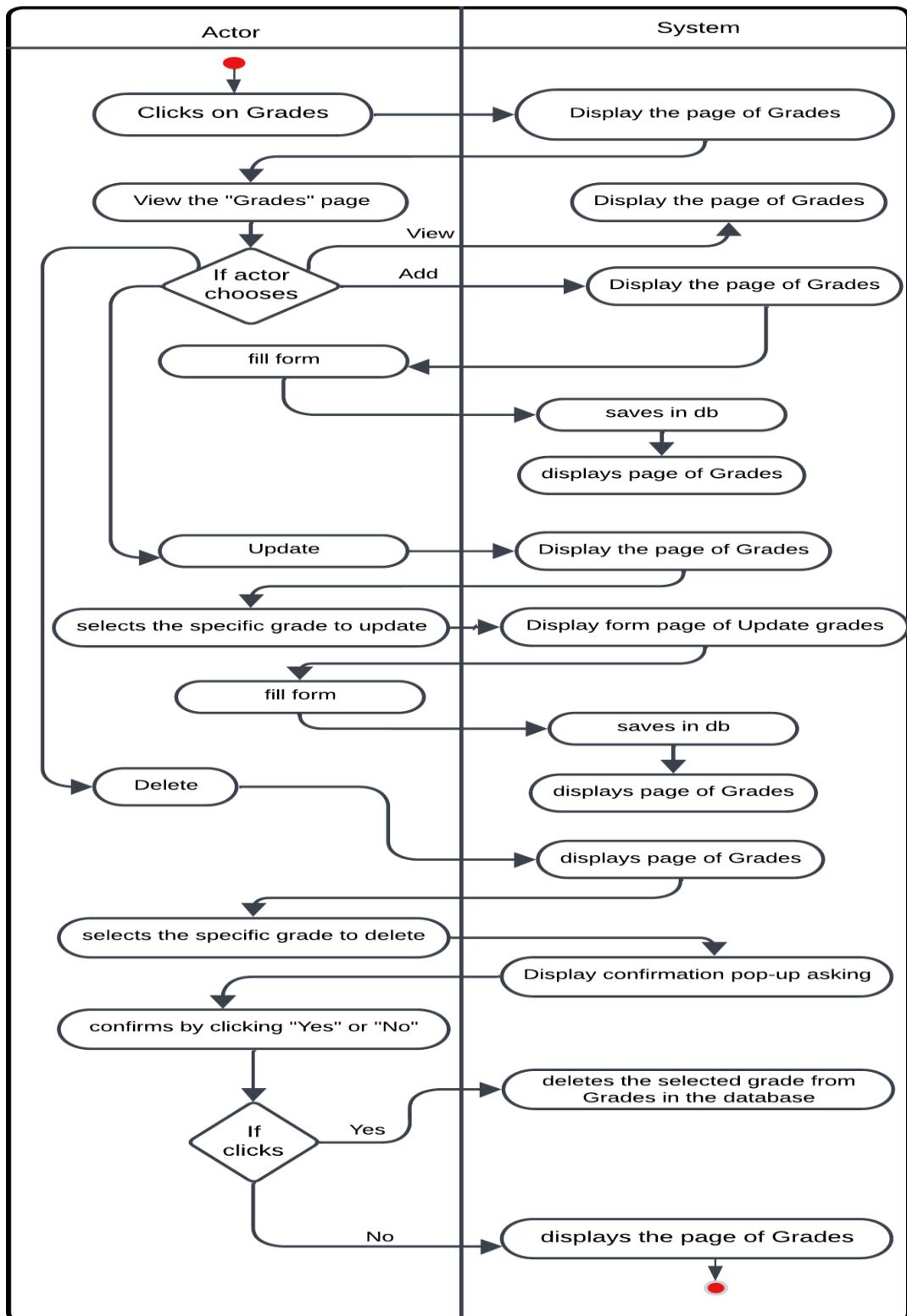


Figure 2.31: Manage grades activity diagram

- Figure 2.33 shows activity diagram steps for a process of communicate with students by Faculty Member Actor

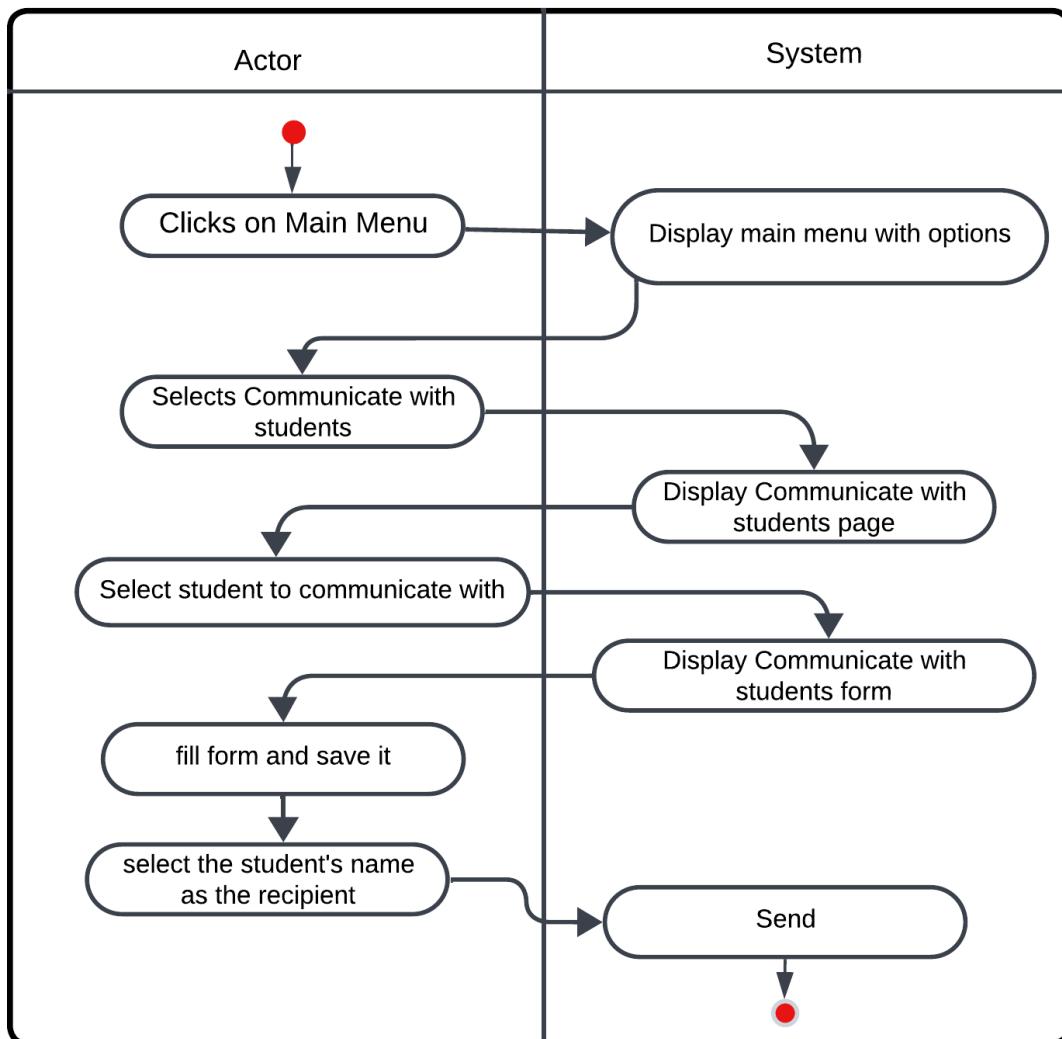


Figure 2.32: Communicate with students activity diagram

- Figure 2.34 shows activity diagram steps for a process of identify students at risk by Faculty Member Actor

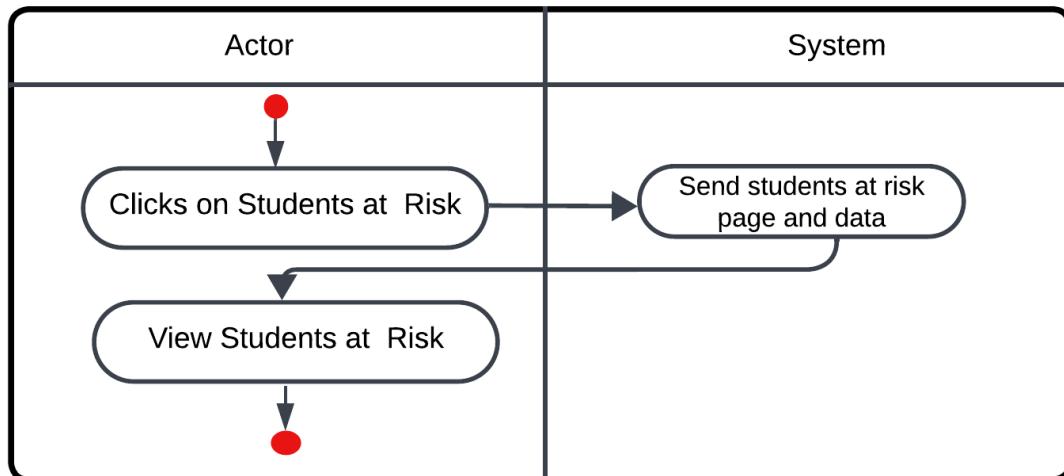


Figure 2.33: Identify students at risk activity diagram

- Figure 2.35 shows activity diagram steps for a process of view subject information by Student Actor

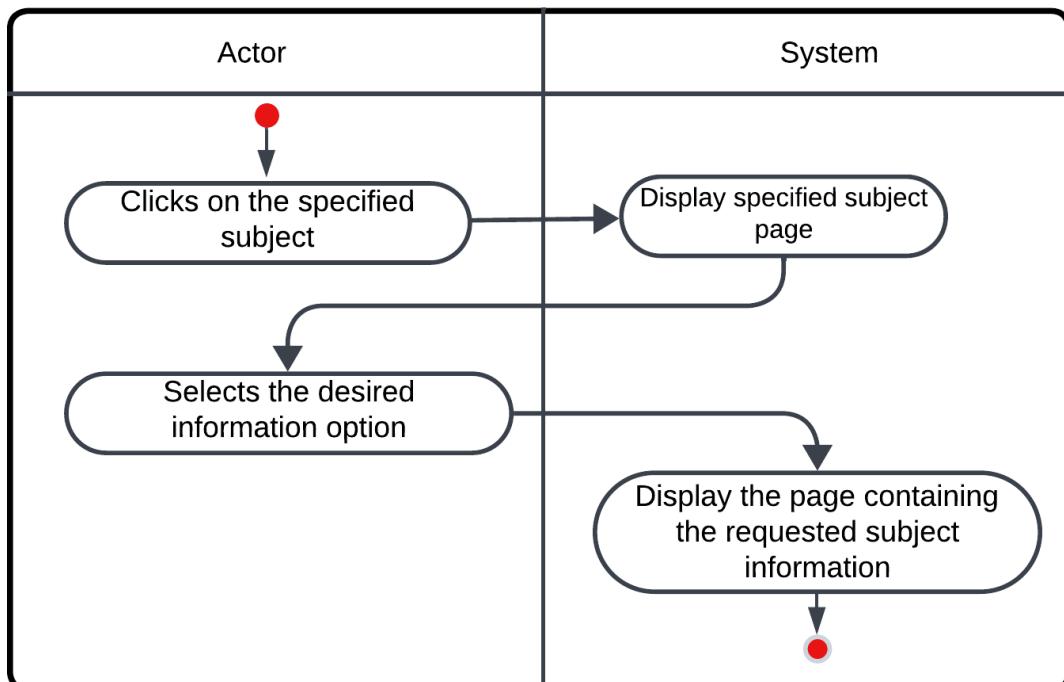


Figure 2.34: view subject information activity diagram

- Figure 2.36 shows activity diagram steps for a process of view the predicted academic performance by Student Actor

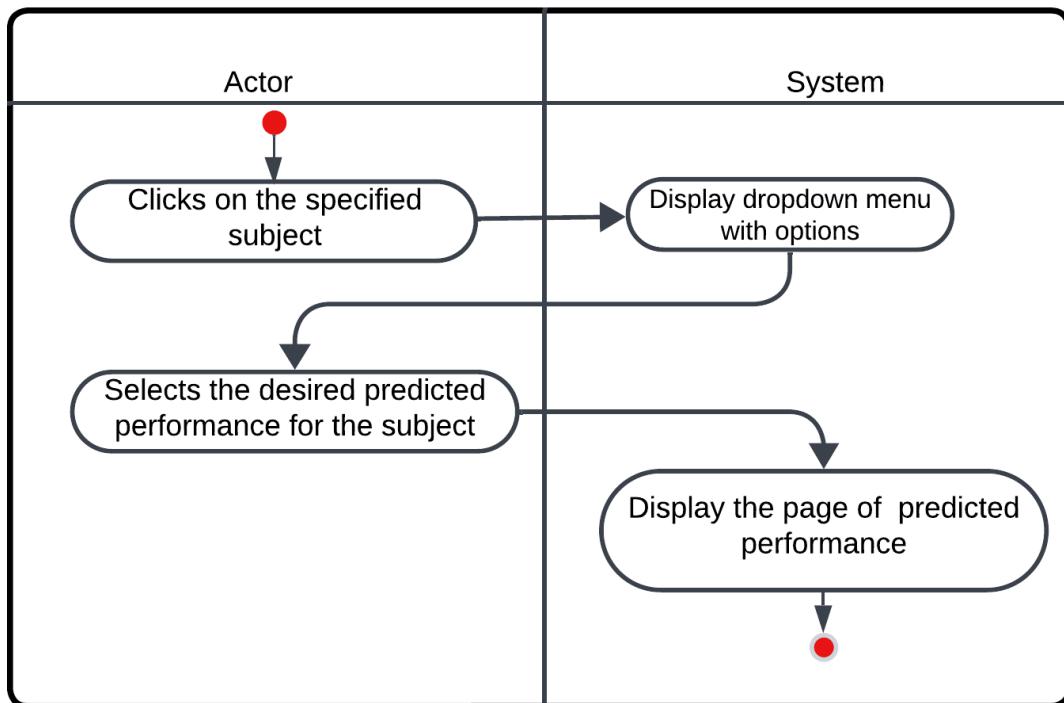


Figure 2.35: view the predicted academic performance activity diagram

- Figure 2.37 shows activity diagram steps for a process of view previous academic performance by Student Actor

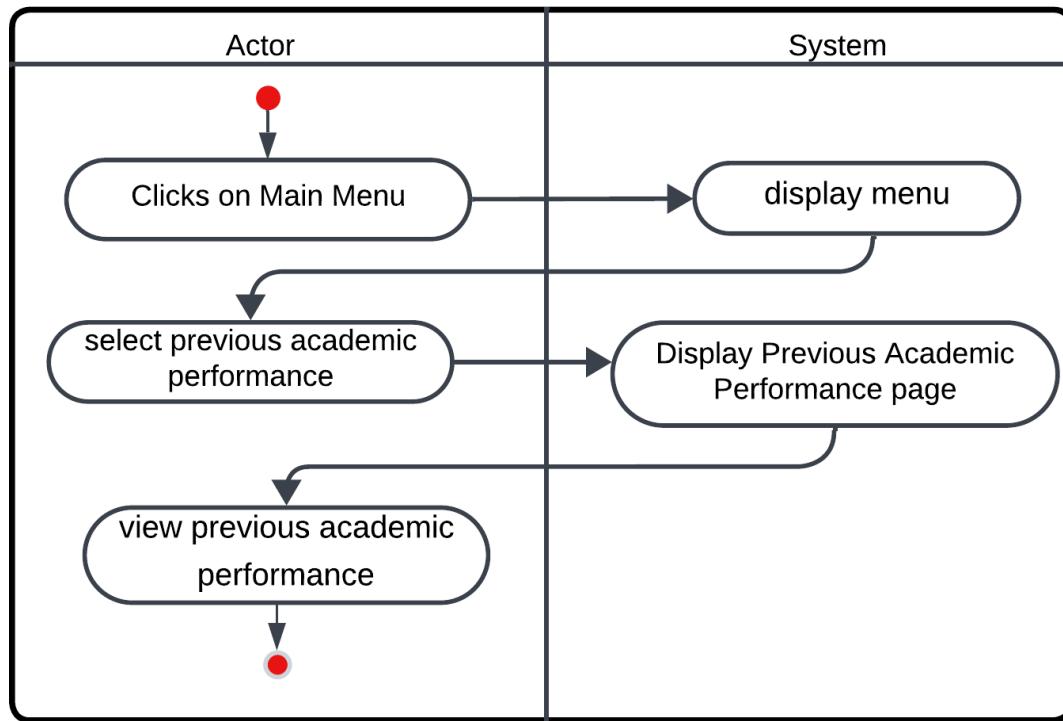


Figure 2.36: View previous academic performance activity diagram

- Figure 2.38 shows activity diagram steps for a process of identify subjects requiring improvement by Student Actor

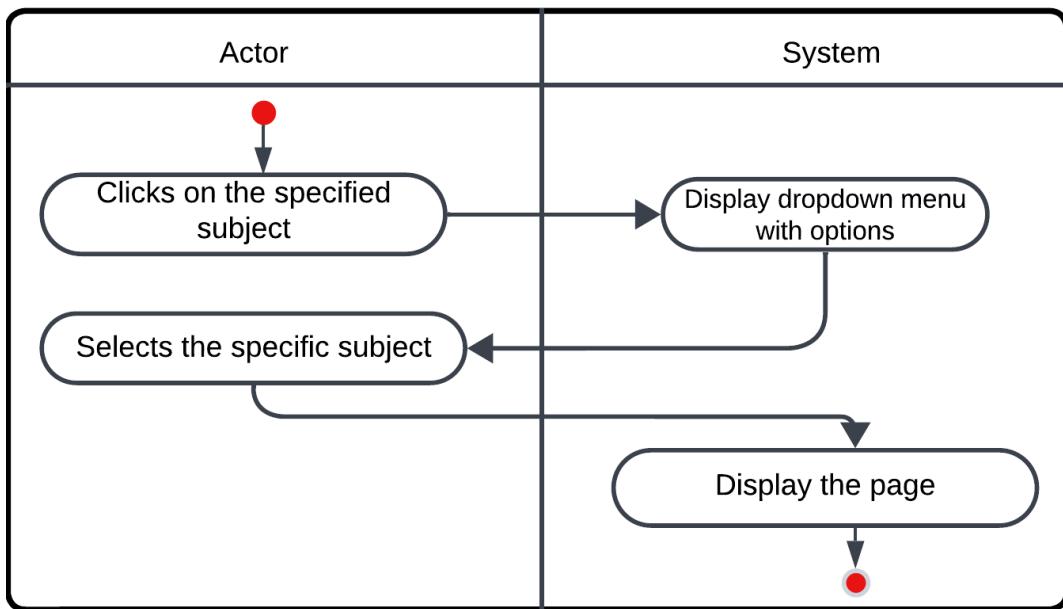


Figure 2.37: Identify subjects requiring improvement activity diagram

- Figure 2.39 shows activity diagram steps for a process of communicate with the Academic Advisor or faculty member by Student Actor

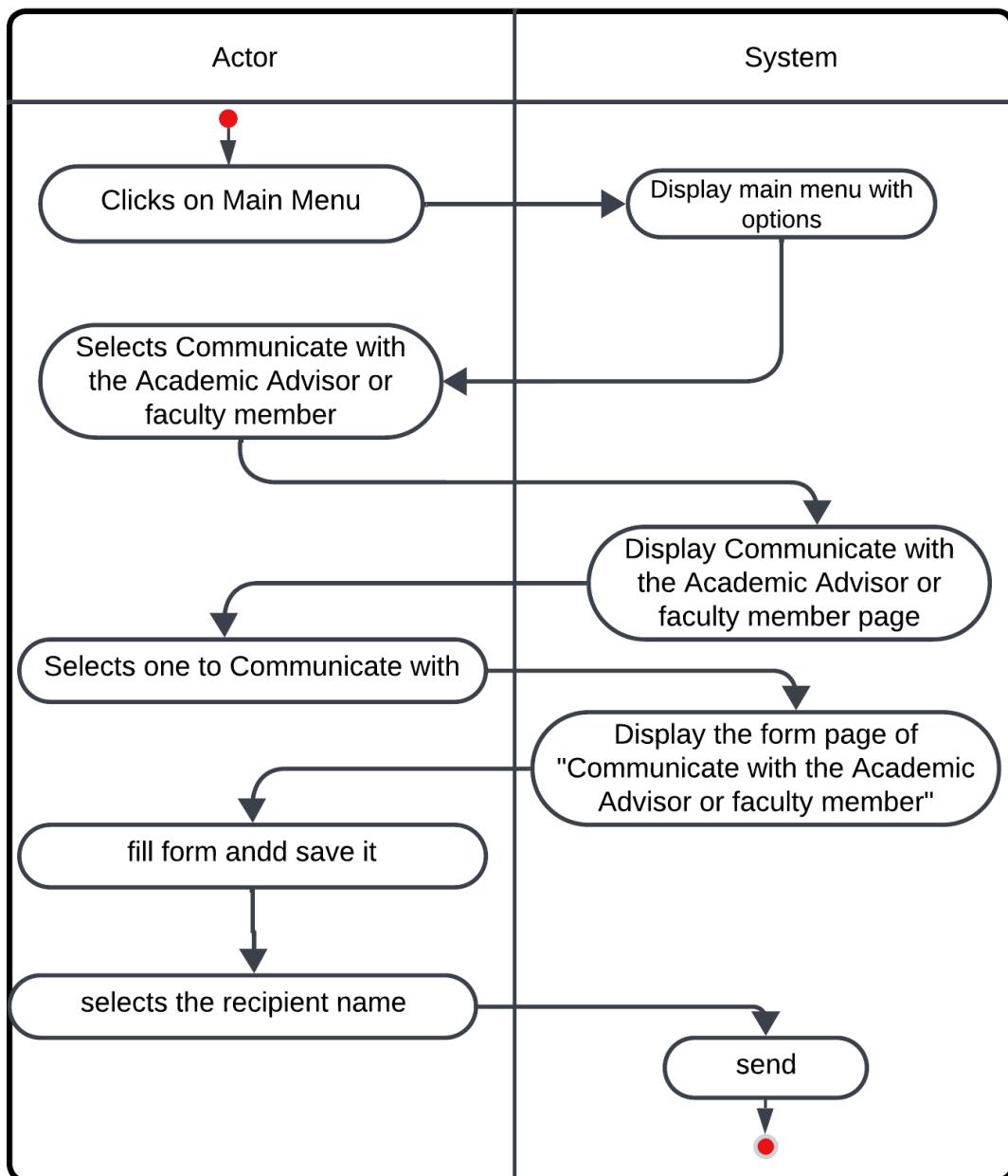


Figure 2.38: Communicate with the Academic Advisor or faculty member activity diagram.

2.6.4 Sequence Diagram

is a graphical representation in the Unified Modeling Language (UML) that illustrates the interactions and order of messages between objects or components within a system. It depicts the sequence of actions and communications, providing a visual representation of the chronological order of interactions between various system elements.

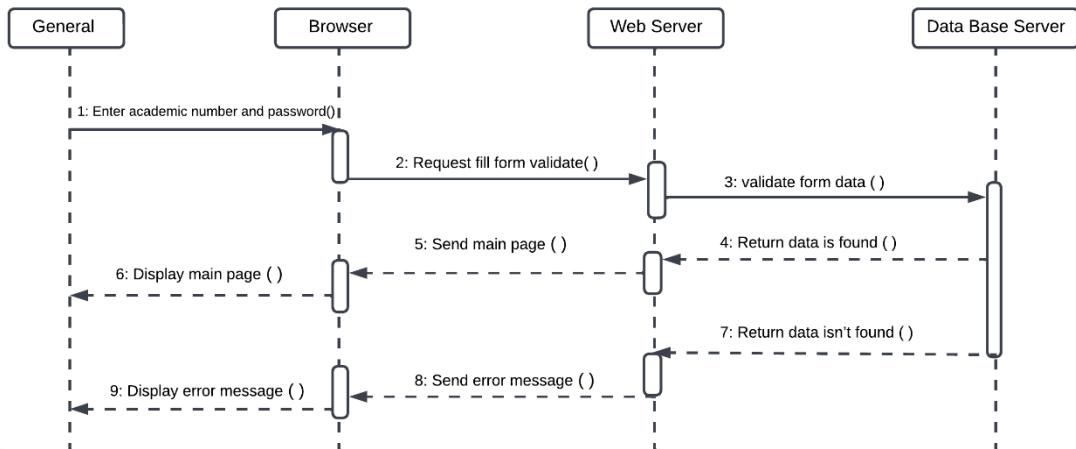


Figure 2.39: Log in Sequence diagram

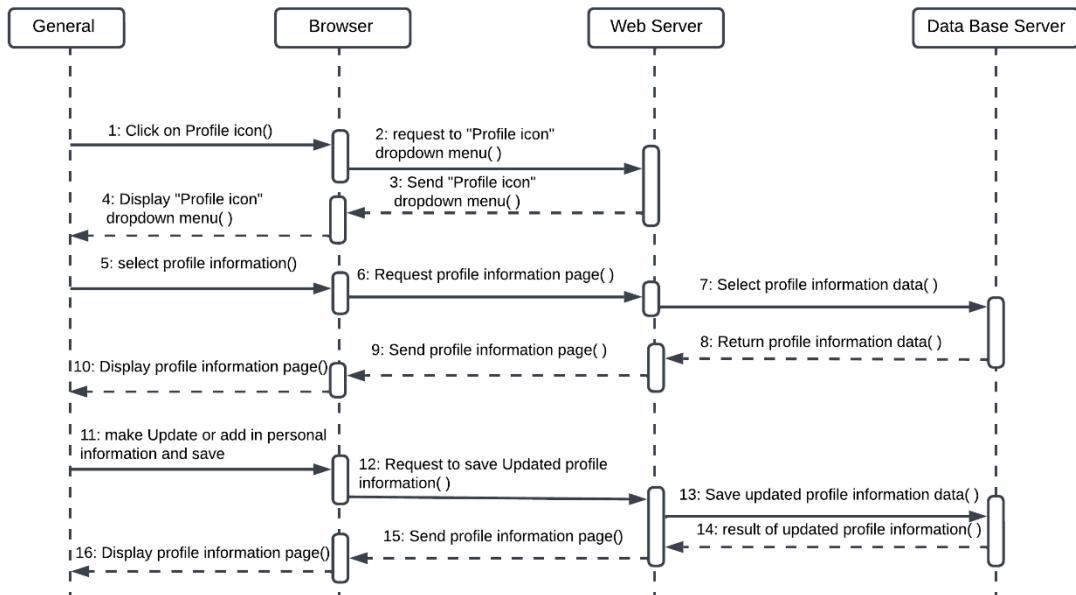


Figure 2.40: Manage User Profile Sequence diagram

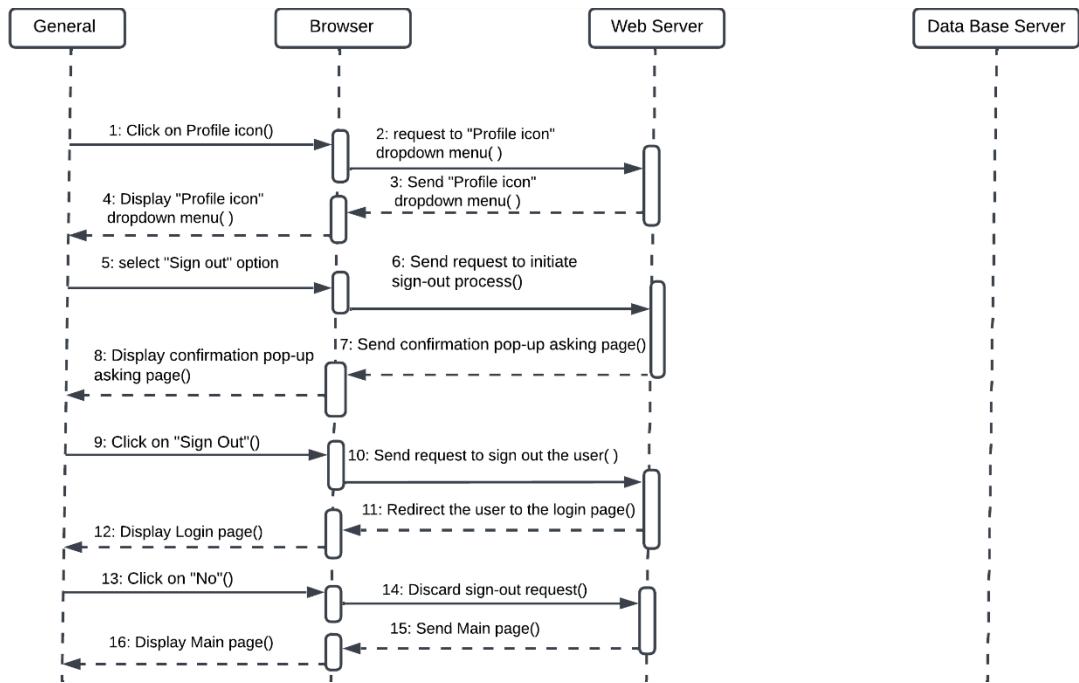


Figure 2.41: Sign out Sequence diagram

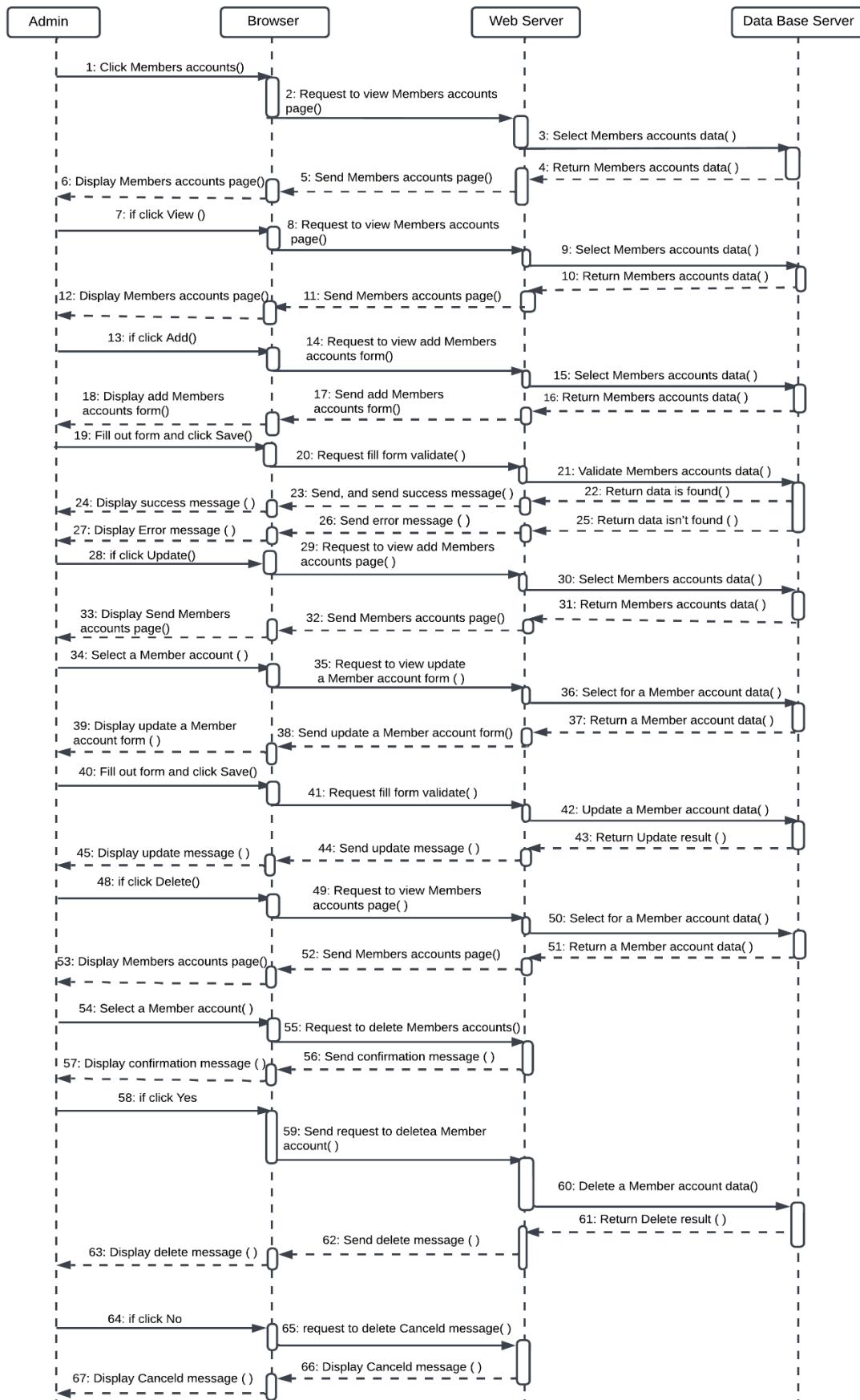


Figure 2.42: Manage members accounts Sequence diagram

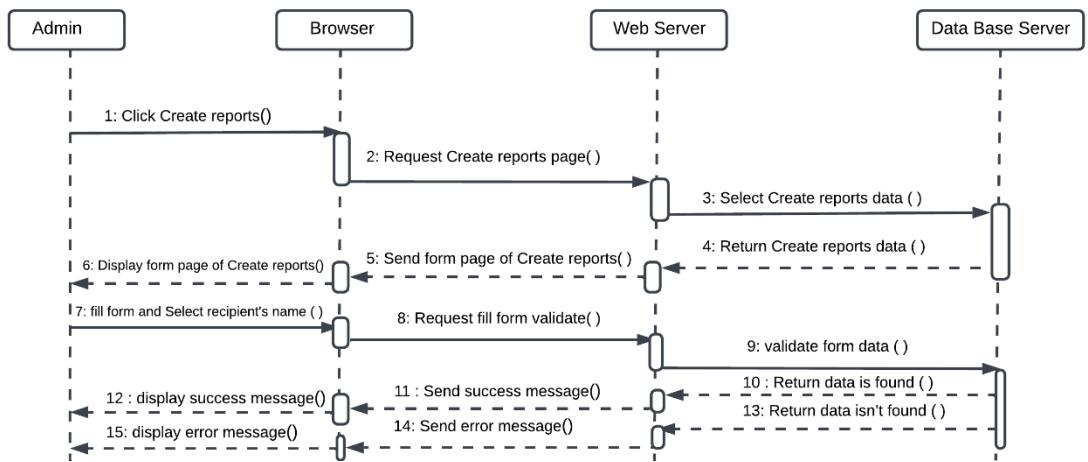


Figure 2.43: Create reports Sequence diagram



Figure 2.44: Manage the basic system data Sequence diagram

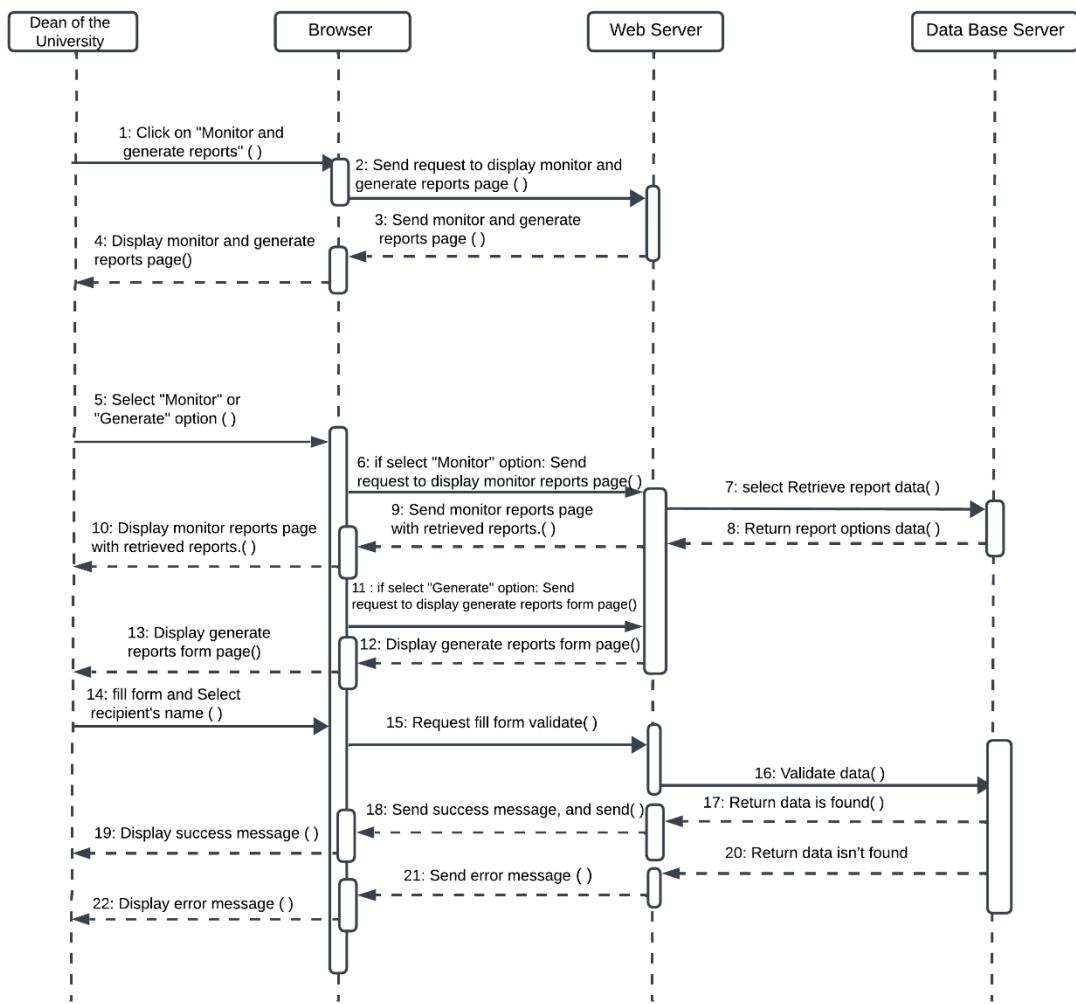


Figure 2.45: Monitor and generate reports Sequence diagram

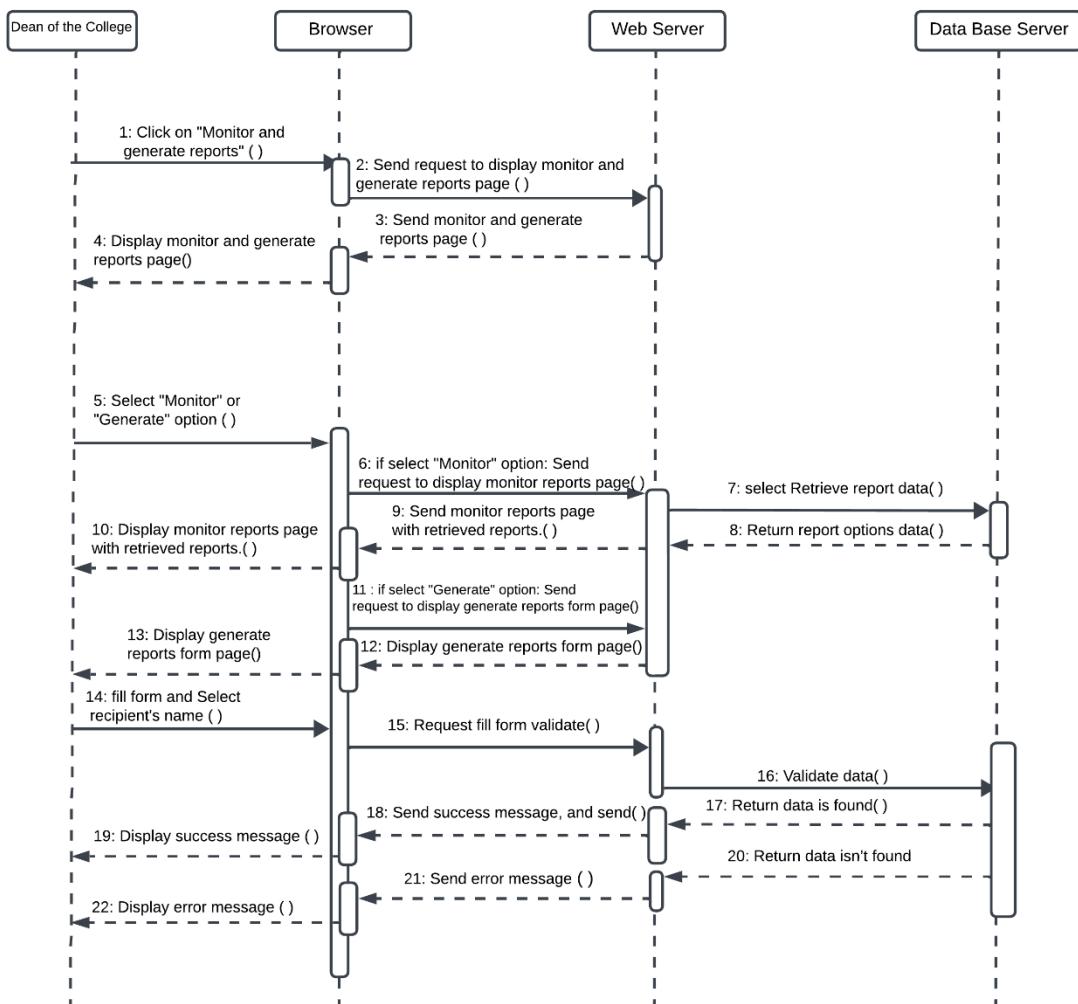


Figure 2.46: Monitor and generate reports Sequence diagram



Figure 2.47 : Manage program basic information Sequence diagram



Figure 2.48: Manage program learning outcomes Sequence diagram

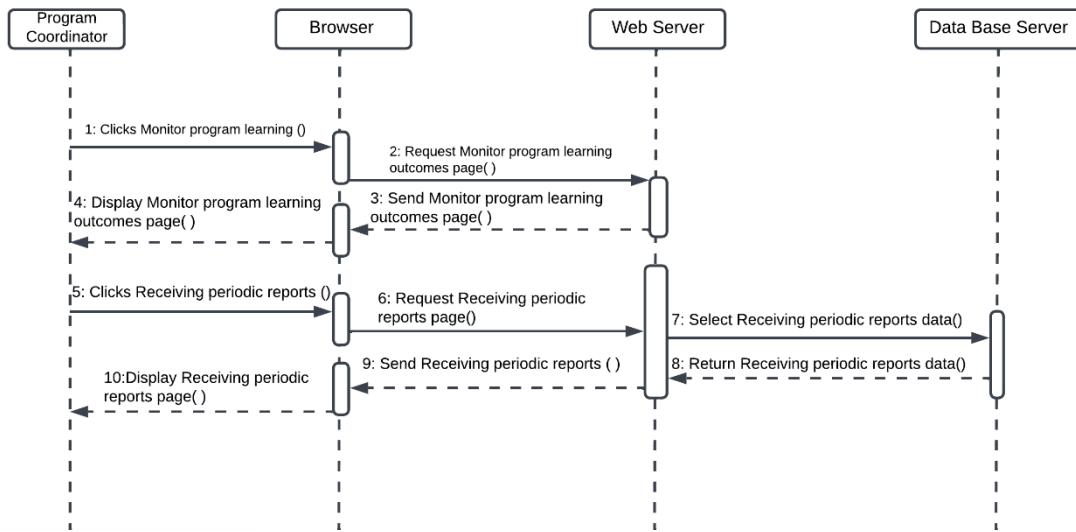


Figure 2.49: Receiving reports Sequence diagram

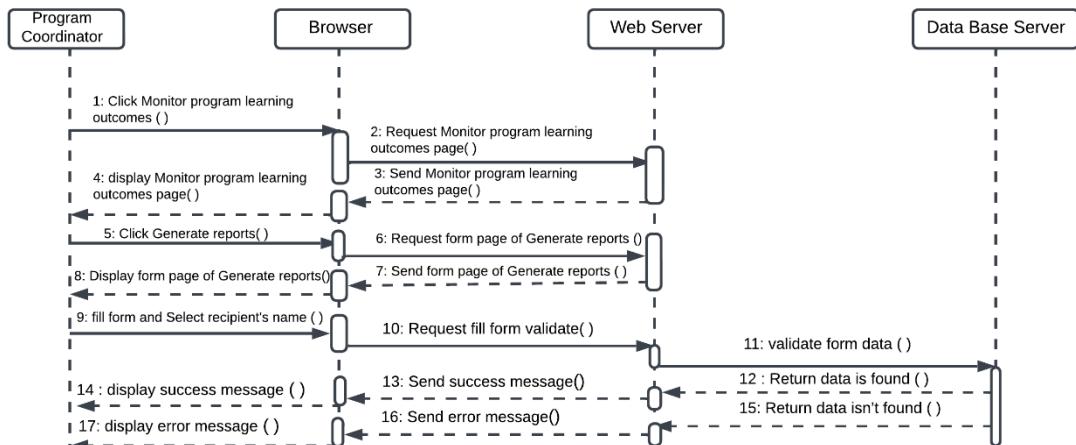


Figure 2.50: Generate reports Sequence diagram

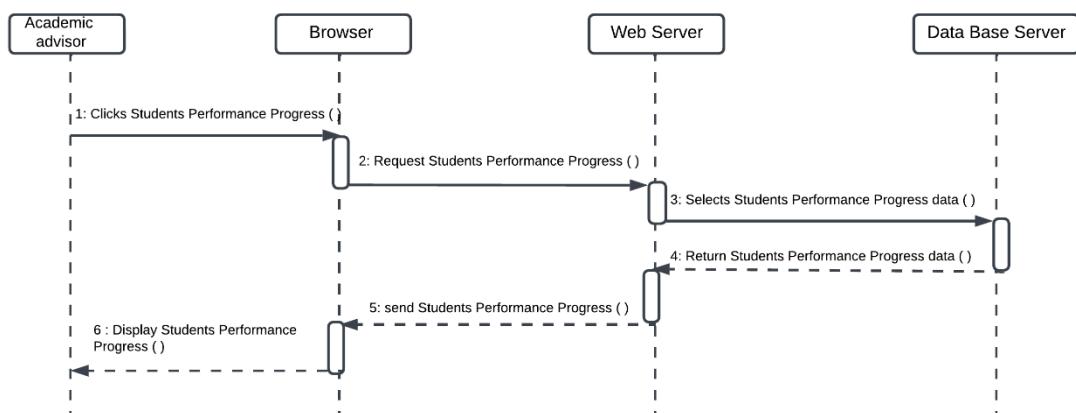


Figure 2.51: Monitor of students Sequence diagram

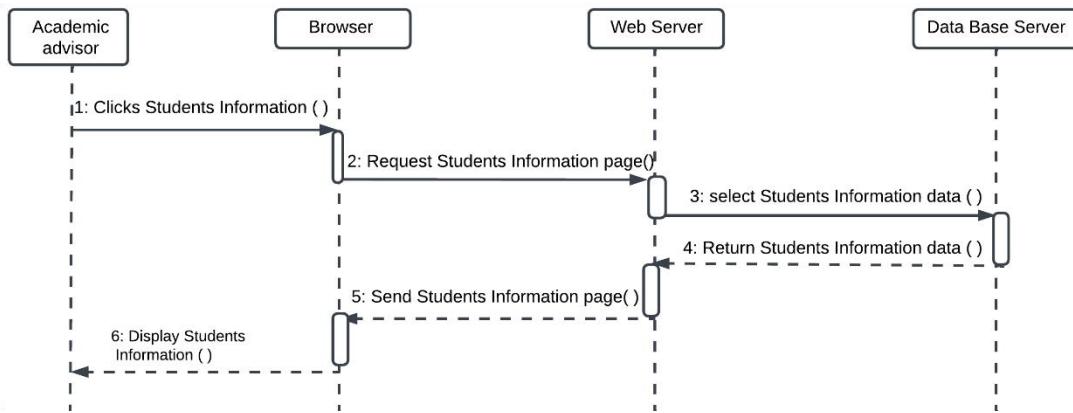


Figure 2.52: View student profiles Sequence diagram

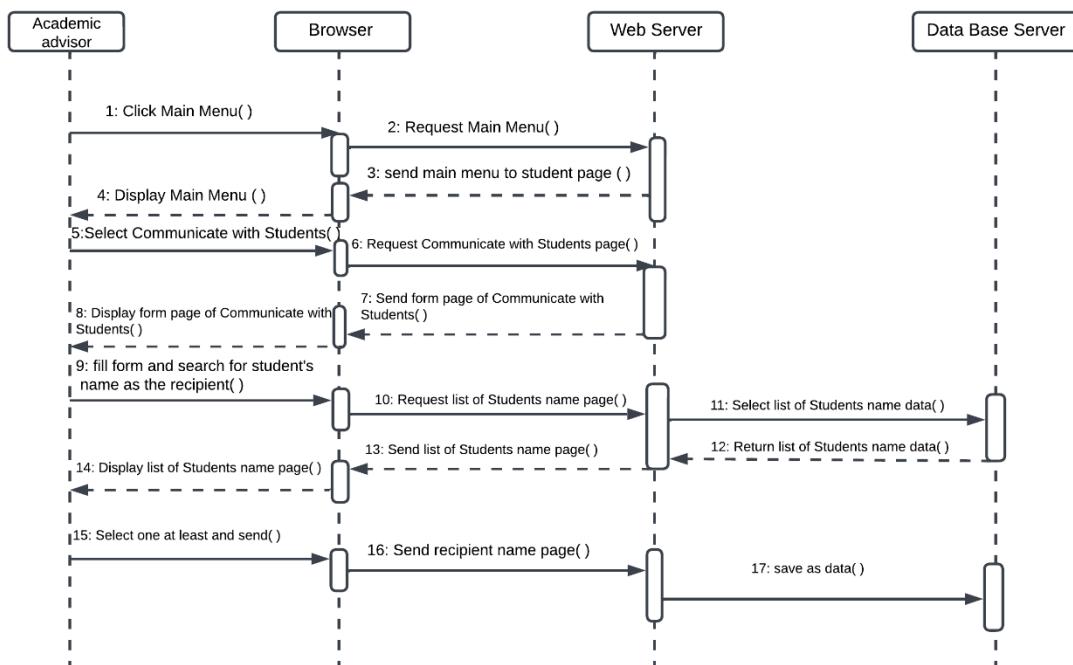


Figure 2.53: Communicate with students Sequence diagram

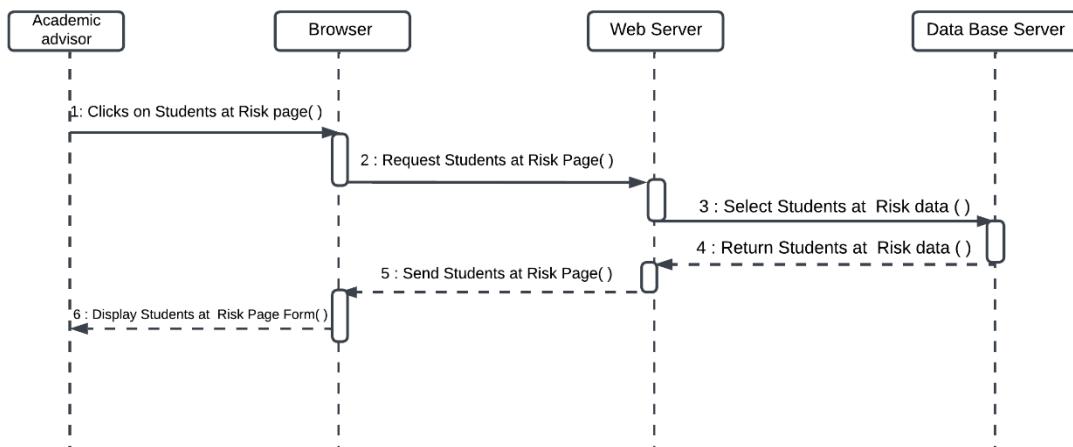


Figure 2.54: Identify students at risk Sequence diagram

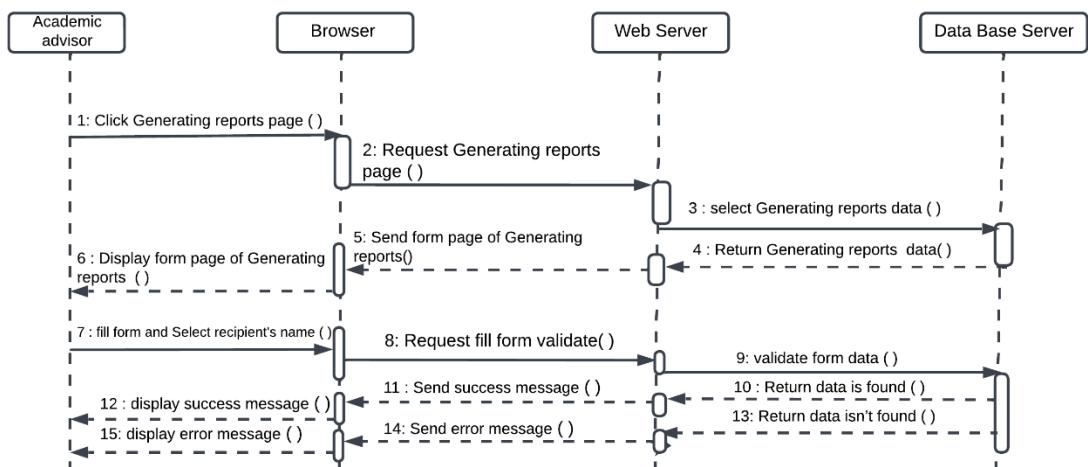


Figure 2.55: Generate reports Sequence diagram

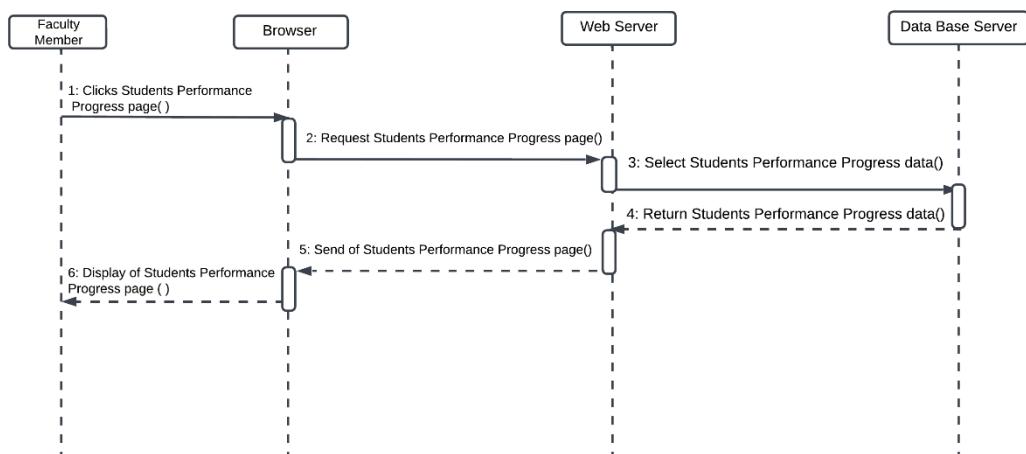


Figure 2.56: Monitor of students Sequence diagram

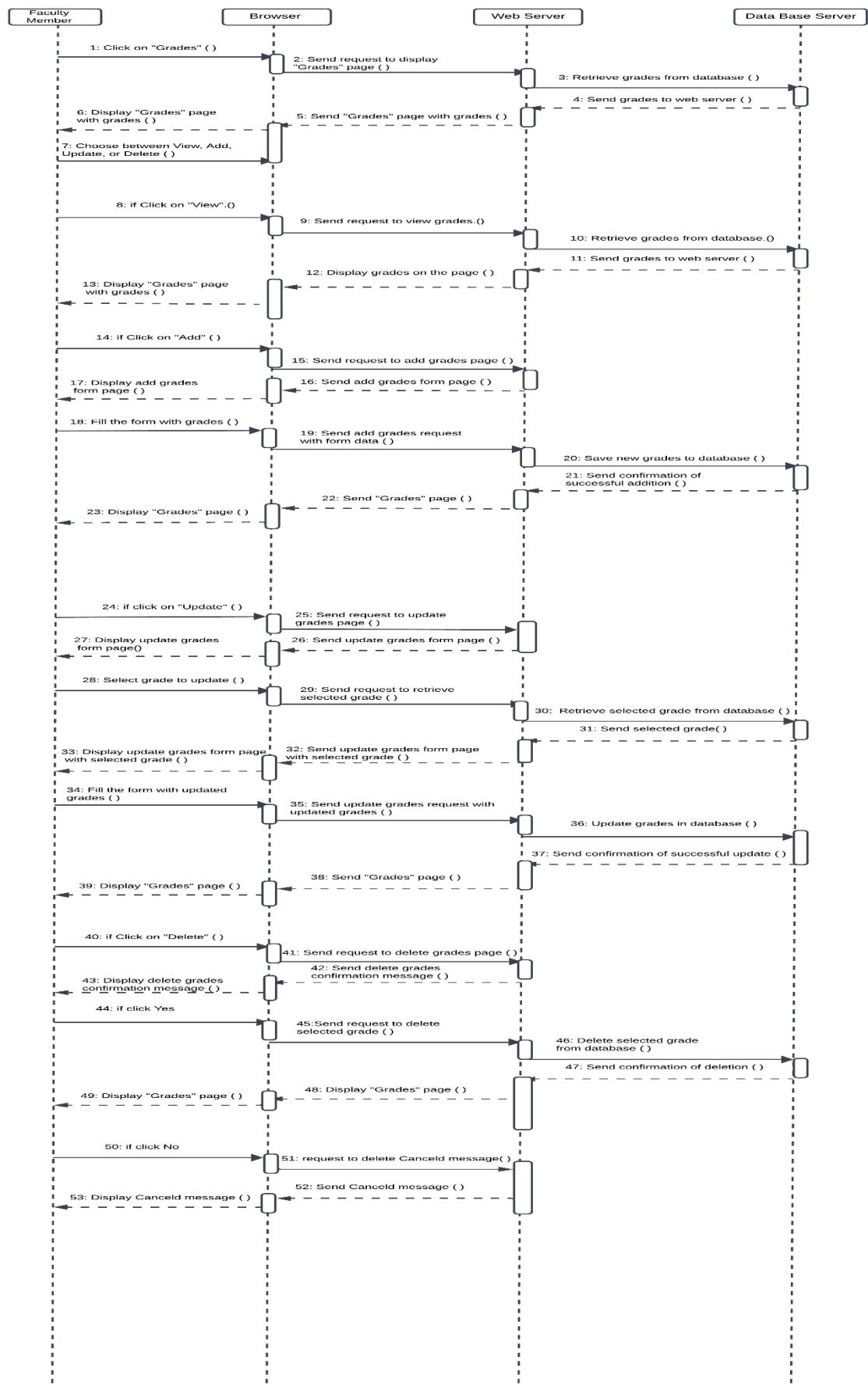


Figure 2.57: Manage grades Sequence diagram

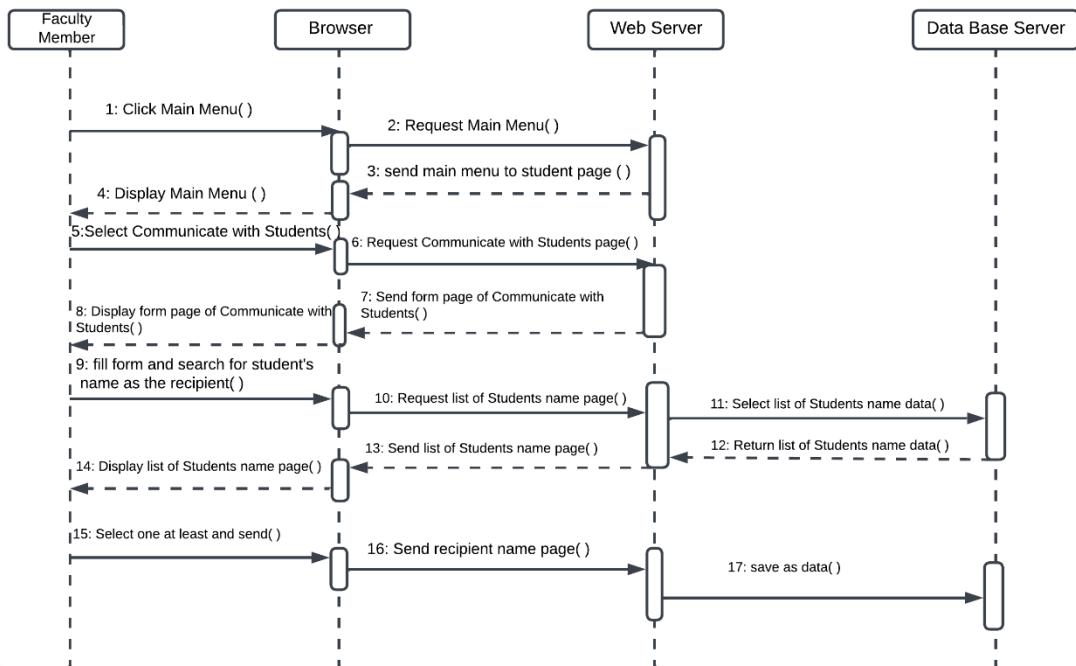


Figure 2.58: Communicate with students Sequence diagram

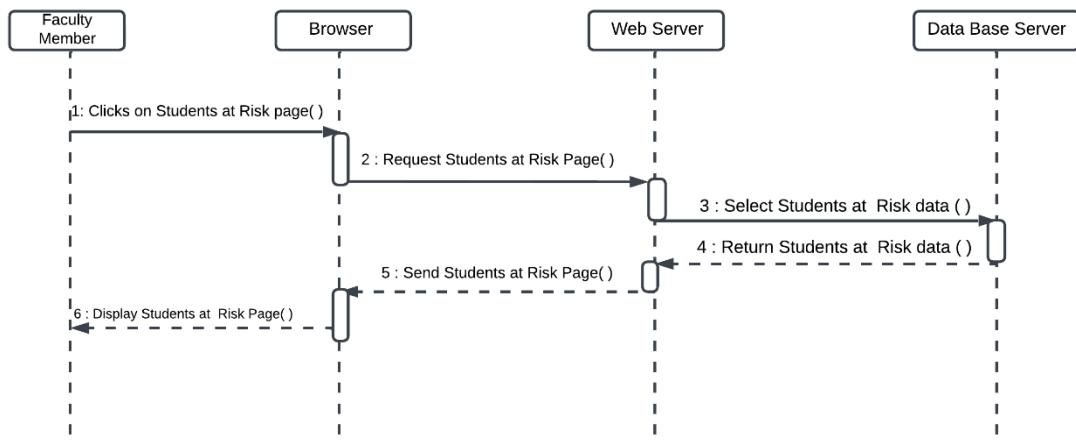


Figure 2.59: Identify students at risk Sequence diagram

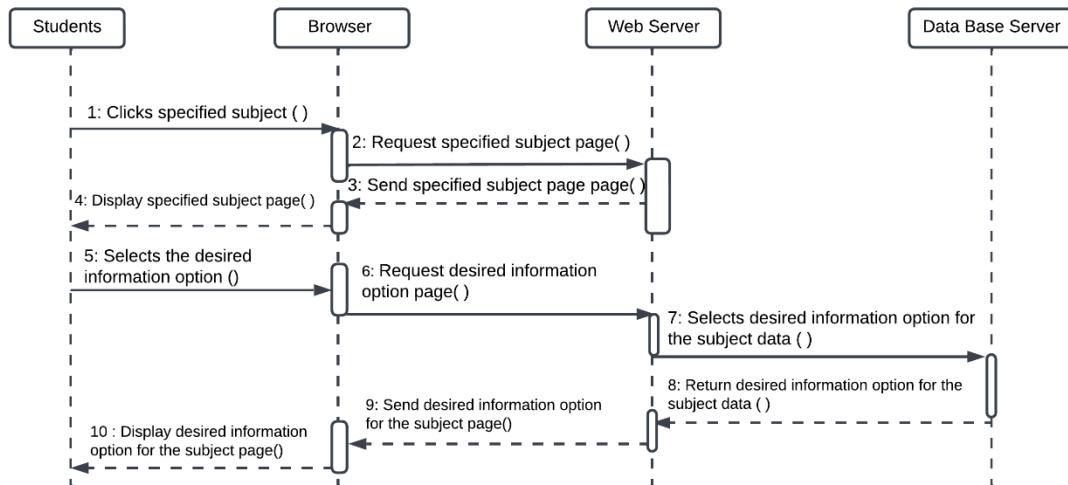


Figure 2.60: view subject information Sequence diagram

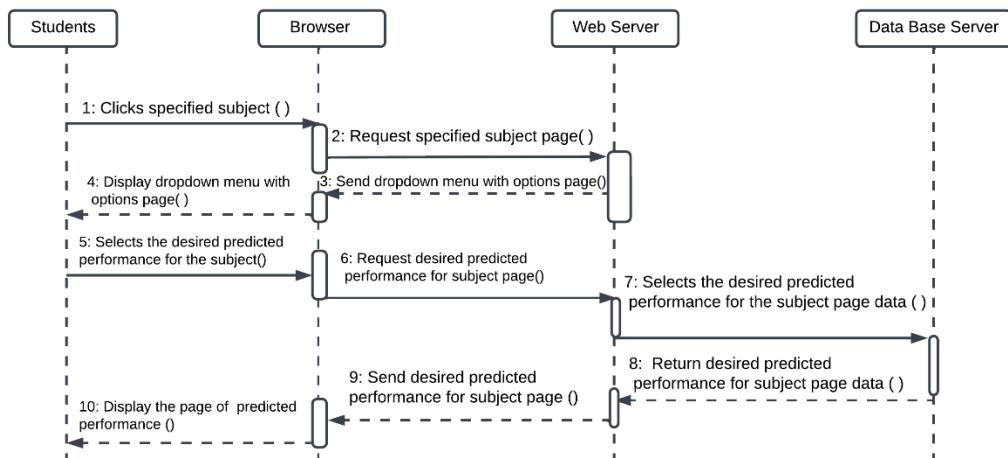


Figure 2.61: view the predicted academic performance Sequence diagram

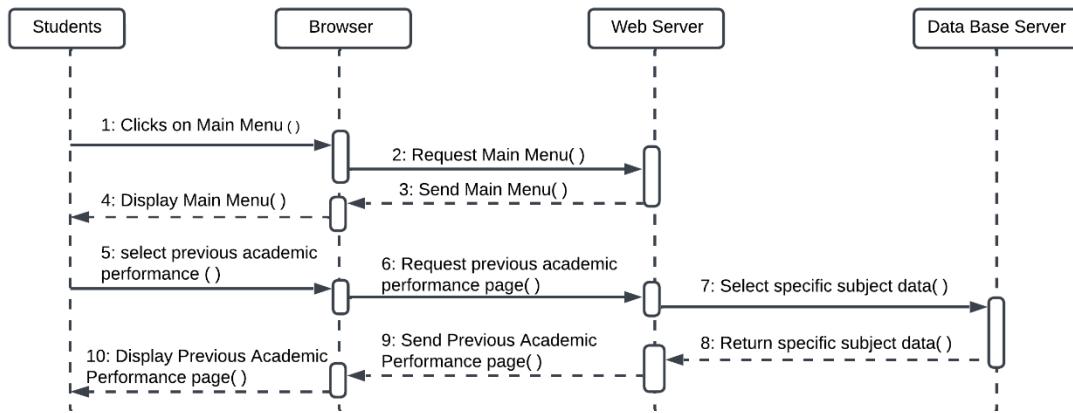


Figure 2.62: View previous academic performance Sequence diagram

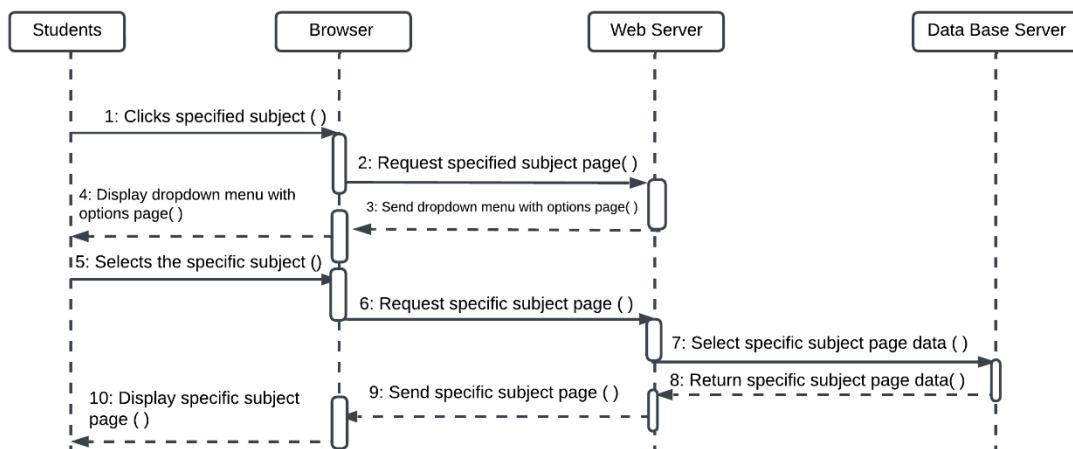


Figure 2.63: Identify subjects requiring improvement Sequence diagram

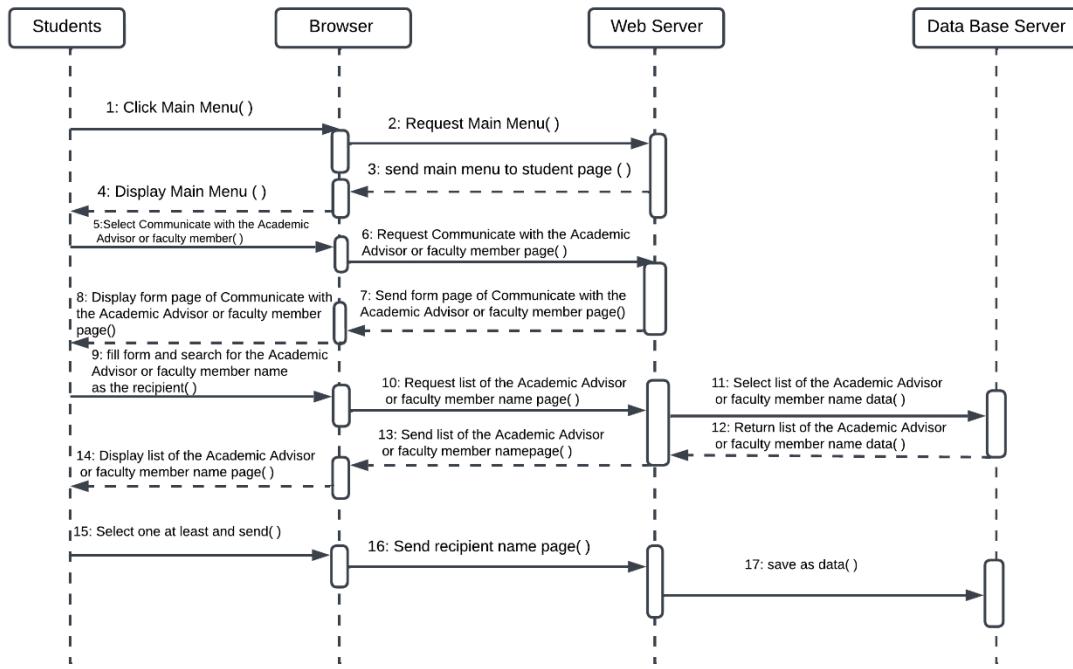


Figure 2.64: Communicate with the Academic Advisor or faculty member Sequence diagram

2.7 Summary

This chapter explored and reviewed related works, presented a collection of functional and non-functional requirements, and conducted an analysis of the model and system by using UML diagrams, such as use case diagrams.

CHAPTER 3

DATABASE DESIGN

3.1 Introduction

This chapter elucidates the system's database design, focusing on the conceptual level. Databases entail a conceptual design, comprising two fundamental types: the Entity Relationship Diagram and the logical Relational Model.

3.2 Entity Relationship Diagram (ERD)

An entity relationship diagram (ERD), also known as an entity-relationship model, is a data model following an object-based data model used at the conceptual level from the three-level architecture of databases, ERD provides a high-level representation of real-world entities and their relationships. the systematic approach to producing a well-organized and scientifically sound model involves the following steps:

1. Identify entities.
2. Determine the relationships between entities and their multiplicities.
3. Verify relationships using a Semantic net.
4. Complete all the previous steps in a unified ERD diagram.

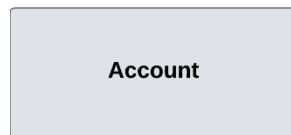
3.2.1 Identify entities

This section represents Identify and determine the entities in the system.



Represents individuals with administrative privileges within the system.

Figure 3.1 : Admin Entity



Represents the basic account information for all users in the system.

Figure 3.2: Account Entity



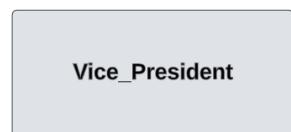
Represents periodic reports and reports at the beginning of each academic year, including assessment colors for grades and feedback.

Figure 3.3 : Report Entity



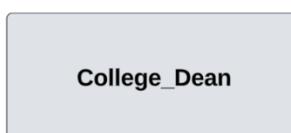
Represents the president of the university actor and their information.

Figure 3.4 : President_of_University Entity



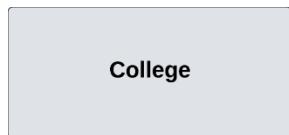
Represents the vice president for academic affairs actor and their information.

Figure 3.5 : Vice_President Entity



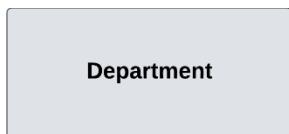
Represents the dean of the college actor and their information.

Figure 3.6 : Dean_of_College Entity



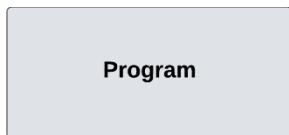
Represents the basic information for College.

Figure3.7 : College Entity



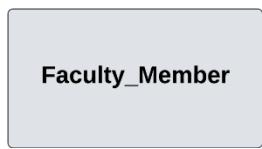
Represents the basic information for Department.

Figure 3.8 : Department Entity



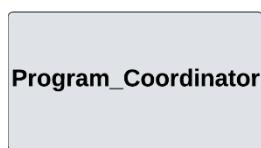
Represents the basic information for Program.

Figure 3.9 : Program Entity



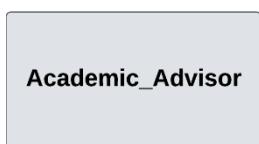
Represents the faculty member actor and their information.

Figure 3.10 : Faculty_Member Entity



Represents the program coordinator actor and their information.

Figure 3.11 : Program_Coordinator Entity



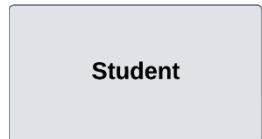
Represents the academic advisor actor and their information.

Figure 3.12 : Academic_Advisor Entity



Represents the subject and its information.

Figure 3.13 : Subject Entity



Represents the student actor and their information.

Figure 3.14 : Student Entity

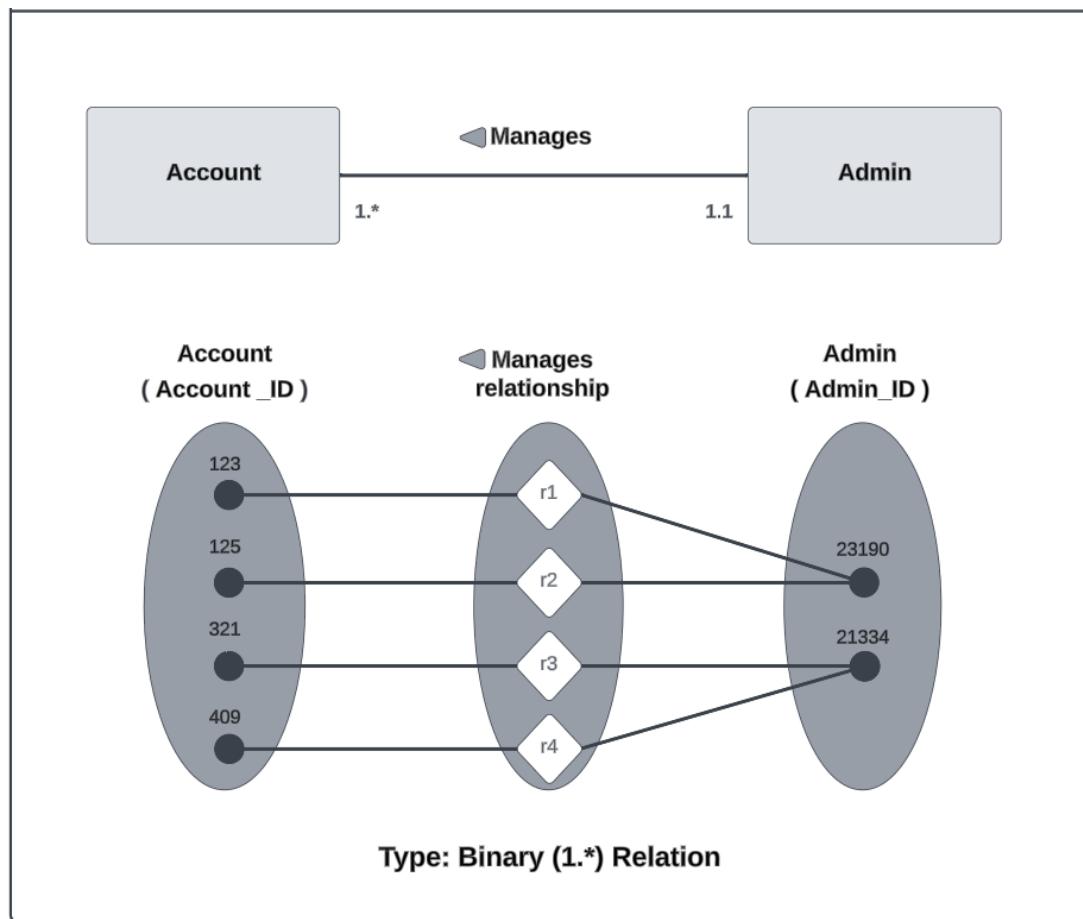


Represents the student coursework assessments for each subject.

Figure 3.15 : Coursework Entity

3.2.1 Determine Entity Relationships, Multiplicities, and Verify

This section represents Determine Entity Relationships, Multiplicities, and Verify relationships using a Semantic net.



3.16: Admin and Account Relationship

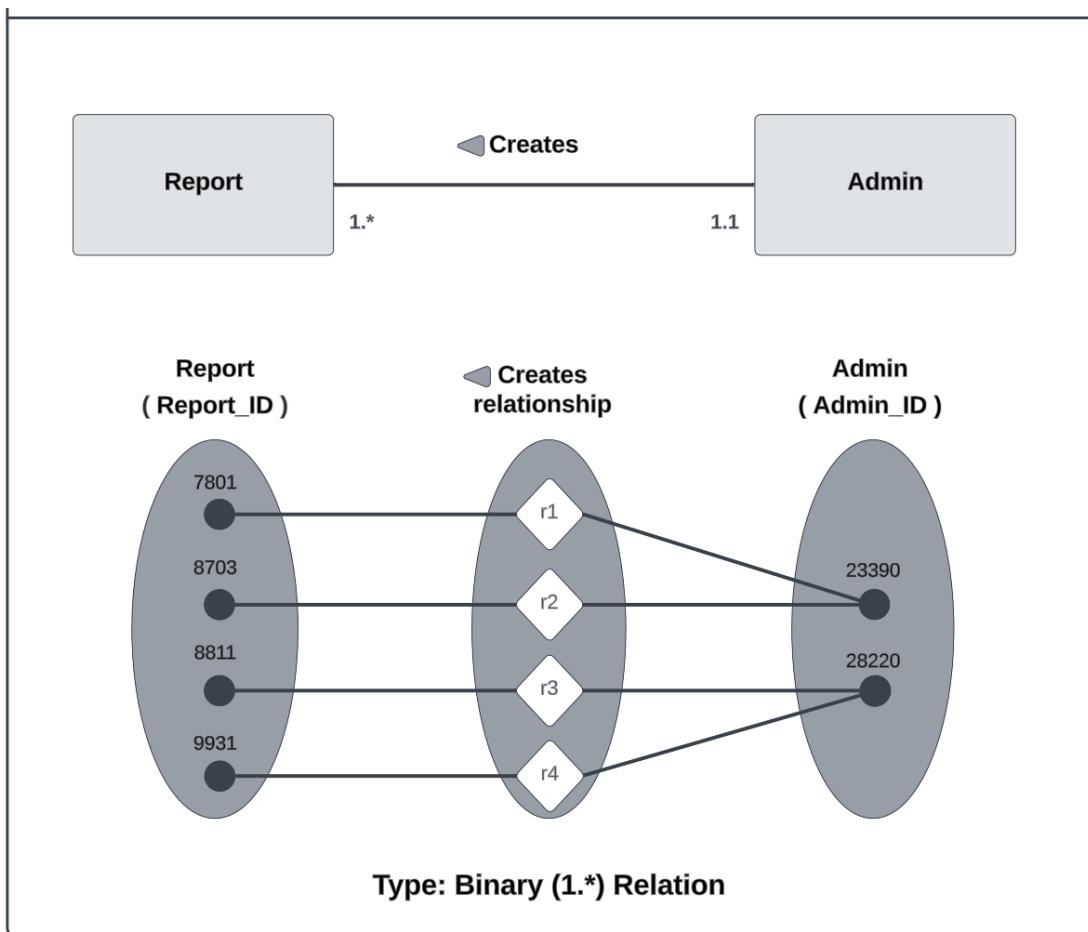


Figure 3.17: Admin and Report Relationship

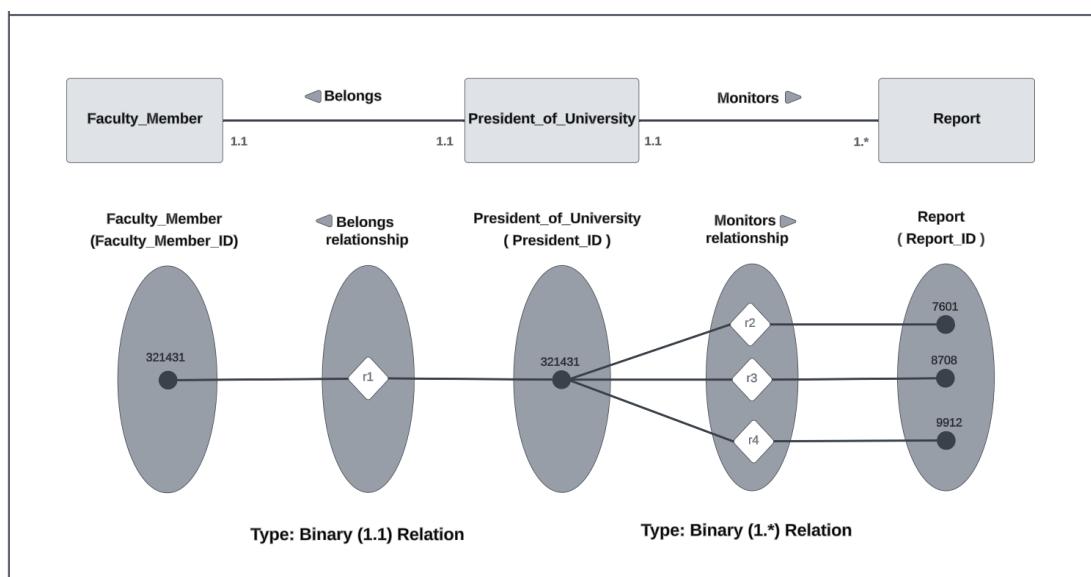


Figure 3.18: President_of_University, Faculty_Member and Report Relationship

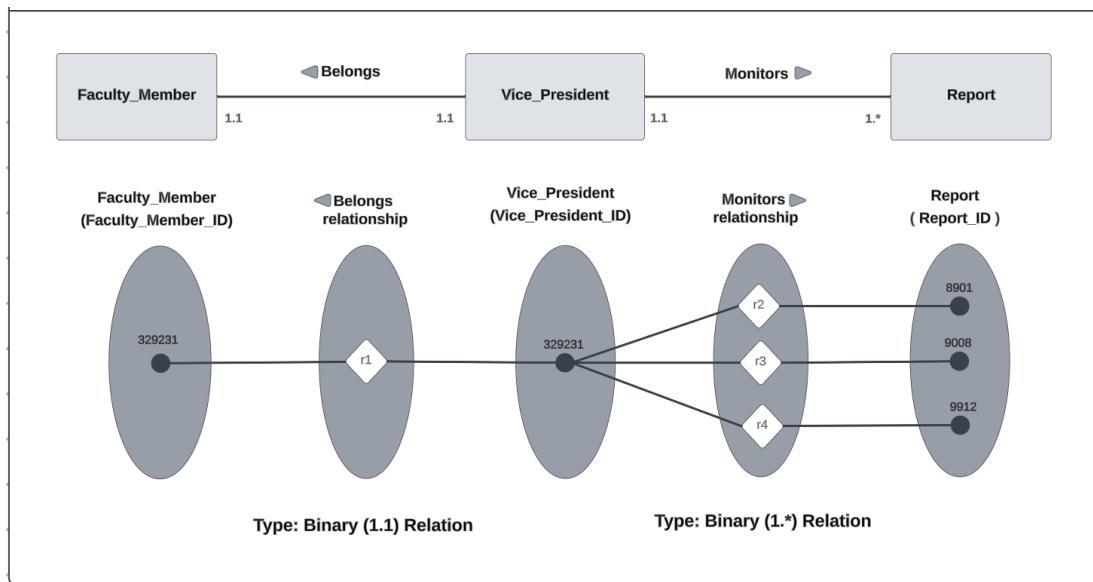


Figure 3.19 : Vice_President, Faculty Member and Report Relationship

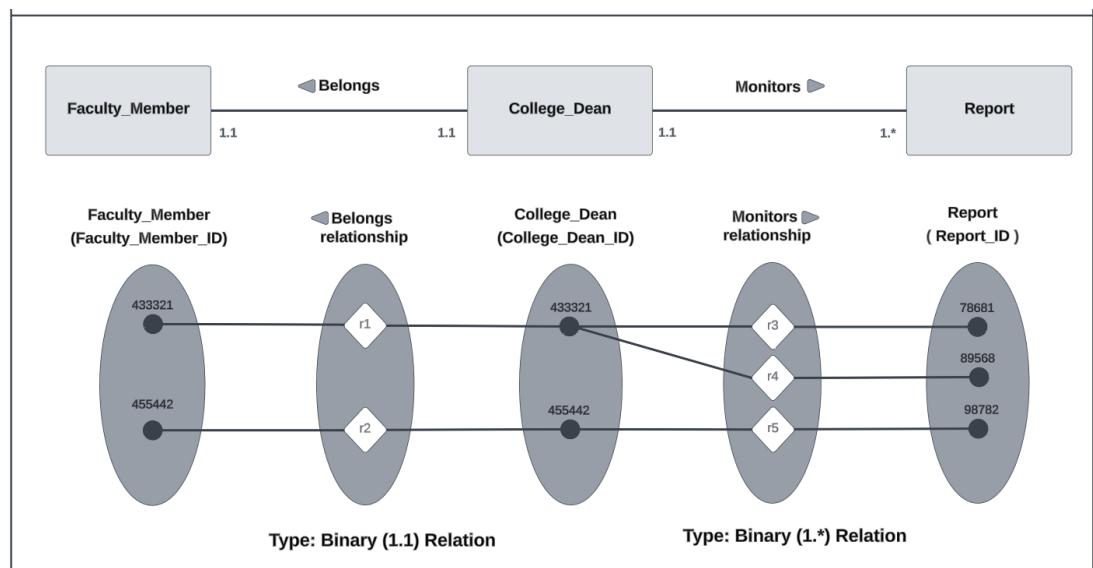


Figure 3.20 : Dean_of_College, Faculty_Member and Report Relationships

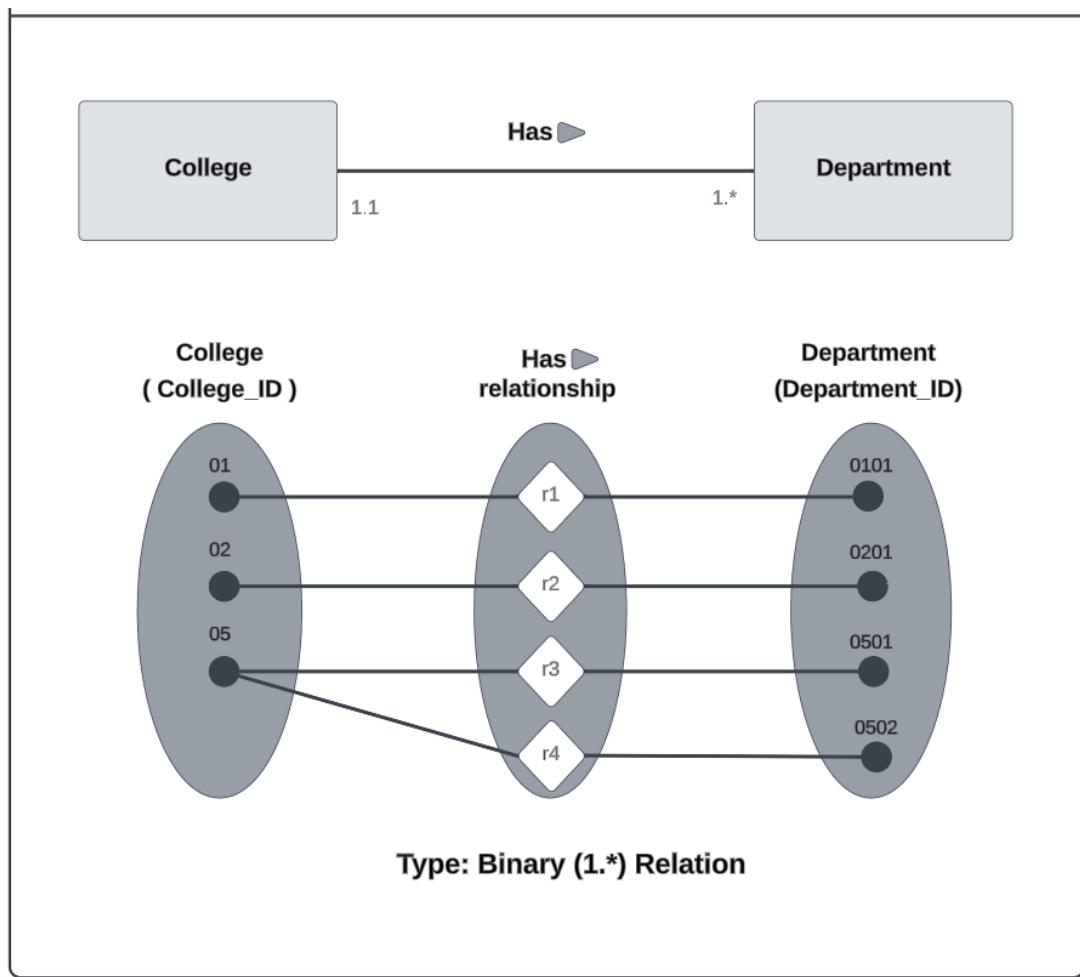


Figure 3.21:College and Department Relationship

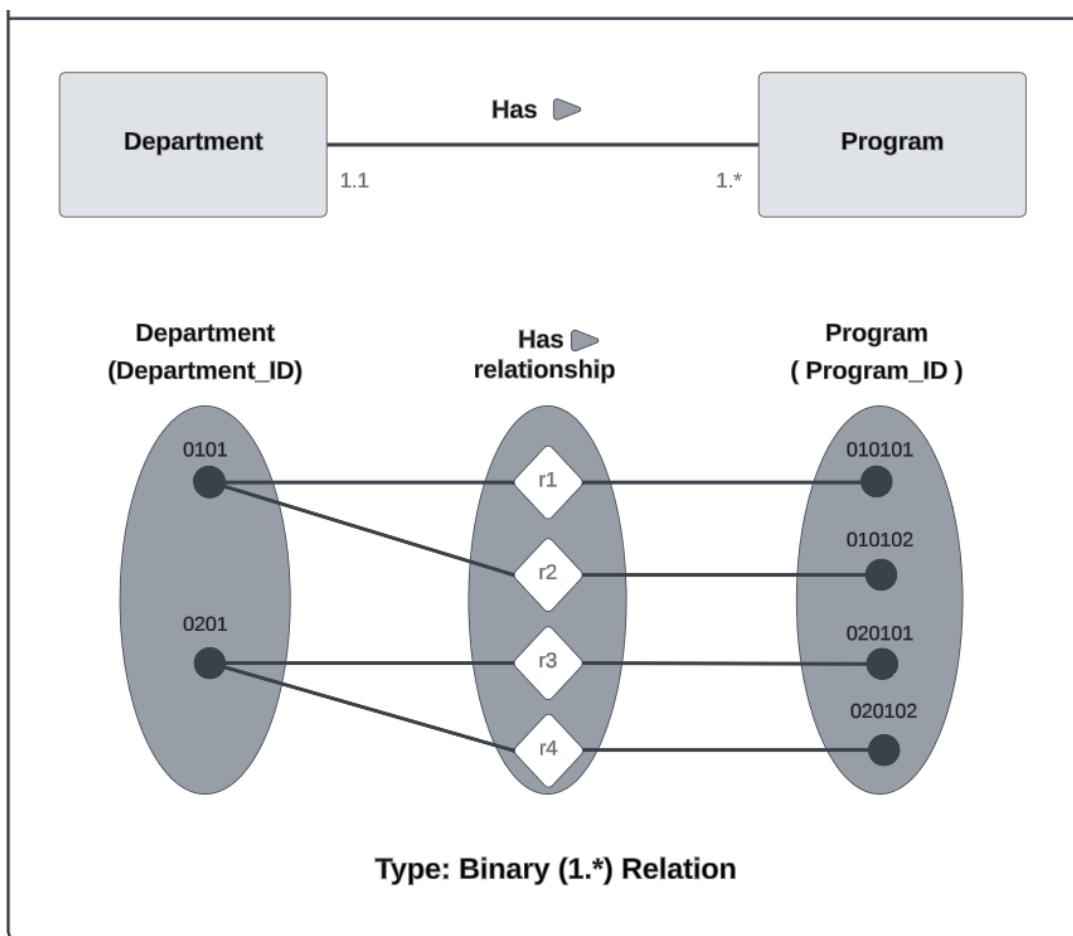


Figure 3.22 : Department and Program Relationship

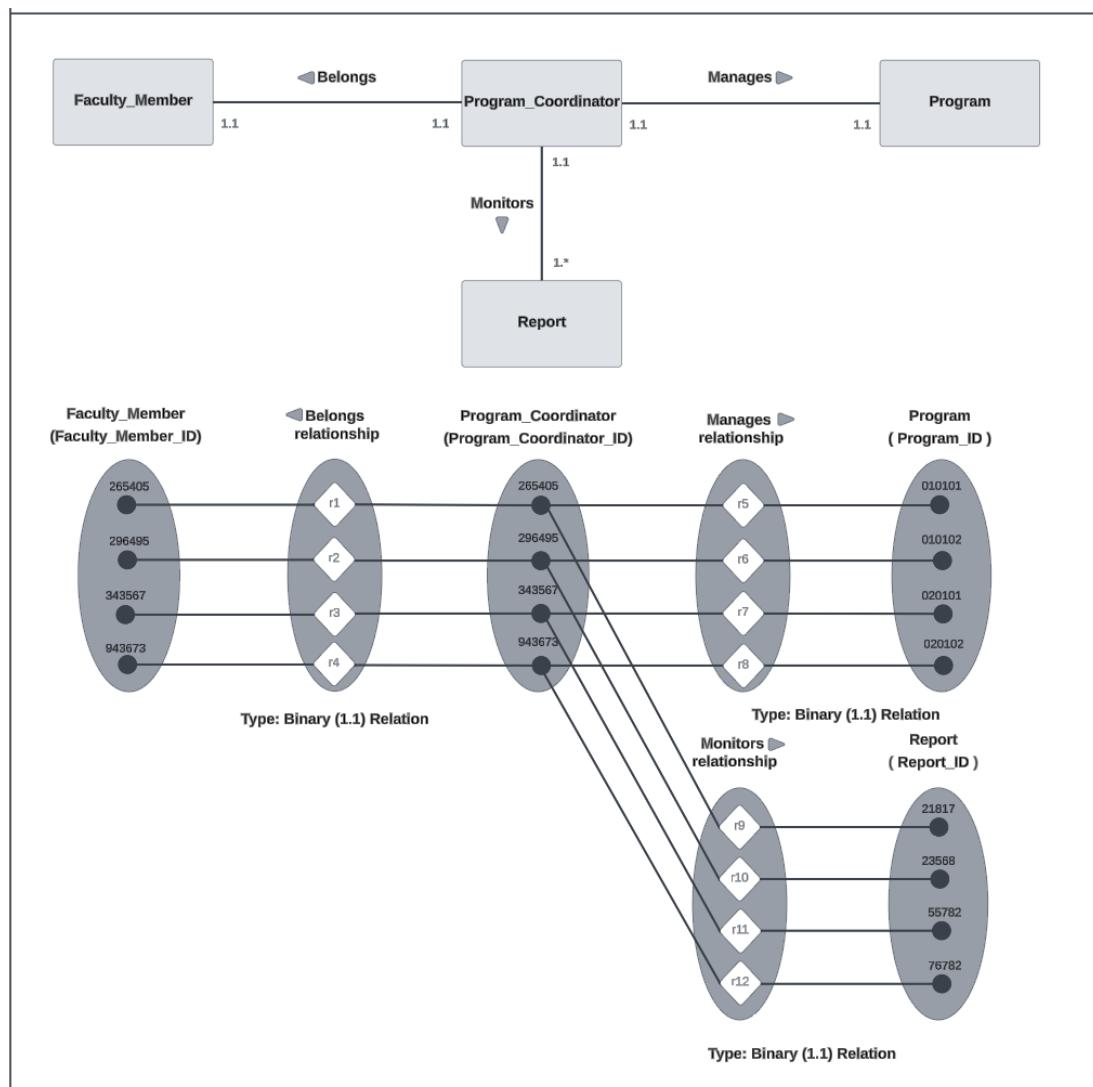


Figure 3.23: Program_Coordinator, Faculty Member, Program and Report Relationship

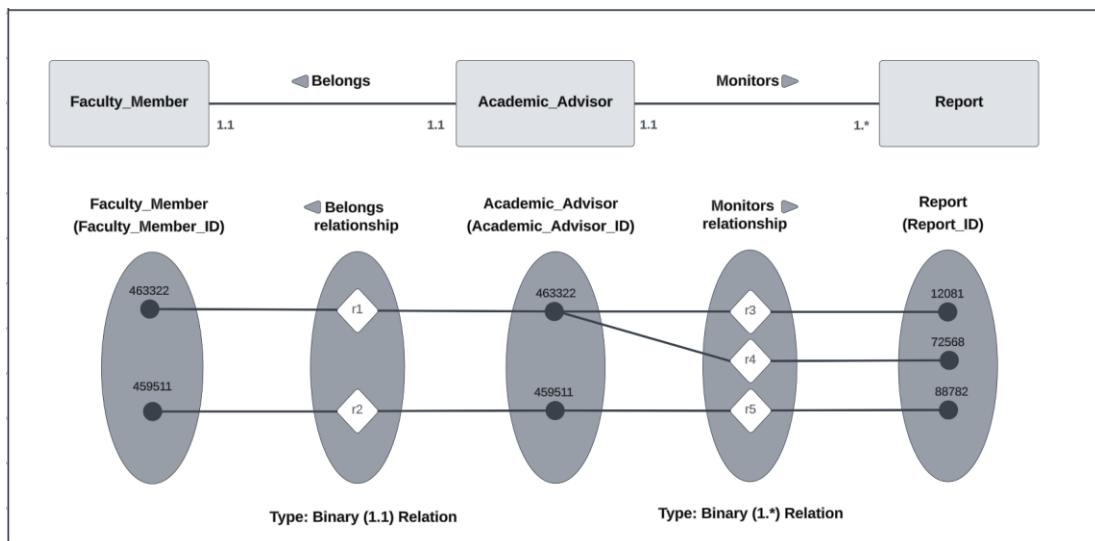


Figure3.24: Academic_Advisor, Faculty Member and Report Relationship

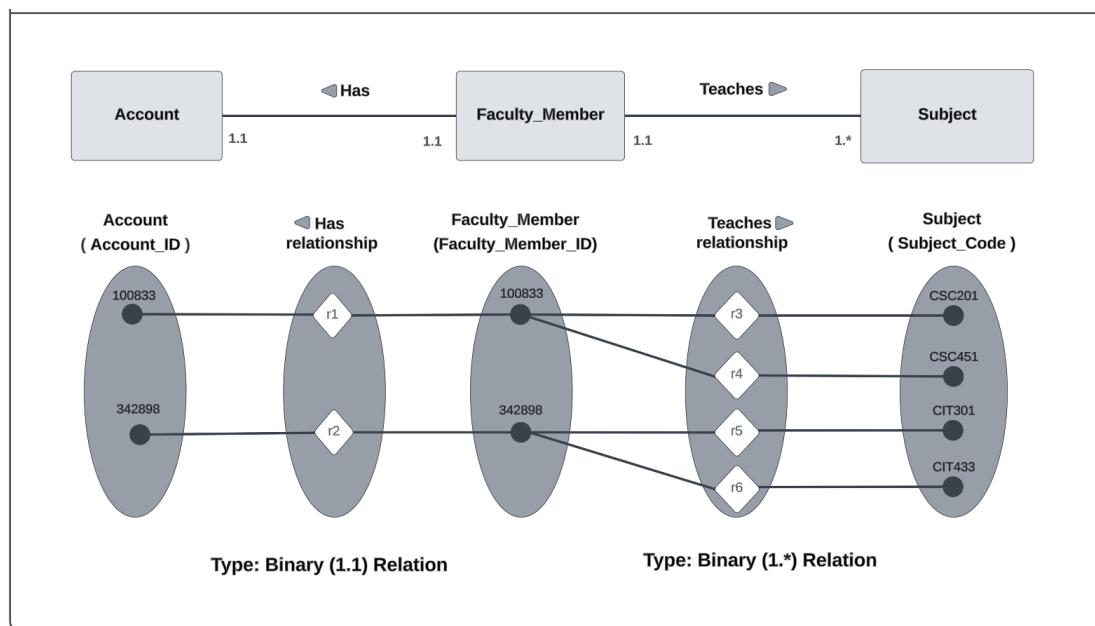


Figure 3.25: Faculty_Member and Account Relationship

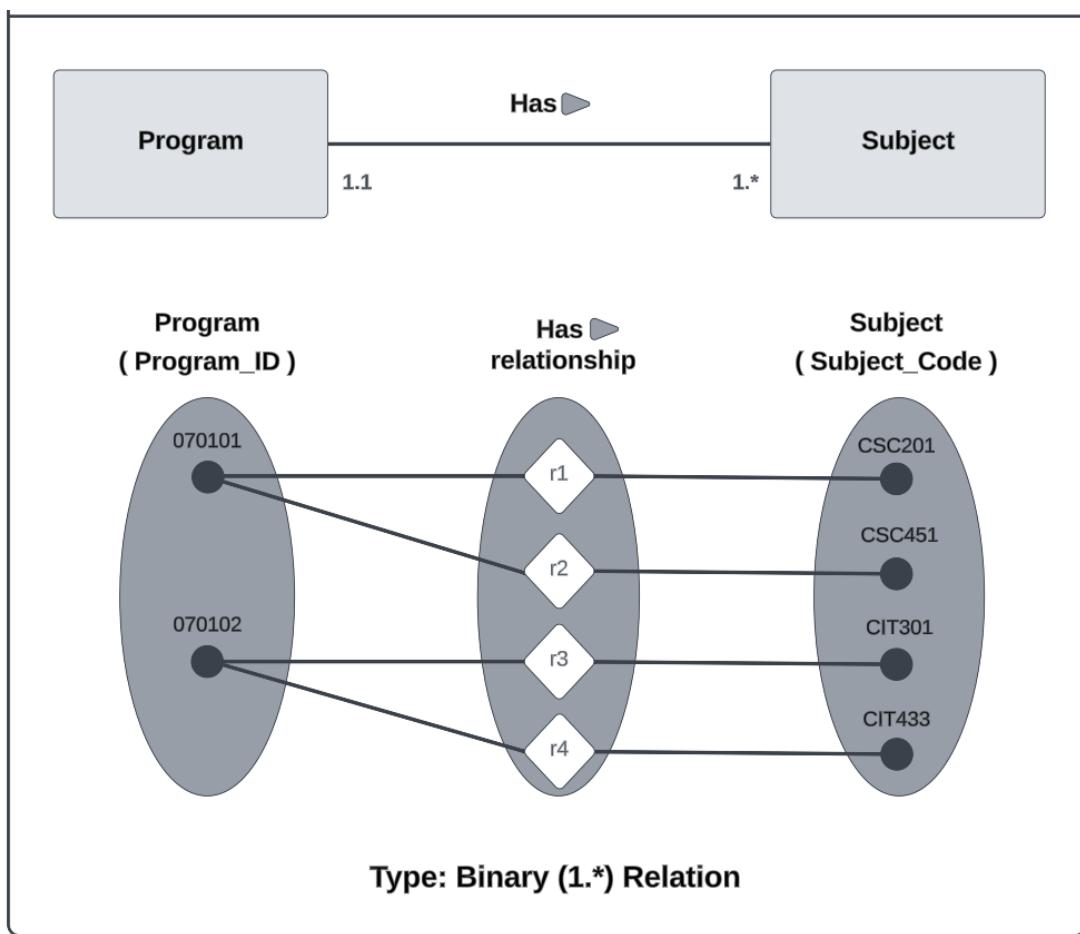


Figure 3.26: Program and Subject Relationship

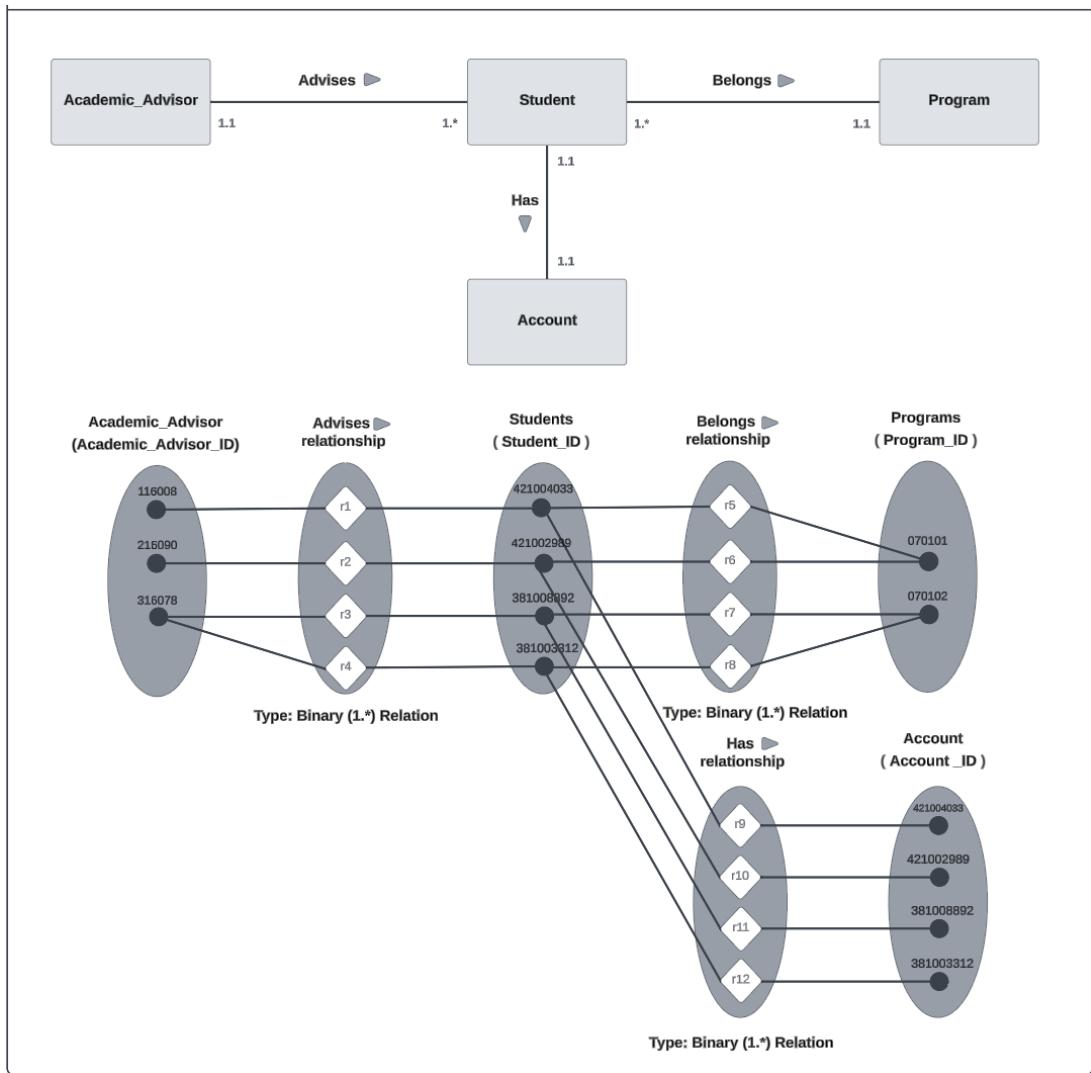


Figure 3.27 : Student, Account, Program and Academic_Advisor relationship

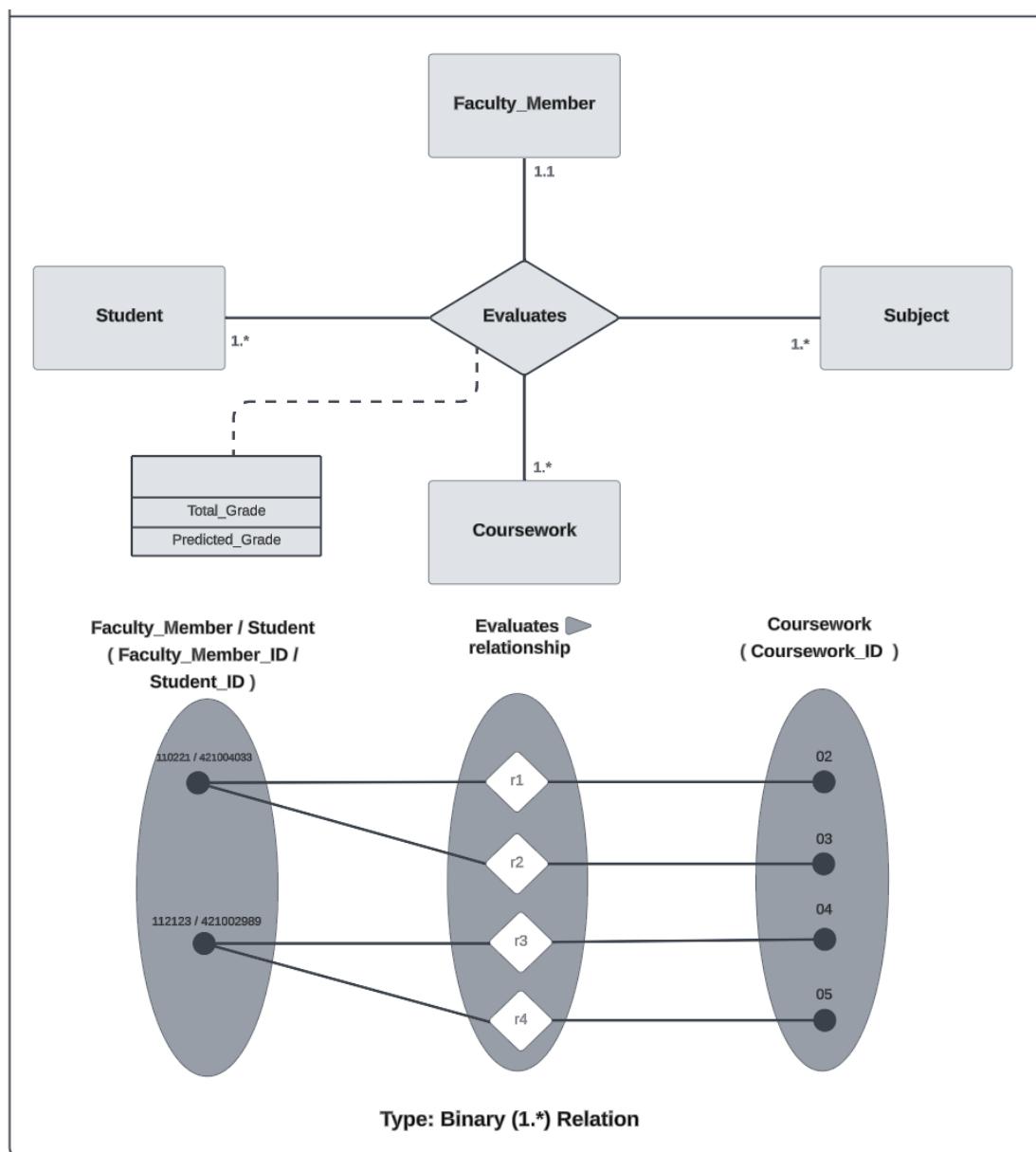


Figure 3.28: Faculty_Member, Coursework , Student and Subject Relationship

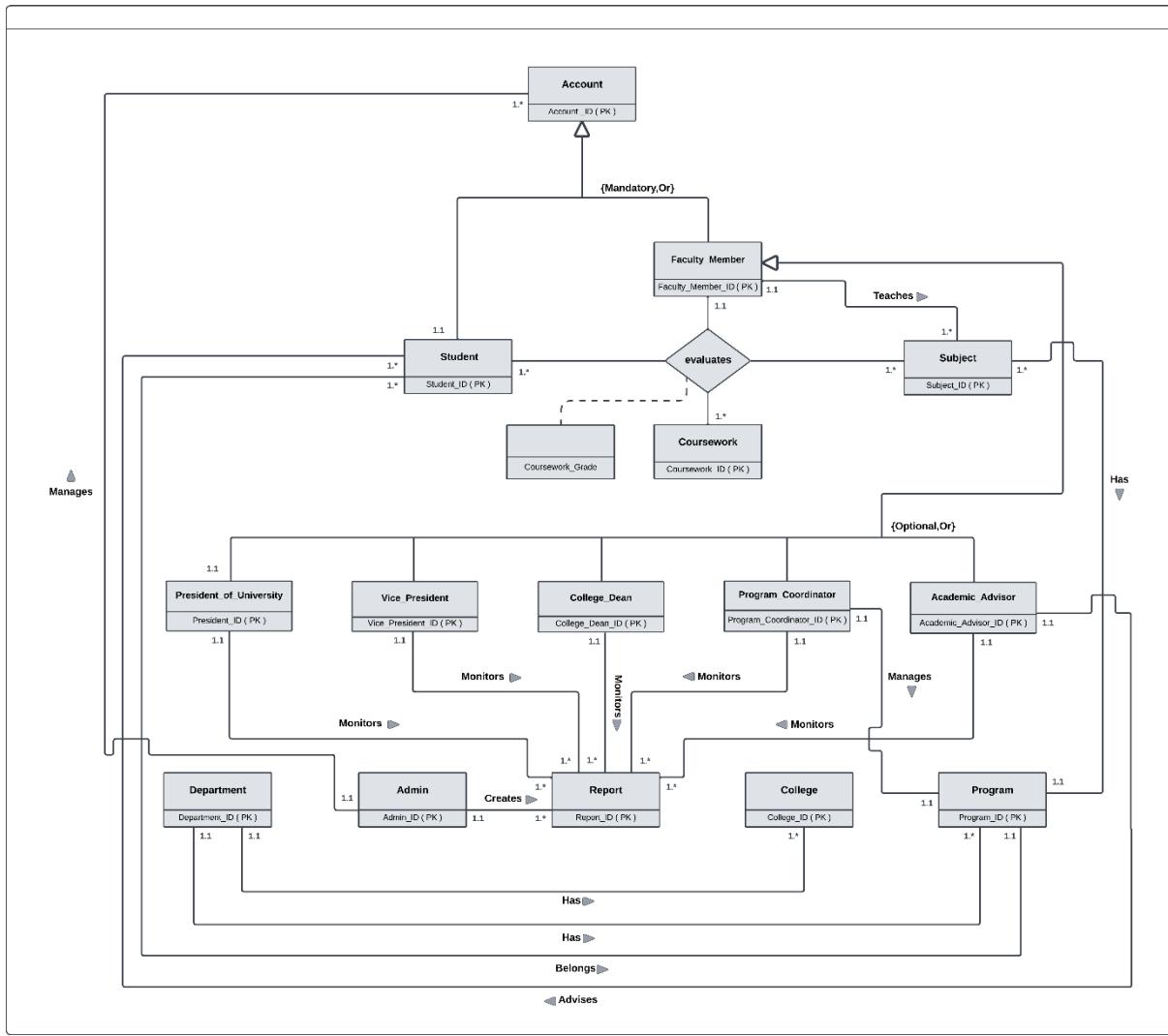


Figure 3.29 : ERD

3.3 logical Relational Model

A relational model is a data model following a record-based data model used at the conceptual level from the three-level architecture of databases, it is a collection of names, definitions, and attributes about data elements that are being used or captured in a database.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Admin_ID	int(2)			No	None		AUTO_INCREMENT
2	From_To	date			No	None		

Figure 3.30 : Admins Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Account_ID	🔑 int(9)		No	None		This ID is the same for each actor in real systems, such as student ID and faculty member ID.	
2	First_Name	varchar(20)	utf8mb4_general_ci	No	None			
3	Last_Name	varchar(20)	utf8mb4_general_ci	No	None			
4	Email	varchar(30)	utf8mb4_general_ci	No	None			
5	Password	varchar(255)	utf8mb4_general_ci	No	None			
6	Mobile	int(10)		No	None			
7	Position	🔑 int(2)		No	None			
8	Status_ID	🔑 int(1)		No	None			

Figure 3.31 : Accounts Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	position_id	🔑 int(2)		No	None			AUTO_INCREMENT
2	position_name	varchar(30)	utf8mb4_general_ci	No	None			

Figure 3.32 : Position Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Status_ID	🔑 int(1)		No	None			
2	Status_Name	varchar(10)	utf8mb4_general_ci	No	None			

Figure 3.33 : Status Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	College_ID	🔑 int(2)		No	None			AUTO_INCREMENT
2	College_Name	varchar(100)	utf8mb4_general_ci	No	None			

Figure 3.34 : Colleges Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Department_ID	🔑 int(4)		No	None			AUTO_INCREMENT
2	Department_Name	varchar(100)	utf8mb4_general_ci	No	None			

Figure 3.35 : Departments Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Program_ID	int(6)			No	None		
2	Program_Name	varchar(50)	utf8mb4_general_ci		No	None		

Figure 3.36 : Programs Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	subject_code	varchar(6)	utf8mb4_general_ci		No	None		
2	Program_ID	int(6)			No	None		
3	subject_name	varchar(70)	utf8mb4_general_ci		No	None		
4	credit_hours	int(1)			No	None		

Figure 3.37: Subjects Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	student_id	int(9)			No	None		
2	subject_code	varchar(6)	utf8mb4_general_ci		No	None		
3	Faculty_member_ID	int(6)			No	None		
4	Semester_Number	int(3)			No	None		

Figure 3.38 : Current semester Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	coursework_id	int(2)			No	None		AUTO_INCREMENT
2	coursework_type_ID	int(2)			No	None		
3	coursework_grade	int(2)			No	None		
4	subject_code	varchar(6)	utf8mb4_general_ci		No	None		

Figure 3.39 : Coursework Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	coursework_type_id	int(2)			No	None		AUTO_INCREMENT
2	coursework_type_name	varchar(20)	utf8mb4_general_ci		No	None		

Figure 3.40 : Coursework type Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	student_ID	int(9)			No	None		
2	coursework_id	int(2)			No	None		
3	coursework_Mark	int(3)			No	None		
4	Subject_Code	varchar(6)	utf8mb4_general_ci		No	None		

Figure 3.41 : Grades Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	student_id	int(9)			No	None		
2	subject_code	varchar(6)	utf8mb4_general_ci		No	None		
3	Semster_Number	int(3)			No	None		
4	grade	int(3)			No	None		

Figure 3.42 : Academic Record Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	student_ID	int(9)			No	None		
2	Semster_Number	int(3)			No	None		
3	GPA	double			No	None		

Figure 3.43 : Student GPA Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Student_ID	int(9)			No	None		
2	Prediction_grade_ID	int(1)			No	None		
3	Semster_Number	int(3)			No	None		

Figure 3.44 : Prediction Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Prediction_grade_ID	int(1)			No		None	
2	Prediction_grade_type	varchar(15)	utf8mb4_general_ci		No		None	

Figure 3.45 : Prediction grade type Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	President_ID	int(6)			No		None	
2	From_To	date			No		None	

Figure 3.46 : University President Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Vice_President_ID	int(6)			No		None	
2	From_To	date			No		None	

Figure 3.47: Vice-President Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	College_Dean_ID	int(6)			No		None	
2	From_To	date			No		None	
3	College_ID	int(2)			No		None	

Figure 3.48: College Dean Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Program_Coordinator_ID	int(6)			No		None	
2	From_To	date			No		None	
3	Program_ID	int(6)			No		None	

Figure 3.49 : Program Coordinator Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Academic_Advisor_ID	🔑 ↴ int(6)			No	None		
2	From_To		date		No	None		

Figure 3.50: Academic Advisor Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Academic_Advisor_ID	🔑 ↴ int(6)			No	None		
2	student_id	🔑 ↴ int(9)			No	None		

Figure 3.51: Academic Advisor for Student Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Faculty_member_ID	🔑 int(6)			No	None		
2	Major	varchar(25)	utf8mb4_general_ci		No	None		
3	Academic_Rank	varchar(25)	utf8mb4_general_ci		No	None		
4	Department_ID	🔑 int(4)			No	None		

Figure 3.52: Faculty Member Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	student_id	🔑 int(9)			No	None		
2	School_type_id	🔑 int(2)			No	None		
3	school_percentage	int(3)			No	None		
4	aptitude_test	int(3)			No	None		
5	academic_achievement	int(3)			No	None		
6	Program_ID	🔑 int(6)			No	None		

Figure 3.53 : Student Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	School_type_id	int(2)			No	None		AUTO_INCREMENT
2	School_type_name	varchar(25)	utf8mb4_general_ci		No	None		

Figure 3.54 : School type Table

3.4 Summary

This chapter elucidated the system's database design, focusing on the conceptual level. Databases entailed a conceptual design, comprising two fundamental types: the Entity Relationship Diagram and the Relational Model.

CHAPTER 4

THE SYSTEM IMPLEMENTATION AND TESTING

4.1 Introduction

This section presents the practical aspects of creating, training, and testing the prediction model. Then, it displays the system screens for each user, along with a description of the purpose of each screen.

4.2 Prediction Model

Two types of prediction models were built: the first at the program level and the second at the course level using Weka.

4.2.1 Program-level prediction

This prediction model aims to determine the expected estimation of students upon graduation from the IT program. The model is constructed using random numbers but in a scientific, logical, unbiased, and unbiased manner. Some manual values have been entered, representing approximately 7% of each column. The classifier was determined by averaging the student's grades, then converting the number into a grade point average, followed by applying a condition to convert the grade into a number. The model consists of four main columns that include university admission data, as follows:

1. School Percentage.
2. Aptitude Test Score.
3. Academic Achievement.
4. Type of School.

Afterwards, students' grades are predicted on two levels based on the programming subjects they have completed.

1. Programming 1.
2. Programming 2.
3. Data Structures and Algorithms.
4. Visual Programming.

A. Second-year students:

- Students who have completed Programming 1 and Programming 2.
- The dataset contains 10,000 rows.
- The predictive model was trained using the Random Forest algorithm with an 80% accuracy rate.
- The correlation coefficient was estimated to be 0.94 after testing.

School_Percentage	Aptitude_Test	Academic_Achievement	Programming1	Programming2	School_type	Color
91	90	95	31	70	2	3
85	96	93	89	77	2	2
94	99	97	73	40	1	2
85	92	95	35	37	1	3
92	93	92	61	79	2	2
87	90	90	51	80	2	2
93	96	86	57	24	1	3
89	91	97	35	51	2	3
92	94	94	97	100	1	2
86	92	98	61	60	1	2
93	94	98	82	47	2	3
86	95	91	92	40	2	2
91	98	95	89	85	1	2
98	88	99	89	57	2	2

Figure 4.1 : Second-year students Dataset

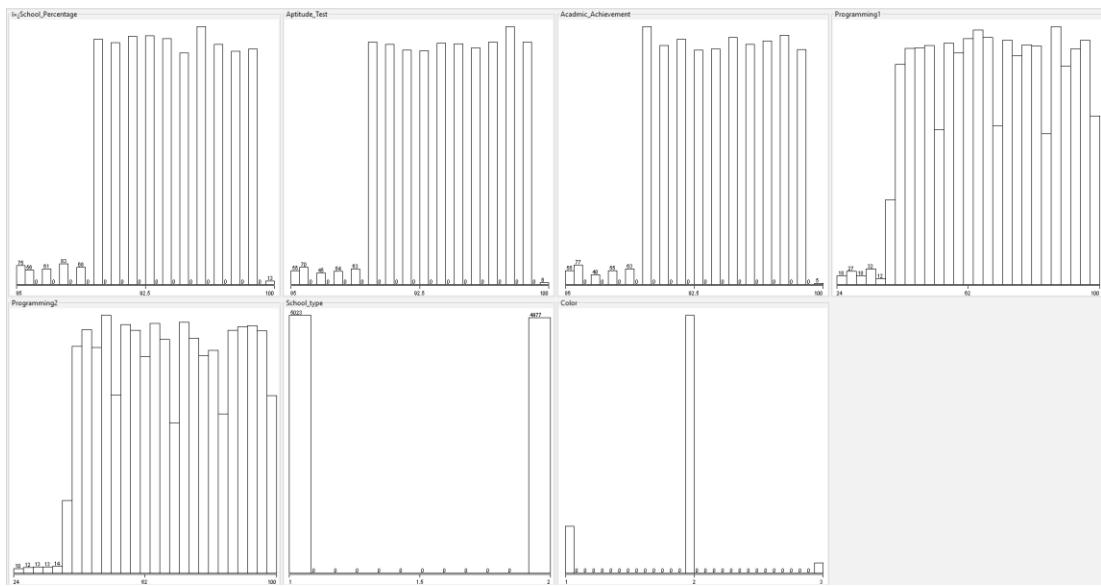


Figure 4.2: S econd-year students Data Visualization

Correlation coefficient	0.9483
Mean absolute error	0.0474
Root mean squared error	0.1326
Relative absolute error	17.9818 %
Root relative squared error	32.1108 %
Total Number of Instances	2000

Figure 4.3 : Second-year students test results

B. Third-year students:

- Students who have completed all programming subjects.
- The dataset contains 5,000 rows.
- The predictive model was trained using the Random Forest algorithm with an 80% accuracy rate.
- The correlation coefficient was estimated to be 0.89 after testing.

School_Percentage	Aptitude_Test	Academic_Achievement	Programming1	Programming2	Data_Structure	Visual_Programming	School_type	Color
91	90	95	31	70	73	59	2	3
85	96	93	89	77	92	59	2	2
94	99	97	73	40	55	87	1	2
85	92	95	35	37	61	71	1	3
92	93	92	61	79	81	60	2	2
87	90	90	51	80	58	86	2	2
93	96	86	57	24	52	92	1	3
89	91	97	35	51	77	59	2	3
92	94	94	97	100	83	60	1	2
86	92	98	61	60	74	92	1	2
93	94	98	82	47	60	42	2	3
86	95	91	92	40	68	82	2	2
91	98	95	89	85	71	56	1	2
98	88	99	89	57	85	88	2	2

Figure 4.4 : Third-year students Dataset

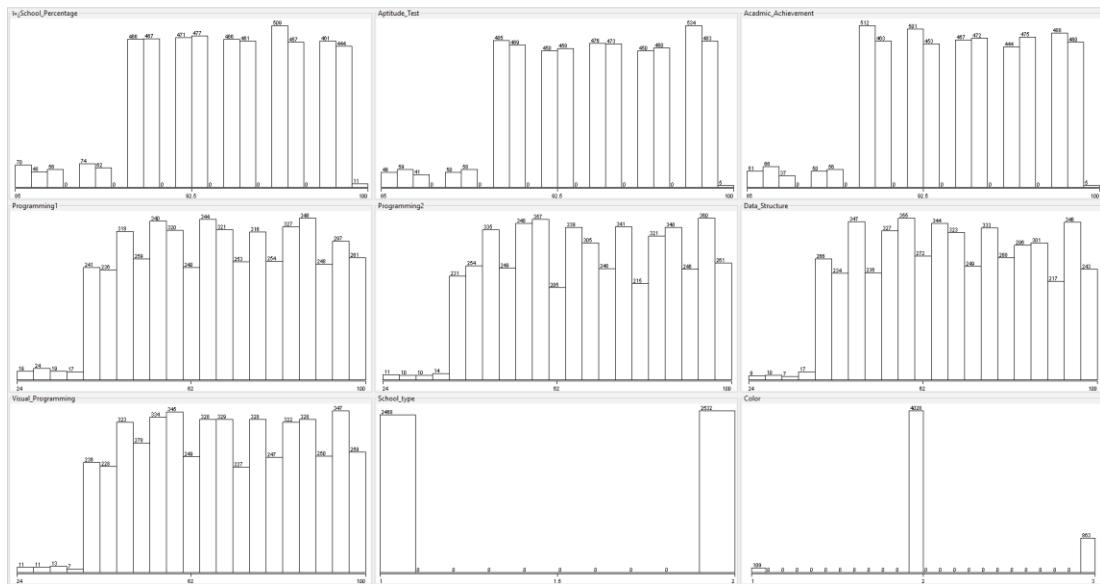


Figure 4.5 : Third-year students Data Visualization

Correlation coefficient	0.8954
Mean absolute error	0.1046
Root mean squared error	0.2019
Relative absolute error	34.4518 %
Root relative squared error	47.0366 %
Total Number of Instances	1000

Figure 4.6 : Third-year students test results

4.2.2 Subject-level prediction

This prediction model aims to determine the grade that a student will achieve when studying the subject 'Database Systems,' based on the coursework they undertake during their study of this subject. The dataset was collected from 200 real records, and to enhance the accuracy and results of the model, 1800 records were added using random numbers, but in a scientific, logical, unbiased, and unbiased manner. Additionally, some manual values were inputted, estimated to be around 7% for each column. The classifier was determined by averaging the student's grades, then applying a condition to convert the grade into a number. The model consists of two levels, and the results vary depending on the timing of completing the coursework for the subject.

1. Quiz 1
2. Midterm Exam 1
3. Lab Quiz
4. Quiz 2
5. Midterm Exam 2
6. Project
7. Quiz 3
8. Lab Exam

A. Prediction in the seventh week:

- The prediction is made by calculating the grades of the first 3 academic activities of the course.
- The predictive model was trained using the M5P algorithm with a 70% accuracy rate.
- The correlation coefficient was estimated to be 0.46 after testing.

The results were not as desired due to the limited number of columns specified for prediction, and there is difficulty in determining the student's behavior at the beginning of the semester.

Quiz1	Mid1	Lab_Quiz	Color
2	6	5	7
3	7	5	6
2	6	5	5
1	5	5	8
2	5	4	9
1	6	0	9
1	5	5	8
2	6	5	8
2	6	5	6
1	4	5	8
2	6	5	7
3	7	5	6
2	4	5	6
3	10	5	1

Figure 4.7 : Prediction in the seventh week Dataset

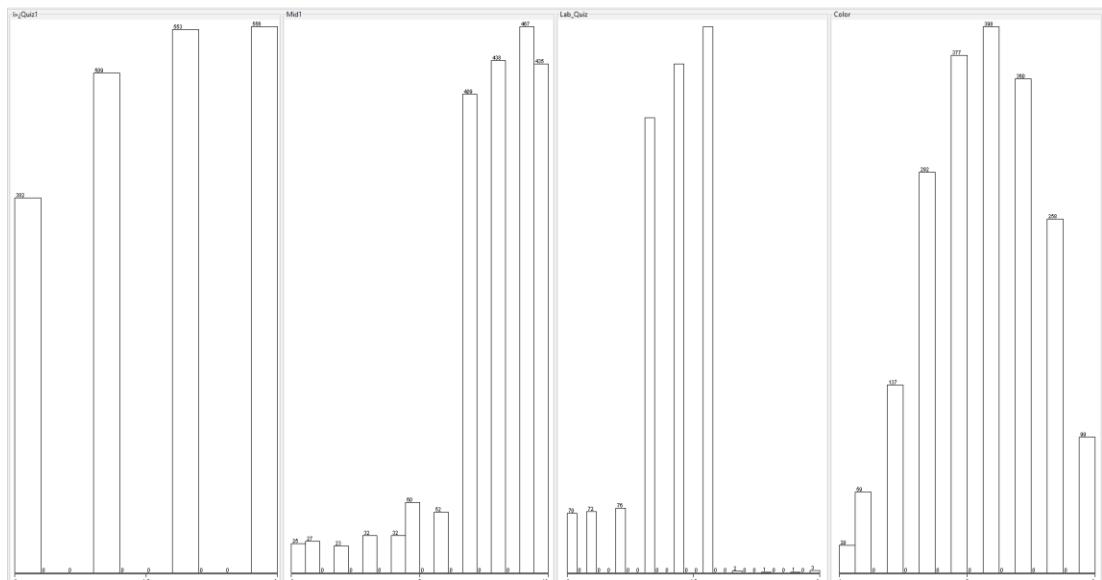


Figure 4.8 : Prediction in the seventh week Data Visualization

Correlation coefficient	0.4677
Mean absolute error	1.307
Root mean squared error	1.5859
Relative absolute error	88.4683 %
Root relative squared error	88.2333 %
Total Number of Instances	500

Figure 4.9 : Prediction in the seventh week test results

B. Prediction in the fourteenth week:

- The prediction is made by calculating the grades of all academic activities of the course.
- The predictive model was trained using the M5P algorithm with a 70% accuracy rate.
- The correlation coefficient was estimated to be 0.68 after testing.

Quiz1	Mid1	Lab_Quiz	Quiz2	Mid2	Project	Quiz3	Lab_Final	Color
2	6	5	2	8	15	1	9	7
3	7	5	0	8	14	1	9	6
2	6	5	1	9	15	2	9	5
1	5	5	0	6	8	0	9	8
2	5	4	0	0	14	0	0	9
1	6	0	2	6	0	0	0	9
1	5	5	3	7	10	1	9	8
2	6	5	2	7	8	2	10	8
2	6	5	2	8	11	2	10	6
1	4	5	2	7	10	1	9	8
2	6	5	1	10	11	1	8	7
3	7	5	3	7	15	4	8	6
2	4	5	1	7	15	4	9	6
3	10	5	3	10	15	4	10	1

Figure 4.10 : Prediction in the fourteenth week Dataset

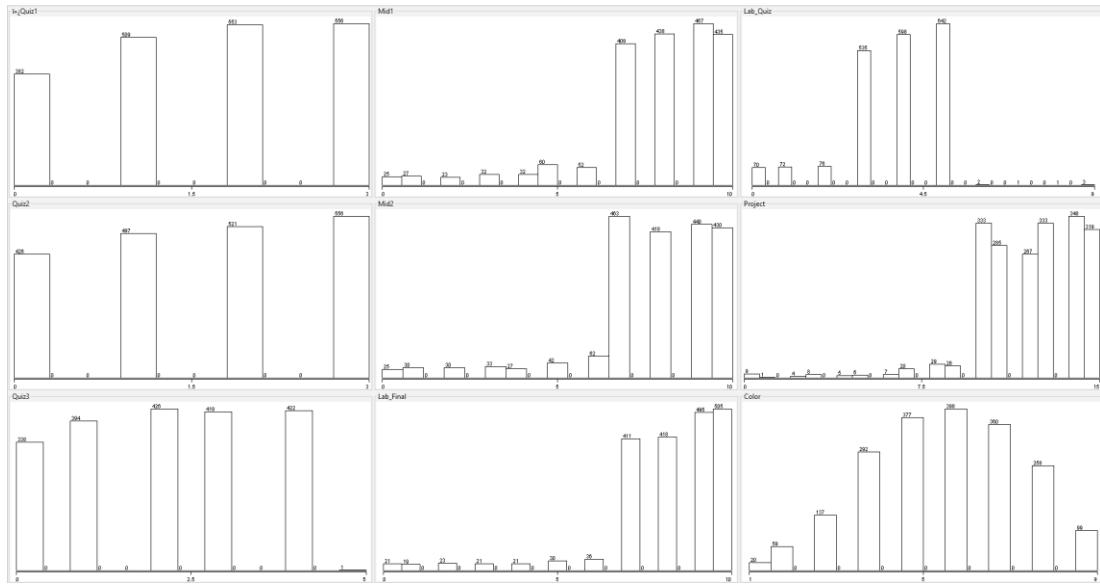


Figure 4.11 : Prediction in the fourteenth week Data Visualization

Correlation coefficient	0.6813
Mean absolute error	1.123
Root mean squared error	1.322
Relative absolute error	75.5318 %
Root relative squared error	73.224 %
Total Number of Instances	600

Figure 4.12 : Prediction in the fourteenth week test results

4.3 System User Interface

This section represents the system user interface for each actor.

4.3.1 General Interfaces

This section represents the user interfaces designed for all users interacting with the system.

1. Login

This page is displayed to all users. It will appear if any user wants to login. The user must enter username and password to login.

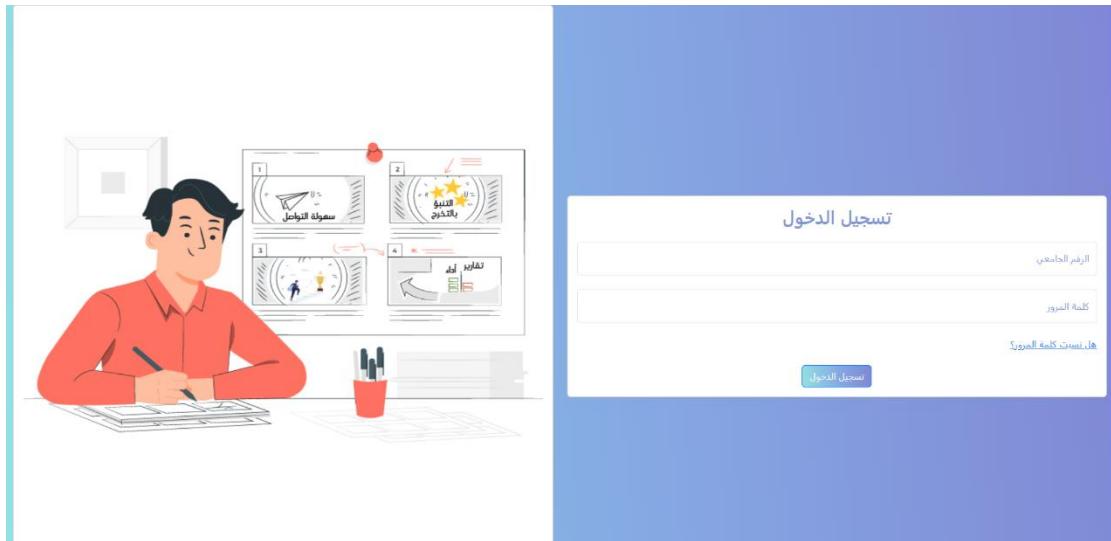


Figure 4.13 : subjects interface

2. Forgot Password

This page is displayed to the user when they select "forgot password" and enter their user ID. Then, a verification code will be sent to the user's email.

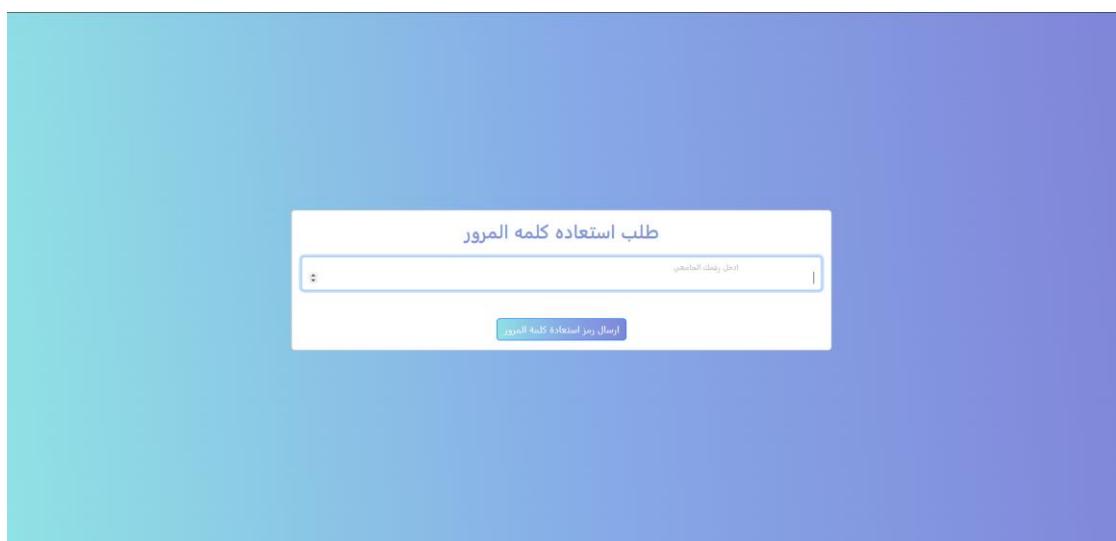


Figure 4.14 : Forgot Password interface

3. Set New Password

This page is displayed to the user for setting a new password by entering the new password and re-entering it a second time.

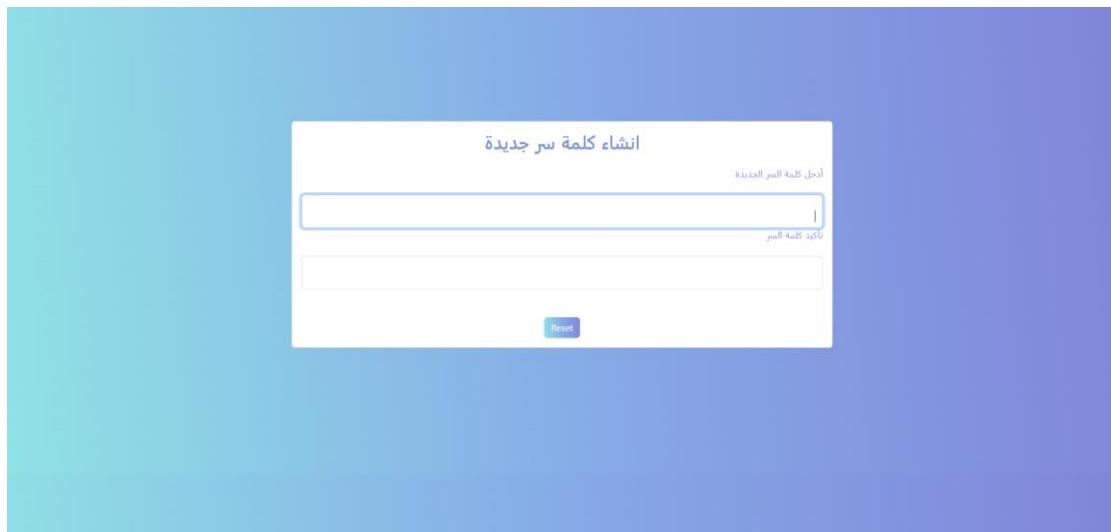


Figure 4.15 : Forgot Password interface

4.3.2 Admin

This section represents the user interfaces designed for Admin interacting with the system.

1. Accounts

This page is displayed to administrators for managing user accounts, allowing them to add, remove, and modify accounts.

نام	حالة	تعديل	النطاق	رقم القوائم	الايميل	الاسم	رقم المعرف
مدرس	مدرس	تعديل	طالب	559705489	mosocyber@gmail.com	مدرس	123123
مدرس	مدرس	تعديل	طالب	556712232	stu.ut.edu.sa@381003212	مدرس	381003212
مدرس	مدرس	تعديل	طالب	556811132	stu.ut.edu.sa@3810032233	مدرس	3810032233
مدرس	مدرس	تعديل	طالب	556122232	stu.ut.edu.sa@421002999	مدرس	421002999
مدرس	مدرس	تعديل	طالب	656144432	stu.ut.edu.sa@421004034	مدرس	421004034

Figure 4.16 : Accounts interface

2. Reports

This page is displayed to administrators, showing reports such as the number of accounts for students and faculty members, etc.

عدد طلاب الجامعة	1	عدد منسقين البرامج	1	عدد المعادلات	1	عدد أعضاء هيئة التدريس	2	عدد المدارس	5
------------------	---	--------------------	---	---------------	---	------------------------	---	-------------	---

Figure 4.17: Reports interface

4.3.3 President of Tabuk University

This section represents the user interfaces designed for President of Tabuk University interacting with the system.

1. Reports

This page is displayed to the President of university, showing reports such as student GPAs, the total number of college students, graduation predictions for students, the total number of students, the number of high-achieving students, and the number of struggling students.

Figure 4.18 : Reports interface

4.3.4 Vice President for Academic Affairs

This section represents the user interfaces designed for Vice President for Academic Affairs interacting with the system.

1. Reports

This page is displayed to the academic affairs student representative, showing reports such as student GPAs, the total number of college students, graduation predictions for students, the total number of students, the number of high-achieving students, and the number of struggling students.



Figure 4.19: Reports interface

4.3.5 Dean of the college

This section represents the user interfaces designed for Dean of the college interacting with the system.

1. Reports

This page is displayed to the dean of the college, showing reports such as student GPAs, the number of students per department, graduation predictions for students, the total number of students, the number of high-achieving students, and the number of struggling students.



Figure 4.20 : Reports interface

4.3.6 Program coordinator

This section represents the user interfaces designed for Program coordinator interacting with the system.

1. Program

This page is displayed to the program coordinator, showing the courses for the program, and allowing them to search, add, remove, and modify courses for the program.

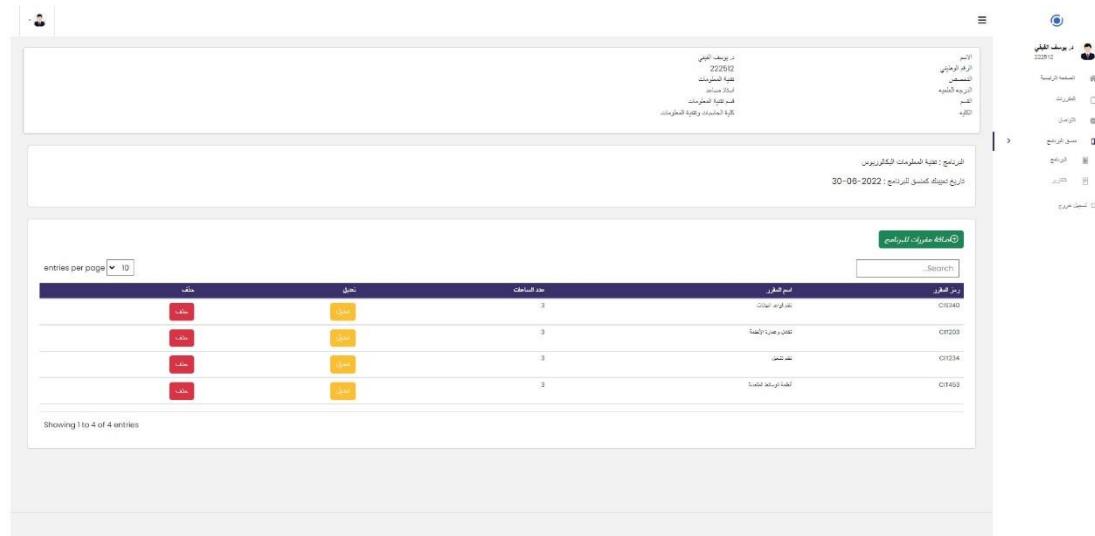


Figure 4.21 : Program interface

2. Reports

This page is displayed to the program coordinator, showing reports for the program such as student GPAs, the number of program students, graduation predictions for students, the number of high-achieving and struggling students, total course hours, and the average number of hours.



Figure4.22 : Reports interface

4.3.7 Academic advisor

This section represents the user interfaces designed for Academic advisor interacting with the system.

1. Students

This page is displayed to the academic advisor showing information about the students under their academic guidance, including current courses, course grades, academic transcripts, and graduation grade predictions.

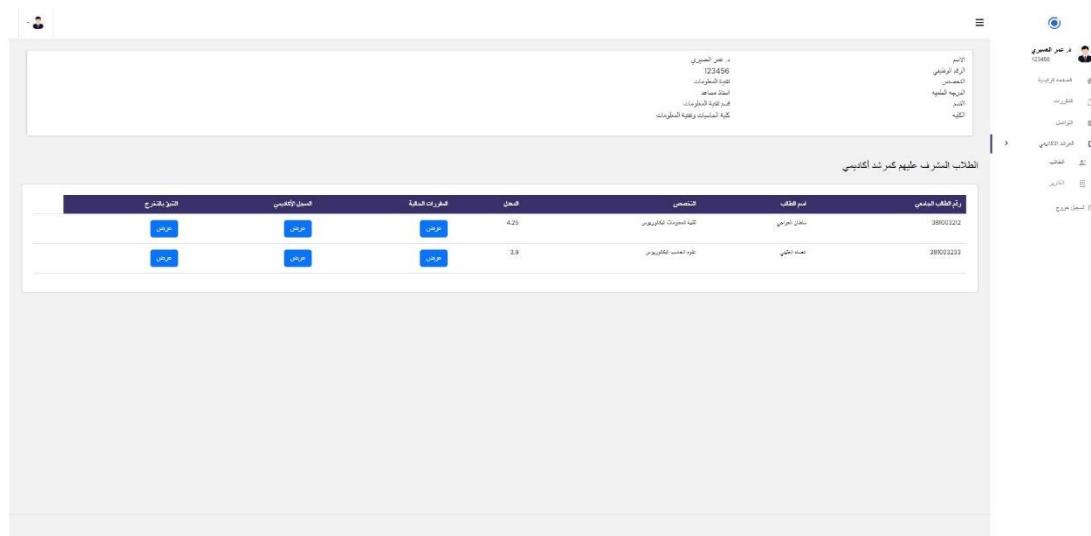


Figure 4.23 : Students interface

2. Reports

This page is displayed to the academic advisor showing reports such as student GPAs, graduation grade predictions, the number of students, and the number of students who excel or struggle academically.

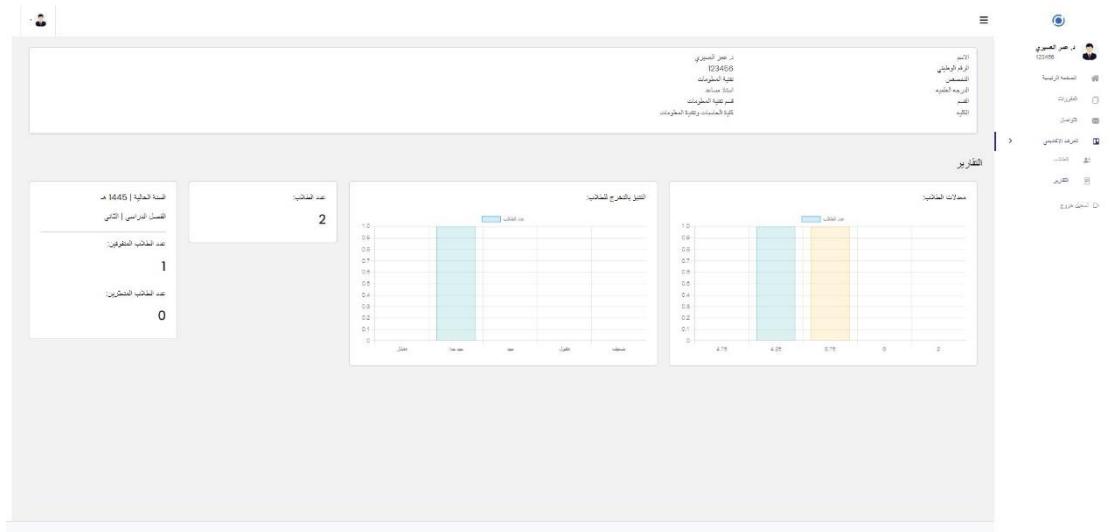


Figure 4.24 : Reports interface

4.3.8 Faculty member

This section represents the user interfaces designed for Faculty member interacting with the system.

1. Subjects

This page is displayed to a faculty member for the subjects they teach, allowing them to add, remove, and modify student activities for each subject they have.

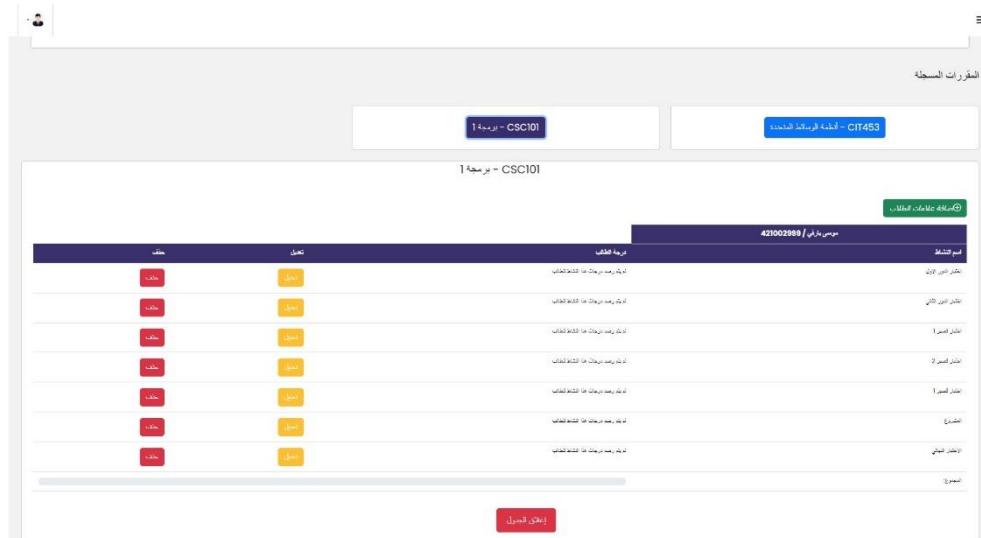


Figure 4.25 : subjects interface

2. Communication

This page is displayed to faculty members for communicating with students enrolled in their subjects.

Figure 4.26 : Communication interface

4.3.9 Student

This section represents the user interfaces designed for students interacting with the system.

1. Registered Subjects

This page appears for a student to display the registered courses for the current semester, show the academic supervisor, and display the communication icon with their supervisor.

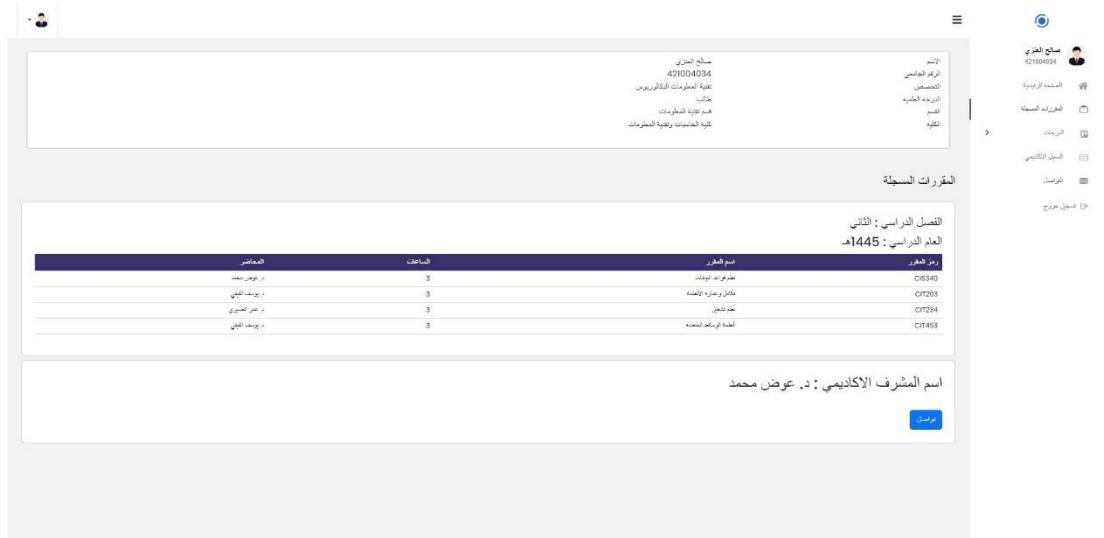


Figure 4.27 : Registered Subjects interface

2. Current subjects

This page displays all current student subjects along with showing the grades of academic activities for each subject and their total.



Figure 4.28 : Current Subjects interface

3. Subject prediction

This page is displayed to a student to predict the subject "Database System" for which coursework grades have been recorded.

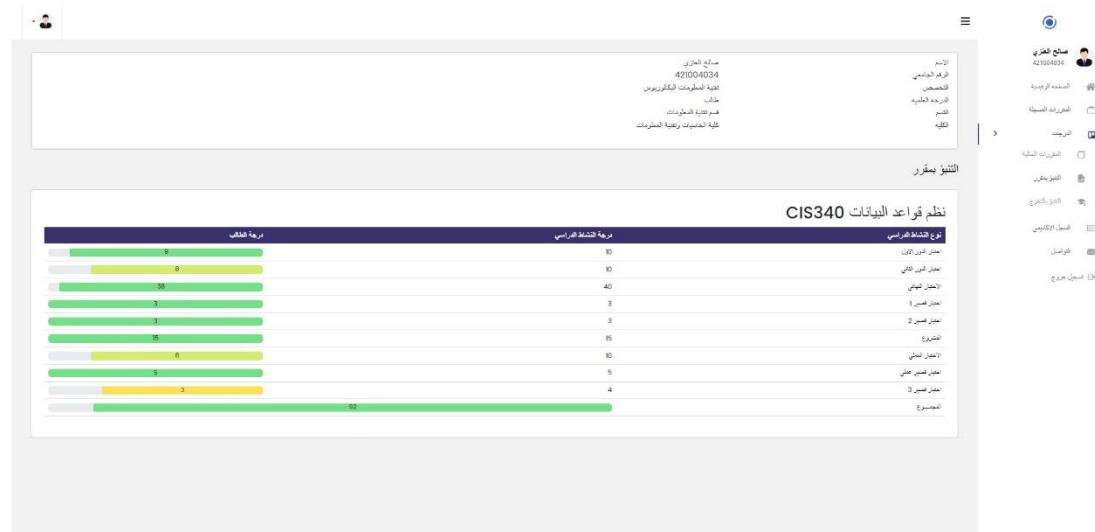


Figure 4.29 : Subject prediction interface

4. Graduation Prediction

This page is displayed to a student to predict the expected grade upon graduation from the university and display the grades of the Subjects they have.

The screenshot shows a user interface for a graduation prediction system. At the top right, there's a sidebar with a user profile (صباح العذري, 421004034) and navigation links: اقسام الدراسات (Academic Majors), المقررات (Courses), التقويم (Calendar), التخرج (Graduation), التخرج بالخارج (Graduation Abroad), and خريجي كلية التربية (Tuition Fees). The main content area displays student information: اسم (Name): صباح العذري, رقم الحاسوب (Computer Number): 421004034, كلية المعلومات (Faculty): كلية المعلومات, and قسم (Degree): بكالوريوس (Bachelor's). Below this, a table shows predicted grades for various subjects: فرقة المرحلة (Grade Level), نتائج الميدل وغوايزن (Midterm and Final Results), درجة (Grade), درجة (Grade), النهاية (Final Grade), المقررات الدراسية (Academic Courses), نوع المدرسة (School Type), مدة (Duration), and نسبة الفرق (Grade Difference). The table includes rows for subjects like نظرية المعلومات (Theory of Information), نظرية الالعاب (Game Theory), and نظرية المعلومات (Information Theory). A note below the table reads: التقدير المتوقع الحصول عليه عند التخرج: (Expected grade upon graduation:). A large green button at the bottom center says ممتاز (Excellent).

Figure 4.30 : Graduation prediction interface

5. Academic Record

This page is displayed to a student to show the academic record including the current and previous semester transcripts, grades, GPA, and credits for all courses they have taken. It also displays the cumulative and semester GPA.

الإسم: 421004034
الرقم الجامعي:
الجنس: ذكور
الدرجة العلمية: البكالوريوس
نوع الملف: ملف طلاب
كلية: كلية الحاسوب وتقنية المعلومات

السجل الأكاديمي

العام الدراسي: 1445 هـ
الصل الدراسي: الثاني

العنوان	النوع	النقطة	النحو	النحو	نوع المقرر
غير متوفر	غير متوفر	3	غير مرددة	غير مرددة	مقرر غير مرددة
غير متوفر	غير متوفر	3	غير مرددة	غير مرددة	مقرر مرددة باللغة
غير متوفر	غير متوفر	3	غير مرددة	غير مرددة	مقرر تمارين
غير متوفر	غير متوفر	3	غير مرددة	غير مرددة	المادة الدراسات الحاسوب

المعدل الفصلي: غير متوفر
المعدل المركب: غير متوفر

العام الدراسي: 1445 هـ
الصل الدراسي: الأول

Figure 4.31 : Academic record interface

6. Communication

page is displayed to a student for communication with a faculty member or academic advisor.

الإسم: 421004034
الرقم الجامعي:
الجنس: ذكور
الدرجة العلمية: البكالوريوس
نوع الملف: ملف طلاب
كلية: كلية الحاسوب وتقنية المعلومات

المحاجة

رسالة

العنوان	النحو	نوع المقرر
د. حمزة محمد	غير مرددة	مقرر غير مرددة
د. سيف العبد	غير مرددة	مقرر مرددة باللغة
د. سيف العبد	غير مرددة	مقرر تمارين
د. حمزة محمد	غير مرددة	المادة الدراسات الحاسوب

البريد الإلكتروني: 421004034@kfupm.edu.sa

Figure 4.32 : Communication interface

4.4 Summary

This section presented the practical aspects of creating, training, and testing the prediction model. Then, it displayed the system screens for each user, along with a description of the purpose of each screen.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The project aimed to develop a web-based predictive model for evaluating Program Learning Outcomes (PLOs) attainment at University of Tabuk (UoT), utilizing predictive analytics to identify students at risk of poor performance in achieving learning outcomes. The web-based system was developed following the Waterfall methodology, and the prediction model was built using standard methodologies in data science and artificial intelligence. These efforts aimed to enhance academic achievements and prepare graduates for the labor market.

5.2 Recommendations

After completion of the system build, here are the points we aim to achieve in the future:

1. Integration with the University of Tabuk (UoT) database: This will facilitate the retrieval of student, course, faculty members, and other relevant data.
2. Acquisition of data from other subject: This will expand the prediction scope to encompass an entire program subject rather than predicting for a single subject only.

5.3 Difficulties

We faced some challenges in developing this system, which can be summarized in the following points:

1. Data acquisition due to its sensitive and highly confidential nature.
2. Dealing with the prediction model and all its related steps was not within the scope of our previous studies. It was a challenge for us to showcase our skills and our ability to acquire new knowledge.

REFERENCES

- [1] https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm Visited on 2023/9/14.
- [2] H. Albalawi, K. Al-Otaibi, and M. Alkhaybari, "Machine Learning Based Assessment of the Programs Learning Outcomes (PLOs) Attainment at the University of Tabuk (UoT), Partial fulfillment Master Thesis Master of Science in Data Science, FCIT, UoT, June-2023.
- [3] Arnold, Kimberley E, & Pistilli, Matthew. (2012). Course Signals at Purdue: using learning analytics to increase student success. Paper presented at LAK12, Vancouver, Canada.
- [4] S. Tally, T. Delworth, and K. Arnold, J. Campbell, D. Oblinger, G. McCartney, "Course Signals at Purdue University Signals students how they're doing even before the test ,," September 2009.
- [5] Kimberly E. Arnold,Matthew D. Pistilli "Course signals at Purdue: Using learning analytics to increase student success" April 2012.
- [6] E.C. Abana, "A Decision Tree Approach for Predicting Student Grades in Research Project using Wek ,," July 2019.
- [7] "UG Partner" A application in the Google Play Store created by the company G.E Manny Inc " 24. jul. 2023.
- [8] Fida K. Dankar, Mahmoud Ibrahim, "Fake It till You Make It: Guidelines for Effective Synthetic Data Generation ", 28 February 2021.

APPENDICES



```
1 <?php include './db/dbms.php'; ?>
2 <?php
3
4
5
6 $errors = [];
7
8 switch ($_GET['action']) {
9     case "sign":
10         // Validation checks
11         if (empty($_POST['Account_ID']) && !ctype_digit($_POST['Account_ID']) && strlen($_POST['Account_ID']) != 9) {
12             $errors[] = "يرجى إدخال رقم جامعي صحيح يتكون من 9 أرقام";
13
14             // إضافة تحذير إضافي لطبيعة القيمة المدخلة
15             echo ': القيمة المدخلة';
16             var_dump($_POST['Account_ID']);
17         }
18         if (empty($_POST['First_Name'])) {
19             $errors[] = "يرجى إدخال الاسم الأول";
20         }
21         if (empty($_POST['Last_Name'])) {
22             $errors[] = "يرجى إدخال الاسم الأخير";
23         }
24         if (empty($_POST['Email'])) {
25             $errors[] = "يرجى إدخال البريد الإلكتروني";
26         }
27         if (empty($_POST['Password'])) {
28             $errors[] = "يرجى إدخال كلمة السر";
29         }
30         if (empty($_POST['Mobile'])) {
31             $errors[] = "يرجى إدخال رقم الهاتف";
32         }
33         if (empty($_POST['Position'])) {
34             $errors[] = "يرجى إدخال المنصب";
35
36         // Add more validation checks as needed...
37
38         if (empty($errors)) {
39             // Check if the ID already exists
40             $existingUser = $db->prepare("SELECT * FROM accounts WHERE Account_ID = :Account_ID");
41             $existingUser->execute([':Account_ID' => $_POST['Account_ID']]);
42             $userExists = $existingUser->fetch();
43
44             if ($userExists) {
45                 $errors[] = "الرقم الجامعي موجود بالفعل في قاعدة البيانات";
46             } else {
47                 // Hash the password
48                 $password = password_hash($_POST['Password'], PASSWORD_DEFAULT);
49
50                 // Insert data into the database
51                 $sql = $db->prepare("INSERT INTO accounts (Account_ID, First_Name, Last_Name, Email, Password, Mobile, Position, Admin_ID)
52                                     VALUES (:Account_ID, :First_Name, :Last_Name, :Email, :Password, :Mobile, :Position, :Admin_ID)");
53
54                 $sql->execute([
55                     ":Account_ID" => $_POST['Account_ID'],
56                     ":First_Name" => $_POST['First_Name'],
57                     ":Last_Name" => $_POST['Last_Name'],
58                     ":Email" => $_POST['Email'],
59                     ":Password" => $password,
60                     ":Mobile" => $_POST['Mobile'],
61                     ":Position" => $_POST['Position'],
62                     ":Admin_ID" => "1",
63                 ]);
64             }
65
66             if (!empty($errors)) {
67                 // Display error messages
68                 echo "<div class='alert alert-danger'>" . implode('<br>', $errors) . "</div>";
69             }
70
71             break;
72         default:
73             echo "Invalid action";
74     }
75 }
76
77 ?>
```

Figure 5.1 : Appendices api page