



Graduation Game

Submitted by:

Md. Mazharul Islam Antor
ID: 171-35-1844
Faculty of Science and Information Technology
Department of Software Engineering
Daffodil International University

Supervised by:

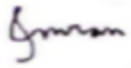
Asif Khan Shakir
Lecturer (Senior Scale)
Faculty of Science and Information Technology
Department of Software Engineering
Daffodil International University

This project report has been submitted in fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering.

APPROVAL OF PROJECT

This project titled on “**Graduation Game**”, submitted by Md. Mazharul Islam Antor (ID: 171-35-1844) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman



K. M. Imtiaz-Ud-Din
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Md Fahad Bin Zamal
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

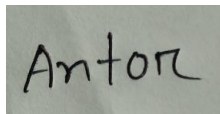


Professor Dr. Md. Nasim Akhtar
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering and Technology, Gazipur

External Examiner

DECLARATION OF PROJECT

I hereby declare that the project titled “**Graduation Game**” has been completed by me under the supervision of Mr. Asif Khan Shakir, Lecturer (Senior Scale), Department of Software Engineering, Daffodil International University, for the purpose of achieving degree of Bachelor of Science in Software Engineering from Daffodil International University. This is also declared by me that neither this project nor any part of this project has been used or submitted elsewhere for any kind of degree or awards.



Md. Mazharul Islam Antor

ID: 171-35-1844

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Certified by:



Mr. Asif Khan Shakir

Lecturer (Senior Scale)

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

ACKNOWLEDGEMENTS

At first, I want to give special thanks to my creator almighty Allah for giving me sufficient patience and of course His mercy on me for completing this project successfully in time. No plan can ever succeed without the permission of Allah. Next, I would like to give thanks to my supervisor Mr. Asif Khan Shakir for providing me the idea of this project and having trusts on me. I had to follow some game developers on YouTube to keep my project running. Big thanks to them.

ABSTRACT

This game is designed for those newcomer students who are going to admit in Software Engineering department of Daffodil International University. When playing this game, they can easily understand the curriculum activities, payment structures, marks distribution, etc., of the department. In addition, they will be familiar with semester courses. After starting official classes, they can utilize the learning knowledge from the game. There are 12 levels/semesters in the game. If a player fails in a semester, he/she cannot progress to further levels until he/she passes the failed level/semester. A player has to complete all the levels successfully to achieve the graduation certificate.

TABLE OF CONTENTS

APPROVAL OF PROJECT	I
DECLARATION OF PROJECT	II
ACKNOWLEDGEMENTS.....	III
ABSTRACT.....	IV
LIST OF FIGURES	VII
1. INTRODUCTION	1
1.1 Project Overview	1
1.2 Project Purpose.....	1
1.2.1 Background.....	1
1.2.2 Benefits and Beneficiaries	2
1.2.3 Goal	2
1.3 Stakeholders	2
1.4 Proposed System Model.....	2
1.5 Project Schedule	4
1.5.1 Gantt Chart.....	4
2. SOFTWARE REQUIREMENTS SPECIFICATION	5
2.1 Software Requirements Specification.....	5
2.2 Functional Requirements	5
2.3 Non-Functional Requirements.....	8
2.3.1 Frames rate.....	8
2.3.2 Usability.....	8
2.3.3 Response time	8
2.3.4 Required resources	9
2.3.5 Platform	9
2.3.6 Maintainability	9
2.3.7 Safety.....	9
3. SYSTEM ANALYSIS	10
3.1 Use Case Diagram	10
3.2 Use Case Description.....	12
3.3 Activity Diagram	23
3.4 Sequence Diagram	45
4. SYSTEM DESIGN SPECIFICATION.....	67

4.1 Game Scripts	67
4.2 Game Data Storage	71
4.2 Development Tools & Technologies	72
5. SYSTEM TESTING	73
5.1 Testing Features.....	73
5.1.1 Features to be Tested.....	73
5.2 Test Case	74
6. USER MANUAL	79
6.1 Player Name Entry.....	79
6.2 Pick Up ID/Bag	80
6.3 Sleeping Area	80
6.4 Player Information	81
6.5 Semester Status.....	81
6.6 Registered Courses	82
6.7 Study Area.....	82
6.8 Class Routine.....	83
6.9 Exam Schedules.....	83
6.10 Fees	84
6.11 Semester Result	84
6.12 Buy Bus Ticket	85
6.13 Enter into Bus	85
6.14 Enter into University.....	86
6.15 Enter into Classroom.....	86
6.16 Class Moments	87
6.17 Class Activity Available.....	87
6.18 Course Marks.....	88
7. PROJECT SUMMARY	89
7.1 Limitations.....	89
7.2 Obstacles & Achievement.....	89
7.3 Future Improvement.....	89
7.4 GitHub Link	90
7.5 References	90
7.6 Plagiarism Report	91

LIST OF FIGURES

Point	Name of Figure	Page No.
Figure 1.1	Spiral model	3
Figure 1.2	Gantt chart	4
Figure 3.1	Use Case Diagram	10
Figure 3.2	Activity Diagram for ‘Enter player’s name’	23
Figure 3.3	Activity Diagram for ‘Check player’s information’	24
Figure 3.4	Activity Diagram for ‘View semester status’	25
Figure 3.5	Activity Diagram for ‘Register semester’	26
Figure 3.6	Activity Diagram for ‘View semester courses’	27
Figure 3.7	Activity Diagram for ‘View course marks’	28
Figure 3.8	Activity Diagram for ‘Study courses’	29
Figure 3.9	Activity Diagram for ‘View class routine’	30
Figure 3.10	Activity Diagram for ‘Pay fees’	31
Figure 3.11	Activity Diagram for ‘View exam schedules’	32
Figure 3.12	Activity Diagram for ‘View semester result’	33
Figure 3.13	Activity Diagram for ‘View fees structure’	34
Figure 3.14	Activity Diagram for ‘Pick up id/bag’	35
Figure 3.15	Activity Diagram for ‘Buy bus ticket’	36
Figure 3.16	Activity Diagram for ‘Take a seat on bus’	37
Figure 3.17	Activity Diagram for ‘Enter into university’	38
Figure 3.18	Activity Diagram for ‘Enter into classroom’	39
Figure 3.19	Activity Diagram for ‘Perform presentation’	40
Figure 3.20	Activity Diagram for ‘Submit assignment’	41

Figure 3.21	Activity Diagram for ‘Give quiz’	42
Figure 3.22	Activity Diagram for ‘Give mid/final exam’	43
Figure 3.23	Activity Diagram for ‘Go to sleep’	44
Figure 3.24	Sequence Diagram for ‘Enter player’s name’	45
Figure 3.25	Sequence Diagram for ‘Check player’s information’	46
Figure 3.26	Sequence Diagram for ‘View semester status’	47
Figure 3.27	Sequence Diagram for ‘Register semester’	48
Figure 3.28	Sequence Diagram for ‘View semester courses’	49
Figure 3.29	Sequence Diagram for ‘View course marks’	50
Figure 3.30	Sequence Diagram for ‘Study courses’	51
Figure 3.31	Sequence Diagram for ‘View class routine’	52
Figure 3.32	Sequence Diagram for ‘Pay fees’	53
Figure 3.33	Sequence Diagram for ‘View exam schedules’	54
Figure 3.34	Sequence Diagram for ‘View semester result’	55
Figure 3.35	Sequence Diagram for ‘View fees structure’	56
Figure 3.36	Sequence Diagram for ‘Pick up id/bag’	57
Figure 3.37	Sequence Diagram for ‘Buy bus ticket’	58
Figure 3.38	Sequence Diagram for ‘Take a seat on bus’	59
Figure 3.39	Sequence Diagram for ‘Enter into university’	60
Figure 3.40	Sequence Diagram for ‘Enter into classroom’	61
Figure 3.41	Sequence Diagram for ‘Perform presentation’	62
Figure 3.42	Sequence Diagram for ‘Submit assignment’	63
Figure 3.43	Sequence Diagram for ‘Give quiz’	64

Figure 3.44	Sequence Diagram for ‘Give mid/final exam’	65
Figure 3.45	Sequence Diagram for ‘Go to sleep’	66

CHAPTER 1

INTRODUCTION

1.1 Project Overview

“Game: Graduation” is a game runs on windows platform. When entering the game for the first time game player will notice an input box to register his/her name. He will be registered as a student of Software Engineering department at Daffodil International University. There are twelve levels/semester in the game. You have to pass a semester to enter into next semester. Under each semester player can view his/her registered courses, semester class routine, exam schedule. For attending mid/final player has to pay required fees from fees section. In study section, player can study courses. Course studies will affect his/her semester results. Player has to attend classes to submit assignment, perform presentation or give quiz. Player can sleep for given hours to pass his/her game time quickly. University is far away from home, so player can use bus as transportation medium. Also player can overview the fees structure of the department. While playing the game, player can feel the graduation experiences of his/her department.

1.2 Project Purpose

This game provides the best graduation experiences. User can get overall idea about the curriculum systems of Software Engineering department. He/she can gather fees related idea, which courses are taught in which semester. He will be able to get an idea about course activities, marks distribution. He/she will be able to know about class time allotment and class frequencies in a week. Furthermore, it is a game of learning with enjoyment for a newcomer student.

1.2.1 Background

Students should know about his/her department’s curriculum activities, fees structure, class time allotment, class frequencies per week, marks distribution, assigned courses, etc. Therefore, there should exist such a game by playing which he/she can gather enough knowledge about these things. Before admitting to the department, a newcomer can gather overall knowledge easily. He/she is playing the game and learning everything with a fun mode. It is the main reason behind this game project.

1.2.2 Benefits and Beneficiaries

By playing this game, a newcomer student will be able to get an overall idea about the department's teaching procedures. In addition, those students who are planning to admit in Software Engineering department will be benefited mostly.

1.2.3 Goal

Our main goal is to provide the newcomer student proper knowledge about the Software Engineering department with an enjoyable way while he/she is playing the game.

1.3 Stakeholders

In this project stakeholders are:

1. Project supervisor
2. Project developer
3. Users (playing the game)

1.4 Proposed System Model

When developing a game, a model should be followed by the developer. In my case, I have followed spiral model to fulfill the project demand. I have to design the game scenario at first. After putting the scenario in the game, I must need the player to interact with the environment or other game functions. After completing coding part, there comes testing part whether the game is progressing or not. Then Evaluation matters most whether I will go with the game progress whatever I have done till now. This cycle happens repeatedly because when developing a game you need to go back for any design or coding correction. After final evaluation, you can declare the end of the project.

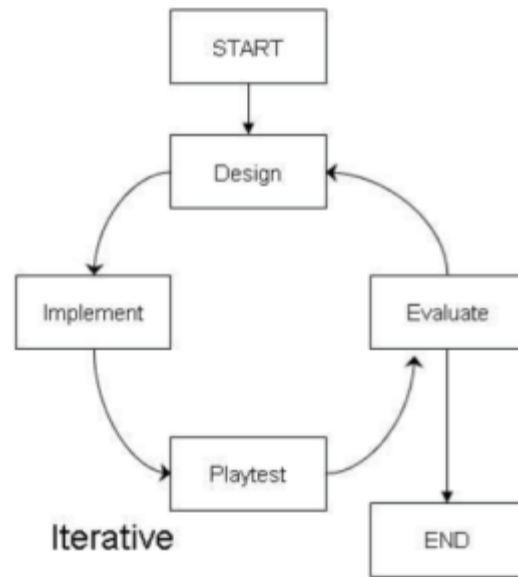


Figure 1.1: Spiral model

1.5 Project Schedule

Project schedule means an overall idea about the project processing time from start to end. To complete the project effectively there should exist a time schedule otherwise progressing will be halt.

1.5.1 Gantt Chart

Weeks→ Works↓	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Project Planning																
Requirement Collection																
Requirement Analysis																
Feasibility Study																
UML Diagrams																
Design																
Implementation																
Testing																
Evaluation																
Documentation																

Figure 1.2: Gantt chart

CHAPTER 2

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Software Requirements Specification

Software requirement specification describes how a software system needs to be developed. In many cases, it prevents project failure. It provides a detailed overview of the mentioned product and targeted audience, goals. It includes all functionalities and features of a product which need to be fulfilled as per as stakeholder needs.

2.2 Functional Requirements

FR-01	Enter player's name
Description	Player can register his/her game name when first entering the game.
Stakeholders	Player
Priority	Medium

FR-02	Check player's information
Description	Player can check his/her information such as his/her name, id number, department and university name.
Stakeholders	Player
Priority	Low

FR-03	View semester status
Description	Player can check locked semester, or which semester he/she has completed or registered.
Stakeholders	Player
Priority	High

FR-04	Register semester
Description	Player can register semester within deadline.
Stakeholders	Player
Priority	High

FR-05	View semester courses
Description	Player can view semester courses.

Stakeholders	Player
Priority	Medium

FR-06	View course marks
Description	Player can view course marks.
Stakeholders	Player
Priority	Medium

FR-07	Study courses
Description	Player can study assigned courses for running semester.
Stakeholders	Player
Priority	High

FR-08	View class routine
Description	Player can check class routine.
Stakeholders	Player
Priority	High

FR-09	Pay fees
Description	Player can pay mid-term and final-term fees.
Stakeholders	Player
Priority	Medium

FR-10	View exam schedules
Description	Player can check mid and final exam schedules to attend the exam in time.
Stakeholders	Player
Priority	High

FR-11	View semester result
Description	Player can check published semester result.
Stakeholders	Player
Priority	Low

FR-12	View fees structure
Description	Player can check fees structure determined by the department.
Stakeholders	Player
Priority	Low

FR-13	Pick up id/bag
Description	Player can pick up his/her id and bag from his/her reading table.
Stakeholders	Player
Priority	Low

FR-14	Buy bus ticket
Description	Player can buy bus ticket from bus counter.
Stakeholders	Player
Priority	Low

FR-15	Take a seat on bus
Description	Player can enter into bus and take a seat if he/she has a ticket.
Stakeholders	Player
Priority	Low

FR-16	Enter into university
Description	Player can enter into university if he/she has id card.
Stakeholders	Player
Priority	Low

FR-17	Enter into classroom
Description	Player can enter into classroom matching class routine.
Stakeholders	Player
Priority	Medium

FR-18	Perform presentation
Description	Player can perform presentation if available on that day.
Stakeholders	Player
Priority	High

FR-19	Submit assignment
Description	Player can submit assignment if available on that day.
Stakeholders	Player
Priority	High

FR-20	Give quiz
Description	Player can give quiz if available on that day.
Stakeholders	Player
Priority	High

FR-21	Give mid/final exam
Description	Player can give mid/final exam according to given schedule.
Stakeholders	Player
Priority	High

FR-22	Go to sleep
Description	Player can sleep for given hours such as six, four, two or one hour.
Stakeholders	Player
Priority	Low

2.3 Non-Functional Requirements

2.3.1 Frames rate

- The game is easily playable at 30 FPS (frame per second).
- Frames dropping are not occurring when playing the game.

2.3.2 Usability

- Every functions are workable when player interacts.
- User experiences 3D graphics in the game.

2.3.3 Response time

- It takes less than 0.5 seconds for response when player interacts with a function.
- Player can play the game with a relax mode with great response time.

2.3.4 Required resources

- The game is playable with minimum of one GB RAM.
- This game does not take more than five to six GB.
- This game does not take too much graphics space.

2.3.5 Platform

- This game is made for windows platform only.

2.3.6 Maintainability

- The game is overlooked twice per week to find any kind of bugs or functional failures.

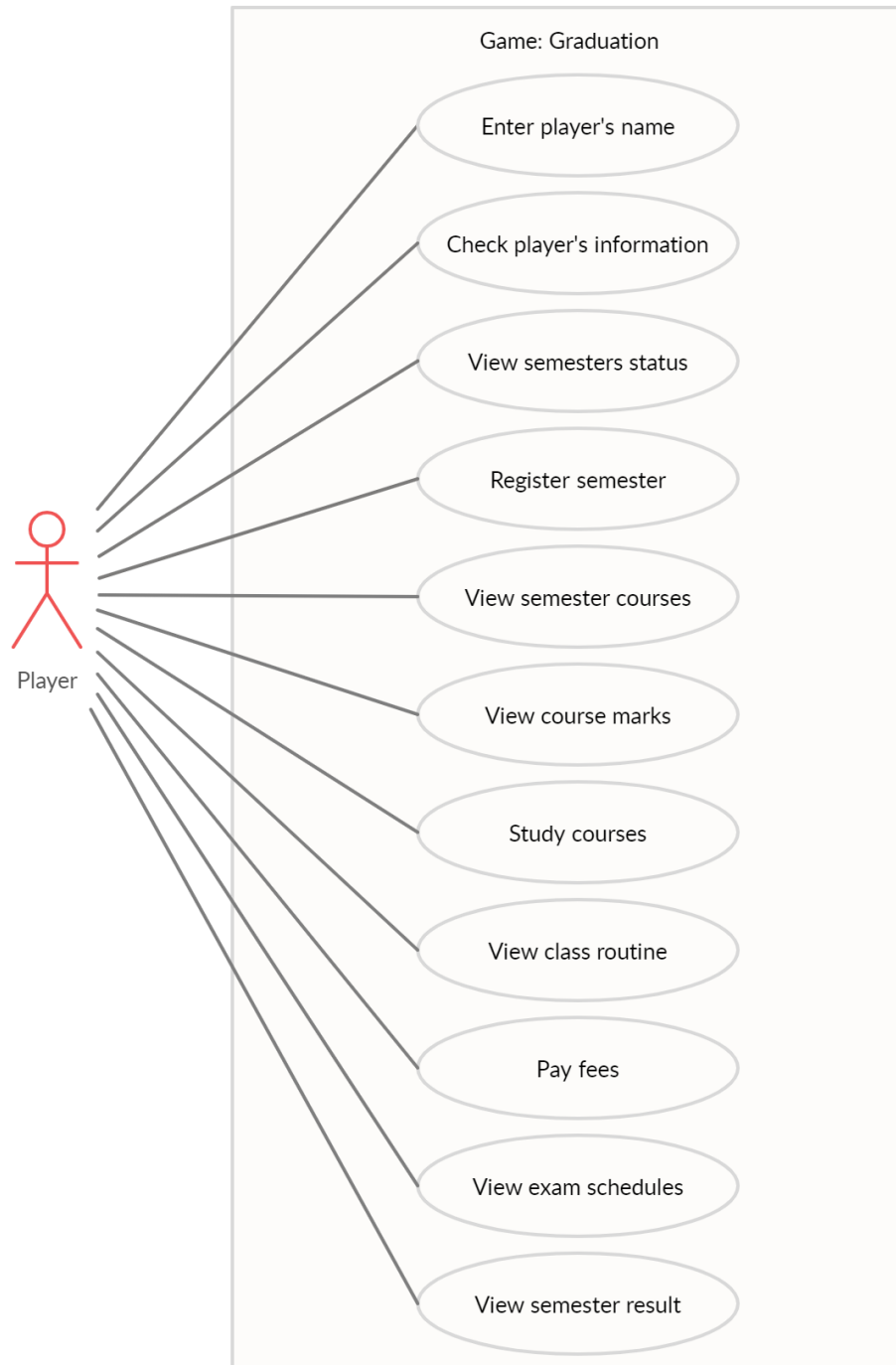
2.3.7 Safety

- The game does not harm user's health and eyes.
- This game does not show such things, which can have negative impacts to society.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Use Case Diagram



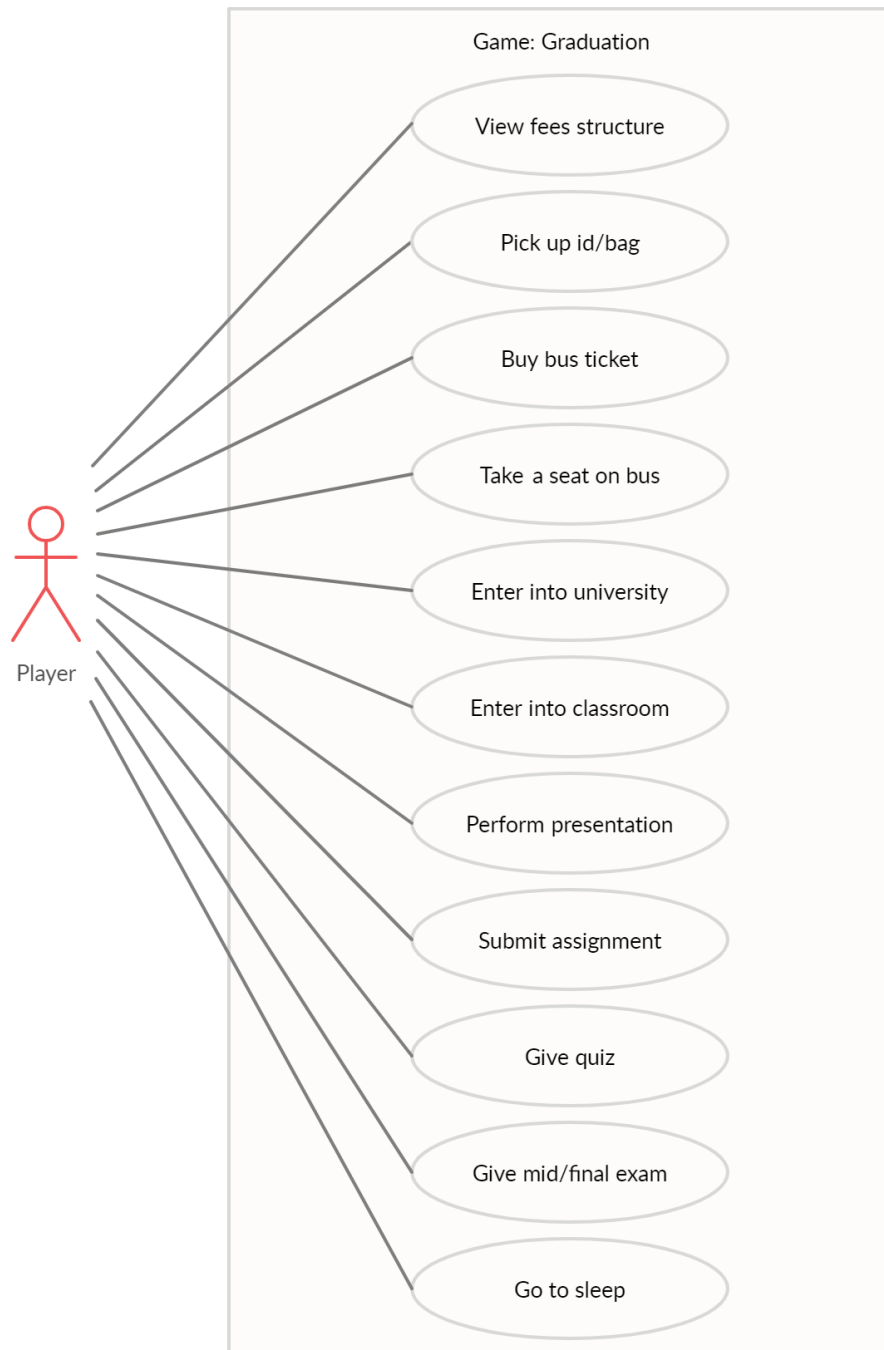


Figure 3.1: Use Case Diagram

3.2 Use Case Description

Use Case No.	01	
Use Case Name	Enter player's name	
Goal	Player can register his/her name.	
Preconditions	Player has to open the game.	
Success End Condition	Player can enroll himself/herself as a student of the department of Software Engineering at Daffodil International University.	
Failed End Condition	Player cannot enroll himself/herself as a student.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player opens the game at first time.	
Description/Main Success Scenario	Step	Action
	1	Player opens the game at first time.
	2	Player gets the input box for entering his/her name.
	3	Player puts his/her name and register himself/herself.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	02	
Use Case Name	Check player's information	
Goal	Player can check his/her information.	
Preconditions	Player has to play the game.	
Success End Condition	Player can check his/her information.	
Failed End Condition	Player cannot check his/her information.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'My Information' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'My Information' button.
	2	Player views his/her information.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	03	
Use Case Name	View semester status	
Goal	Player can check locked semester, or which semester he/she has completed or registered.	
Preconditions	Player has to play the game.	
Success End Condition	Player can check semester status whether it remains locked, registered or completed.	
Failed End Condition	Player cannot check semester status.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Education Corner' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Education Corner' button.
	2	Player views semester status
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	04	
Use Case Name	Register semester	
Goal	Player can continue his/her studies.	
Preconditions	Registration date must be opened.	
Success End Condition	Player can register semester.	
Failed End Condition	Player cannot register semester.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Register' named button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Register' named button.
	2	Player is registered for this semester.
	3	Player can continue his/her studies.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	05	
Use Case Name	View semester courses	
Goal	Player can check which courses are assigned for the semester.	
Preconditions	Semester must be registered.	
Success End Condition	Player can view semester courses.	
Failed End Condition	Player fails to view semester courses.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Courses' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Courses' button.
	2	Player views semester courses.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	06	
Use Case Name	View course marks	
Goal	Player can check course marks.	
Preconditions	Semester must be registered.	
Success End Condition	Player views course marks.	
Failed End Condition	Player fails to view course marks.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'View' button inside of 'Courses' area.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'View' button inside of 'Courses' area.
	2	Player views course marks such as attendance, three assignments, two presentations, three quizzes, mid, final. Attendance, assignments, presentations, mid and final marks are viewable after semester result.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	07	
Use Case Name	Study courses	
Goal	Player can study assigned courses for running semester.	
Preconditions	Semester must be registered and running.	
Success End Condition	Player can study courses.	
Failed End Condition	Player cannot study courses.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Study' button inside of 'Study Area'.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Study' button inside of 'Study Area'.
	2	Then player can see different types of category of studying named as 'Regular study', 'Prepare assignment', 'Presentation preparation', 'Quiz study', 'Mid-term study' and 'Final-term study'.
	3	Player spends time on a category clicking on 'Spend' button until per day maximum spend time not crossed.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Player cannot spend more than given time per day.

Use Case No.	08	
Use Case Name	View class routine	
Goal	Player can check class routine.	
Preconditions	Semester must be registered.	
Success End Condition	Player can check class routine.	
Failed End Condition	Player fails to check class routine.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Class Routine' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Class Routine' button.
	2	Player checks class routine.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality	Step	Requirement

Requirements	1	N/A
---------------------	---	-----

Use Case No.	09	
Use Case Name	Pay fees	
Goal	Player can pay mid-term and final-term fees.	
Preconditions	Semester must be registered.	
Success End Condition	Player can successfully pay required fees.	
Failed End Condition	Player fails to pay required fees.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Fees' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Fees' button.
	2	Player pays required mid-term or final-term fees for attending exam.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Player has to pay exact given amount.

Use Case No.	10	
Use Case Name	View exam schedules	
Goal	Player can check mid-term and final-term exam schedules.	
Preconditions	Semester must be registered.	
Success End Condition	Player can successfully view exam schedules.	
Failed End Condition	Player fails to view exam schedules.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Exam Schedules' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Exam Schedules' button.
	2	Player checks mid-term or final-term exam schedules.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality	Step	Requirement

Requirements	1	N/A
---------------------	---	-----

Use Case No.	11	
Use Case Name	View semester result	
Goal	Player can check semester result.	
Preconditions	Semester must be completed.	
Success End Condition	Player can successfully check result.	
Failed End Condition	Player cannot check result.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Result' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Result' button.
	2	Player chooses semester from dropdown list.
	3	Player gets the result.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	If semester is not completed, result will not be shown.

Use Case No.	12	
Use Case Name	View fees structure	
Goal	Player can check fees structure.	
Preconditions	Player has to play the game.	
Success End Condition	Player can successfully check fees structure.	
Failed End Condition	Player cannot check fees structure.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player clicks on 'Fees Structure' button.	
Description/Main Success Scenario	Step	Action
	1	Player clicks on 'Fees Structure' button.
	2	Player views fees structure determined by the department.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality	Step	Requirement

Requirements	1	N/A
---------------------	---	-----

Use Case No.	13	
Use Case Name	Pick up id/bag	
Goal	Player can pick up his/her university id card or bag.	
Preconditions	Player has to play the game.	
Success End Condition	Player can successfully pick up his/her id/bag.	
Failed End Condition	Player cannot pick up id/bag.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword or 'G' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword and 'G' keyword from keyboard.
	2	Player picks up id and bag respectively.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	14	
Use Case Name	Buy bus ticket	
Goal	Player can buy bus ticket from counter.	
Preconditions	Player has to stand in front of ticket counter.	
Success End Condition	Player can successfully buys ticket.	
Failed End Condition	Player cannot buy ticket for not having sufficient money.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword from keyboard.
	2	Player buys up/down ticket.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality	Step	Requirement

Requirements	1	N/A
---------------------	---	-----

Use Case No.	15	
Use Case Name	Take a seat on bus	
Goal	Player can get into the bus and take seat.	
Preconditions	Player has to stand in front of bus door.	
Success End Condition	Player can successfully get into the bus take a seat.	
Failed End Condition	Player cannot get into the bus take a seat.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword from keyboard.
	2	Player enters into the bus and takes a seat.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Without a bus ticket, player cannot enter into the bus.

Use Case No.	16	
Use Case Name	Enter into university	
Goal	Player can enter into university.	
Preconditions	Player has to stand in front of university gate.	
Success End Condition	Player can enter into the university.	
Failed End Condition	Player cannot enter into the university.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword from keyboard.
	2	Player enters into the university.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Player must have id card to enter.

Use Case No.	17	
Use Case Name	Enter into classroom	
Goal	Player can enter into classroom.	
Preconditions	Player has to stand before classroom door.	
Success End Condition	Player can enter into the classroom.	
Failed End Condition	Player cannot enter into the classroom.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword from keyboard.
	2	Player enters into the classroom.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Player must match class routine to enter.

Use Case No.	18	
Use Case Name	Perform presentation	
Goal	Player can perform presentation.	
Preconditions	Player has to attend the presentation class.	
Success End Condition	Player can successfully perform presentation.	
Failed End Condition	Player fails to perform presentation.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword from keyboard.
	2	Player performs presentation.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	19	
Use Case Name	Submit assignment	
Goal	Player can submit assignment.	
Preconditions	Player has to attend the assignment submission class.	
Success End Condition	Player can successfully submit assignment.	
Failed End Condition	Player fails to submit assignment.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword from keyboard.
	2	Player submits assignment.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	20	
Use Case Name	Give quiz	
Goal	Player can give quiz.	
Preconditions	Player has to attend the quiz class.	
Success End Condition	Player can successfully give quiz.	
Failed End Condition	Player fails to give quiz.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F' keyword from keyboard.
	2	Player gives quiz.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

Use Case No.	21	
Use Case Name	Give mid/final exam	
Goal	Player can give exam.	
Preconditions	Player has to attend the exam room.	
Success End Condition	Player can successfully give exam.	
Failed End Condition	Player fails to give exam.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player enters into the exam hall.	
Description/Main Success Scenario	Step	Action
	1	Player enters into the exam hall.
	2	Player gives exam.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Exam schedules must be matched.

Use Case No.	22	
Use Case Name	Go to sleep	
Goal	Player can go to sleep.	
Preconditions	Player has to stand in front of bed.	
Success End Condition	Player can successfully go to sleep.	
Failed End Condition	Player cannot go to sleep.	
Primary Actors	Player	
Secondary Actors	N/A	
Trigger	Player presses down 'F', 'G', 'H' or 'J' keyword from keyboard.	
Description/Main Success Scenario	Step	Action
	1	Player presses down 'F', 'G', 'H' or 'J' keyword from keyboard.
	2	Player sleeps for six, four, two, one hour respectively for the 'F', 'G', 'H' and 'J' keyword.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	N/A

3.3 Activity Diagram

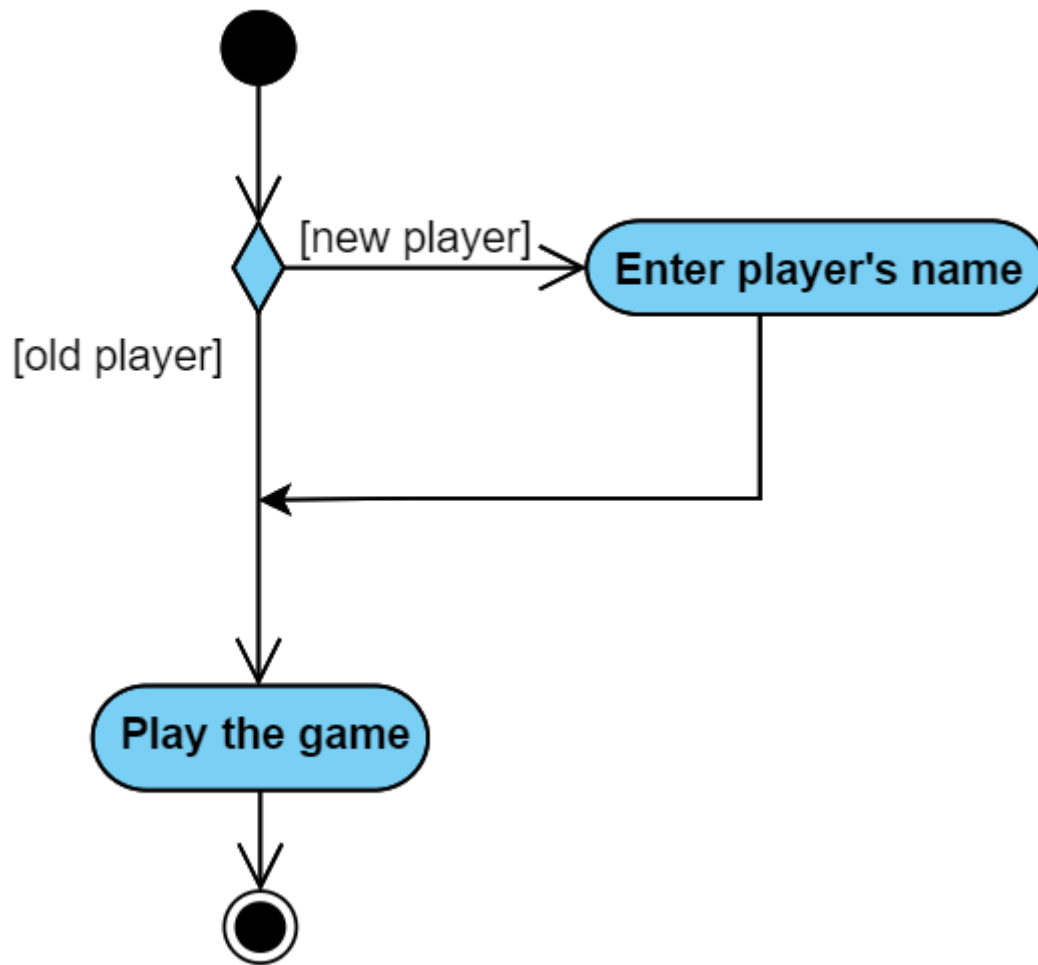


Figure 3.2: Activity Diagram for 'Enter player's name'

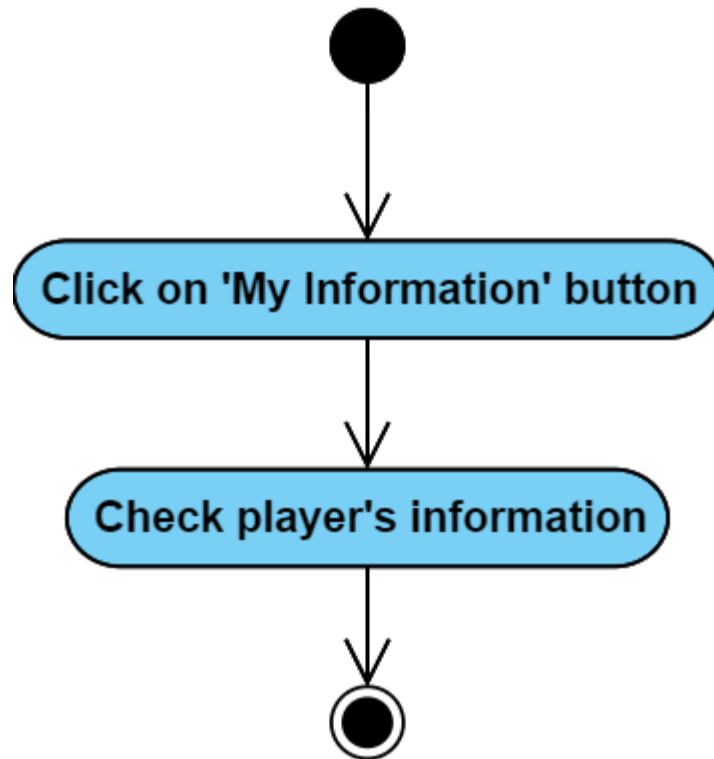


Figure 3.3: Activity Diagram for 'Check player's information'

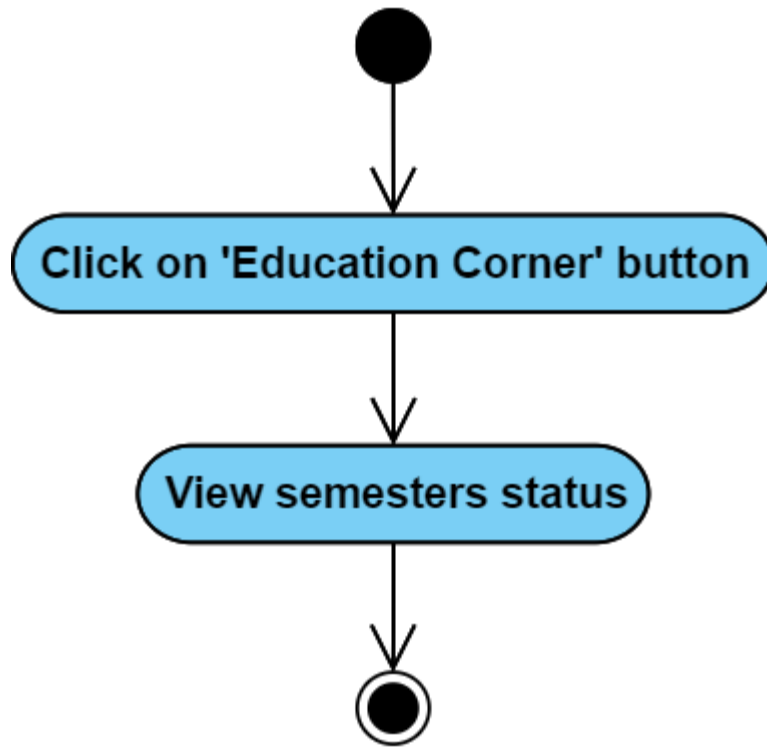


Figure 3.4: Activity Diagram for 'View semester status'

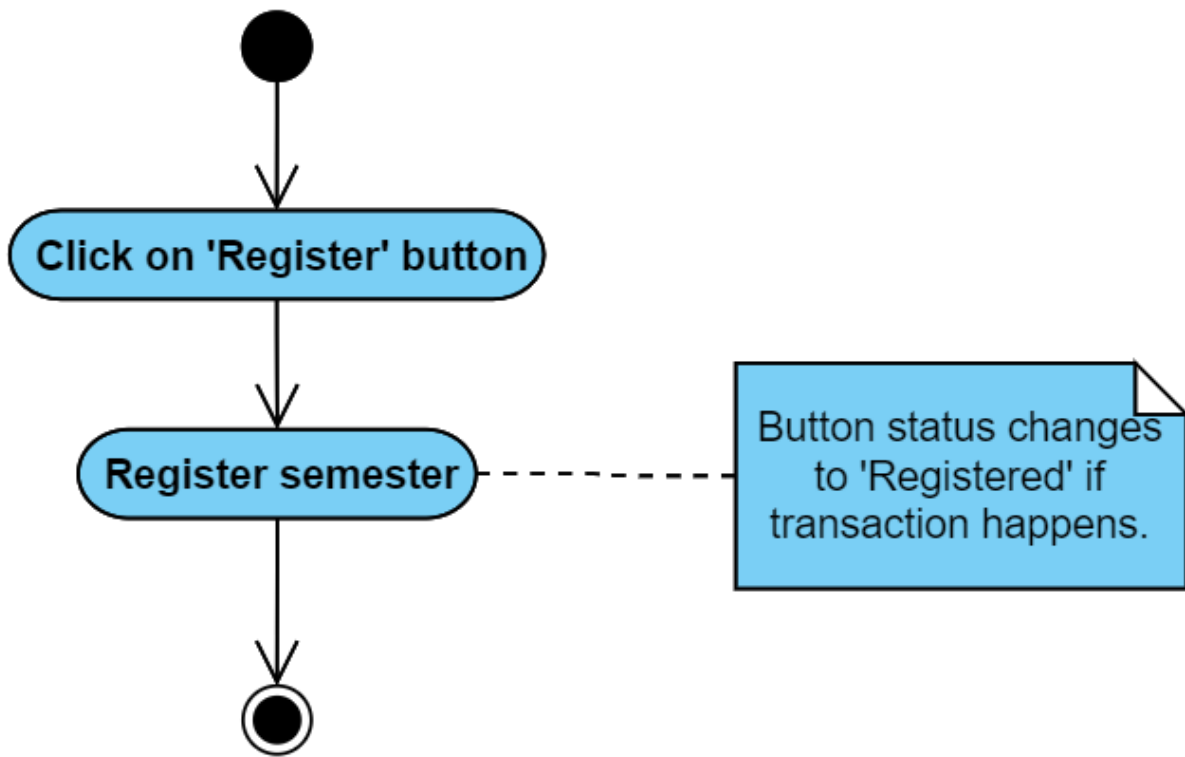


Figure 3.5: Activity Diagram for 'Register semester'

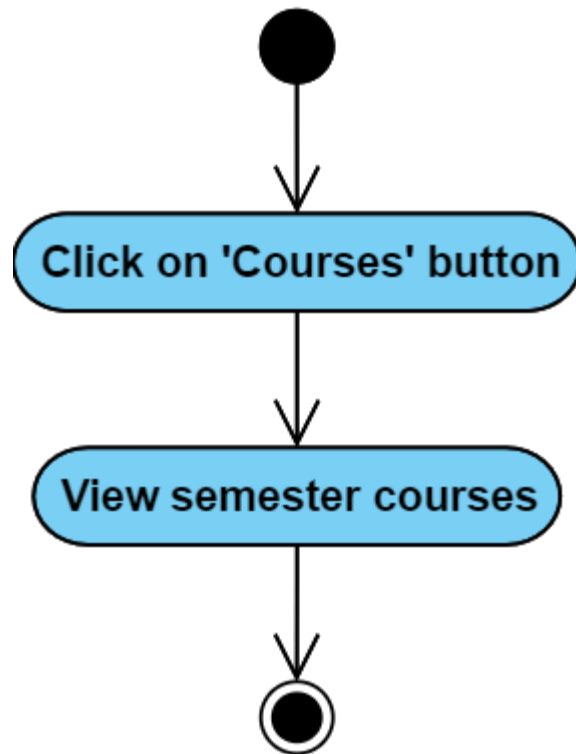


Figure 3.6: Activity Diagram for 'View semester courses'

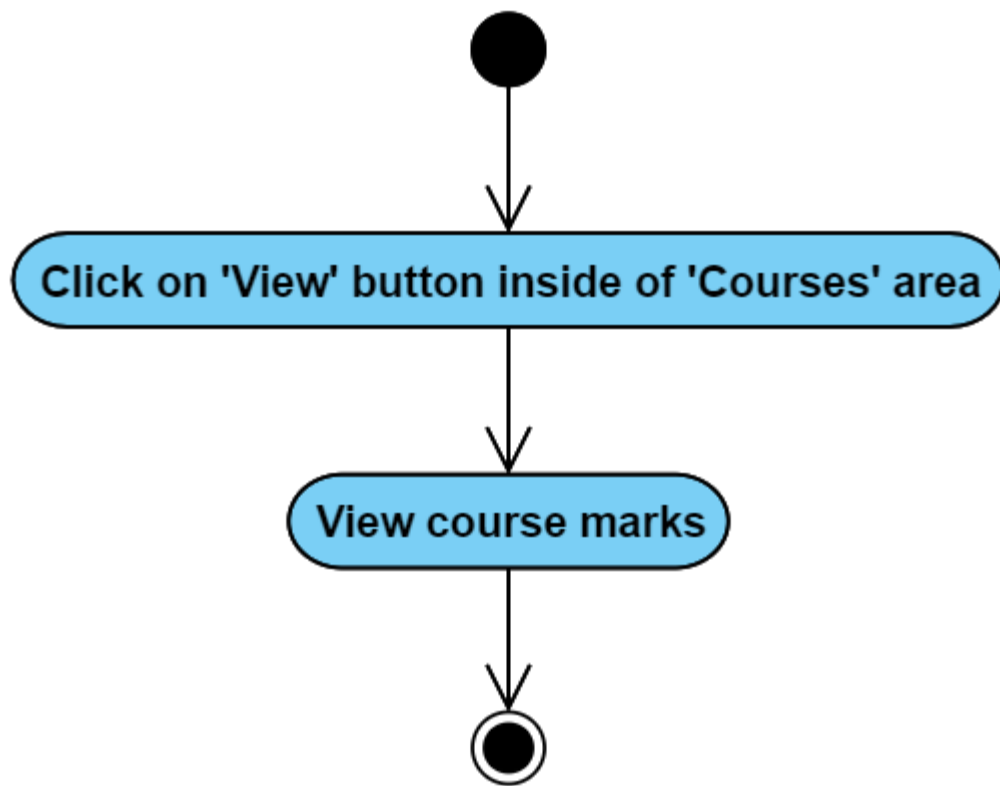


Figure 3.7: Activity Diagram for 'View course marks'

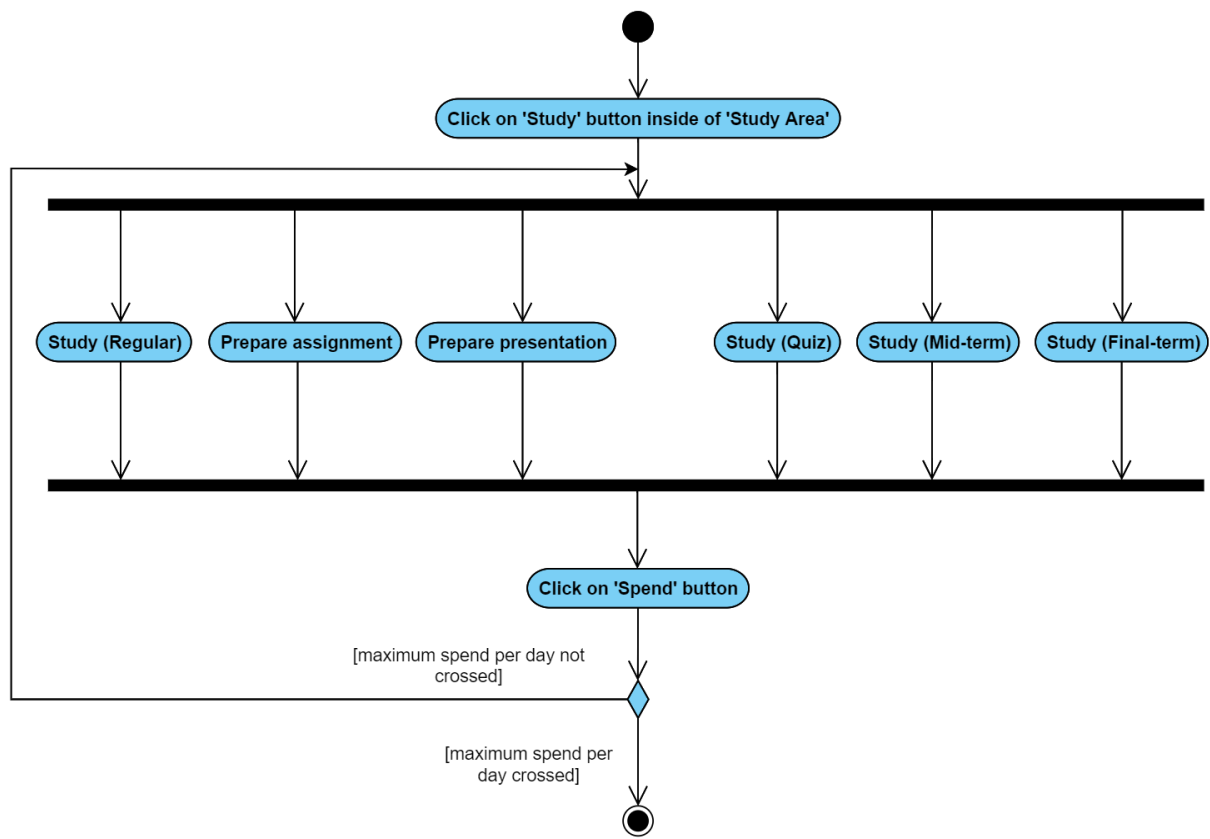


Figure 3.8: Activity Diagram for ‘Study courses’

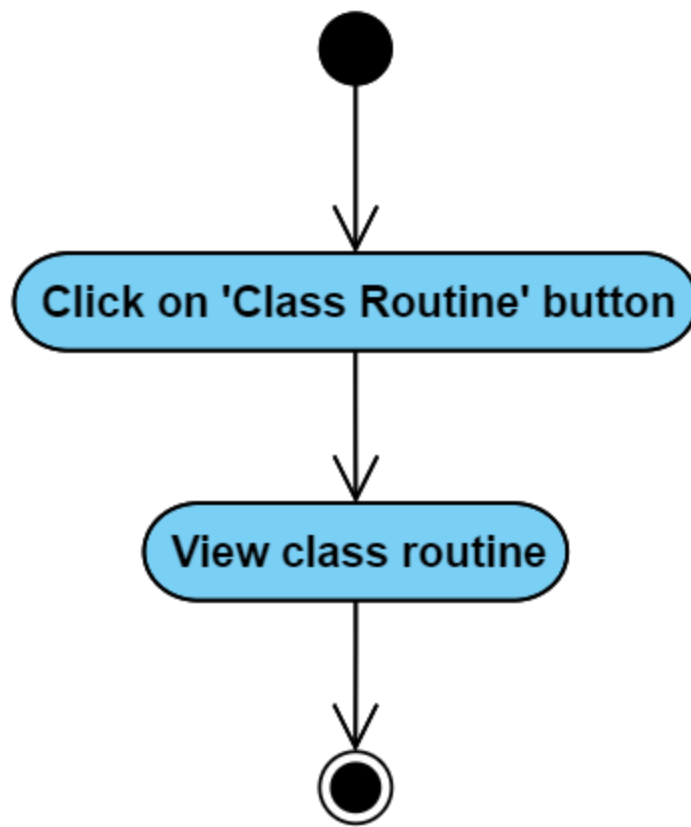


Figure 3.9: Activity Diagram for 'View class routine'

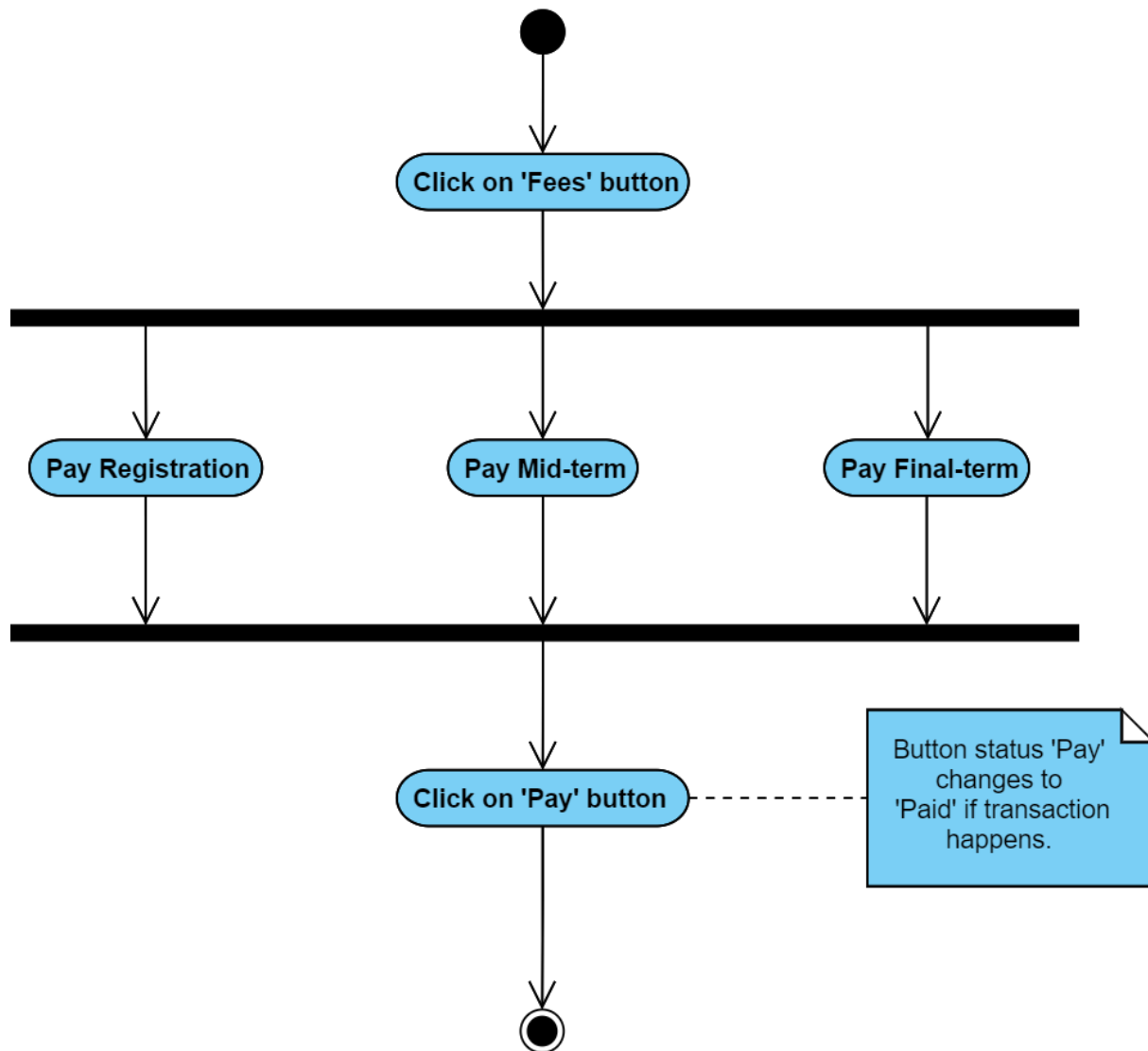


Figure 3.10: Activity Diagram for 'Pay fees'

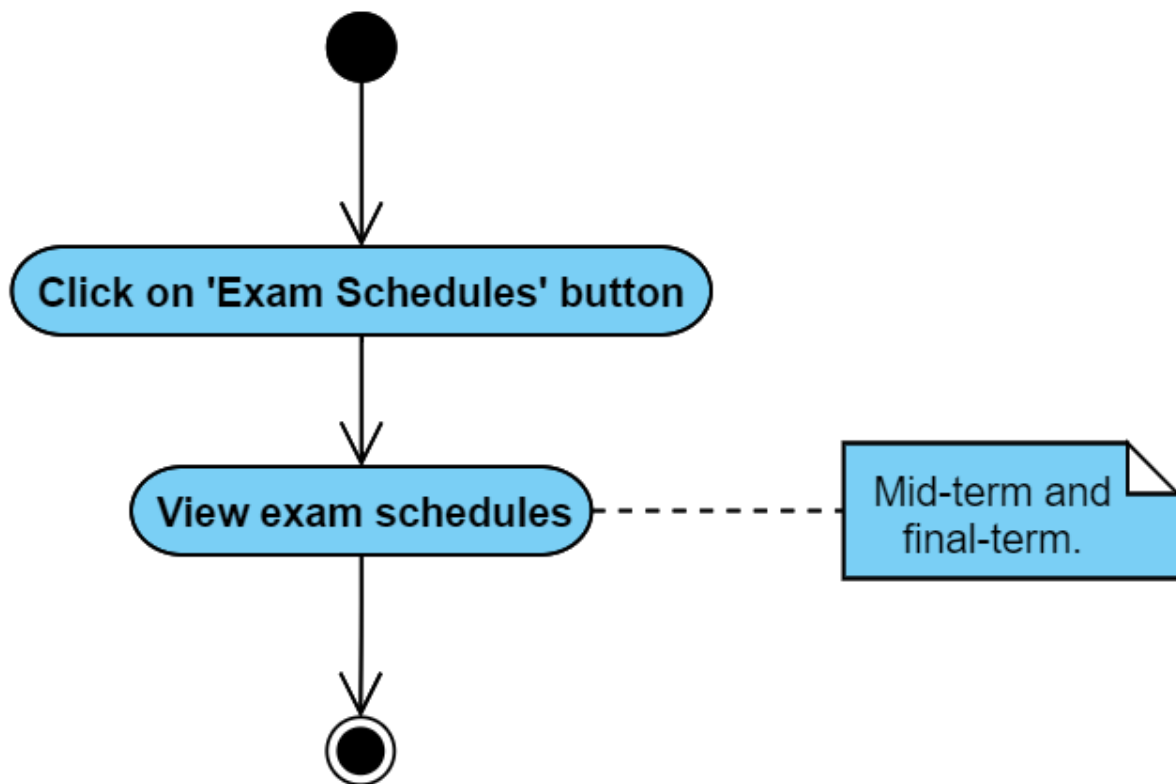


Figure 3.11: Activity Diagram for 'View exam schedules'

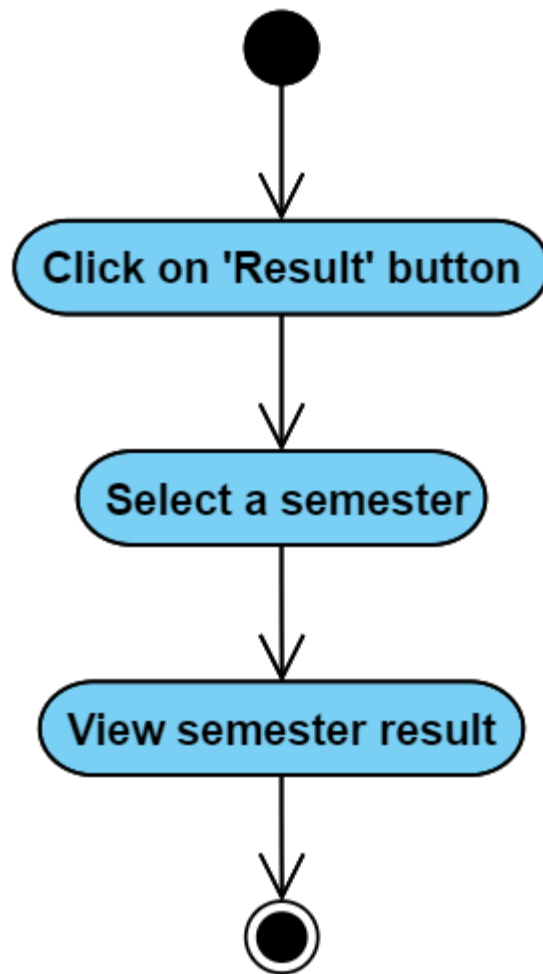


Figure 3.12: Activity Diagram for 'View semester result'

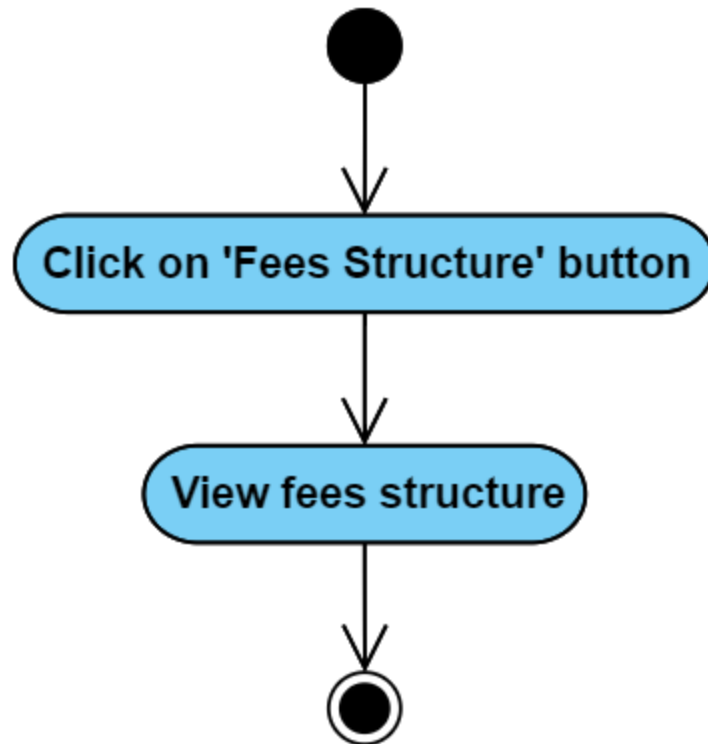


Figure 3.13: Activity Diagram for 'View fees structure'

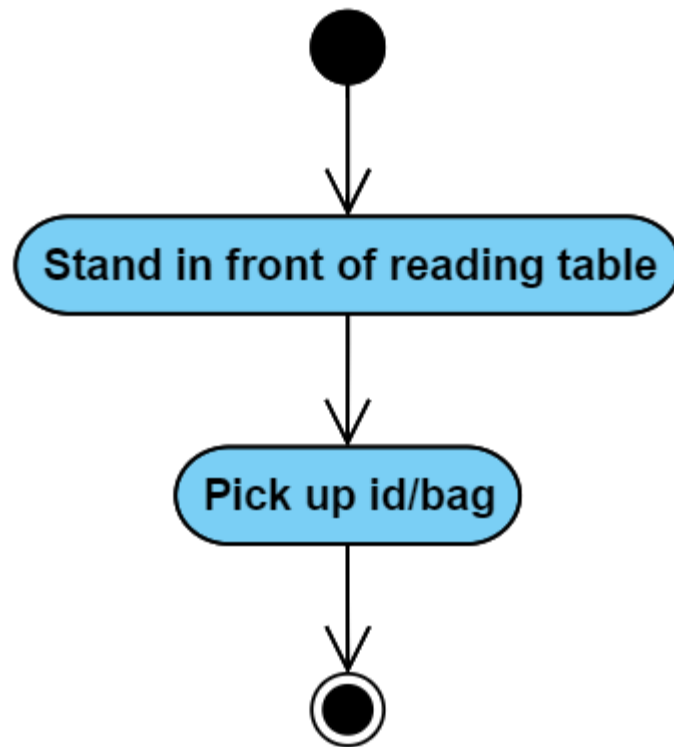


Figure 3.14: Activity Diagram for 'Pick up id/bag'

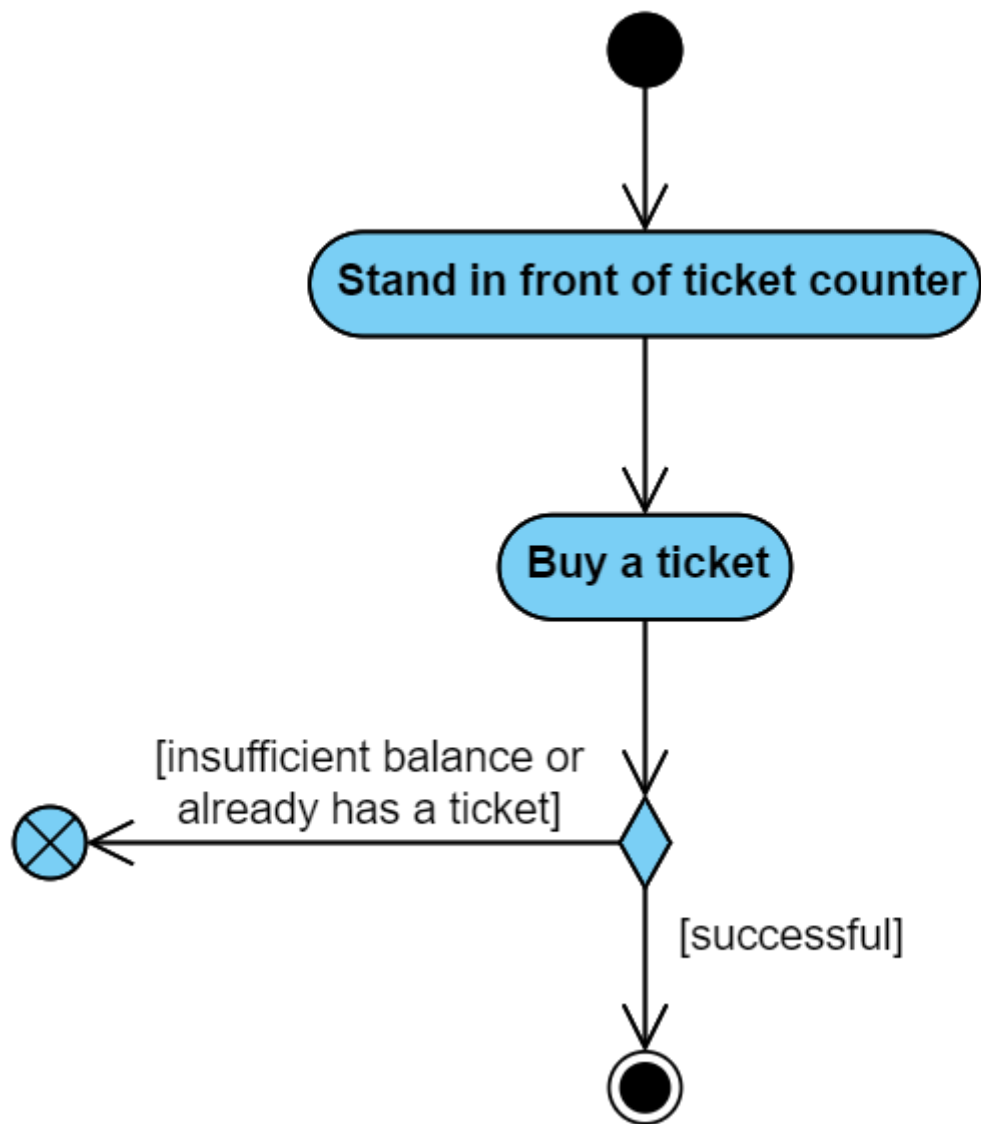


Figure 3.15: Activity Diagram for ‘Buy bus ticket’

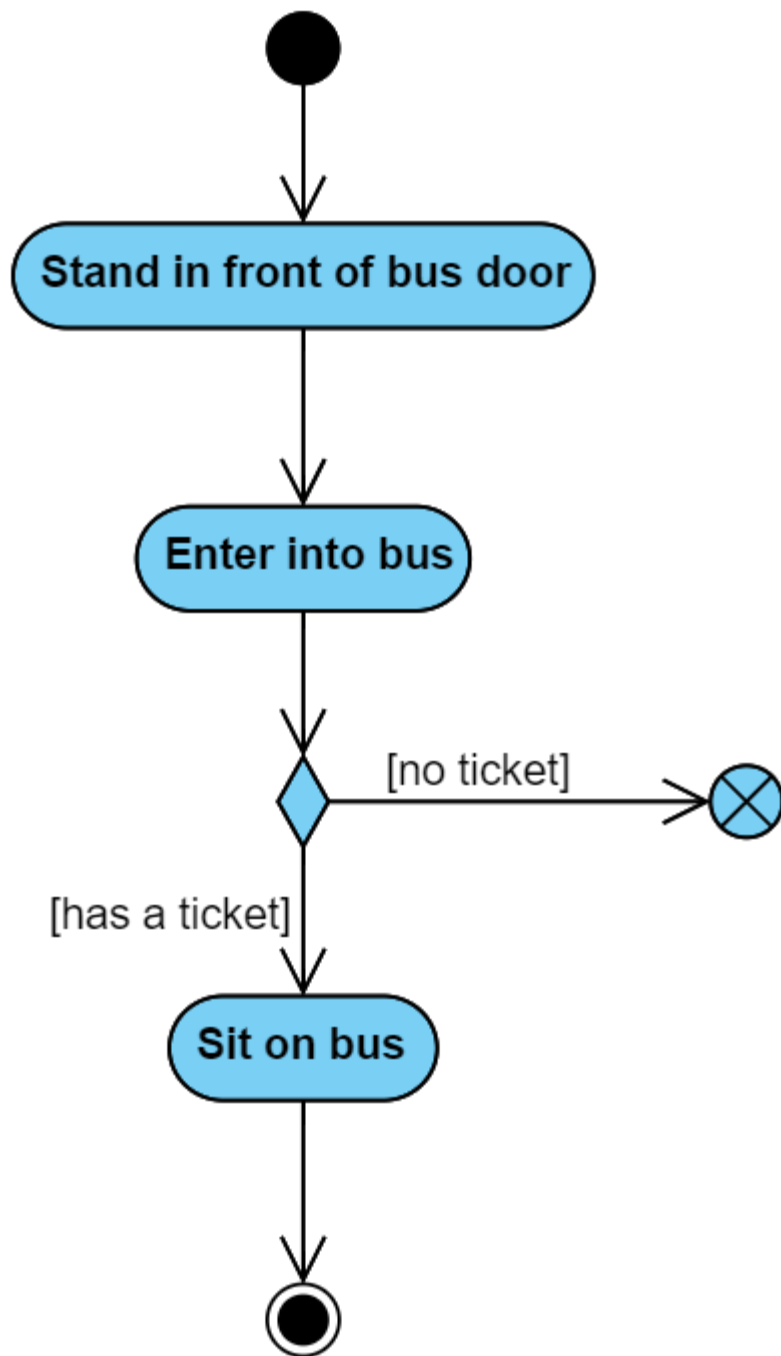


Figure 3.16: Activity Diagram for 'Take a seat on bus'

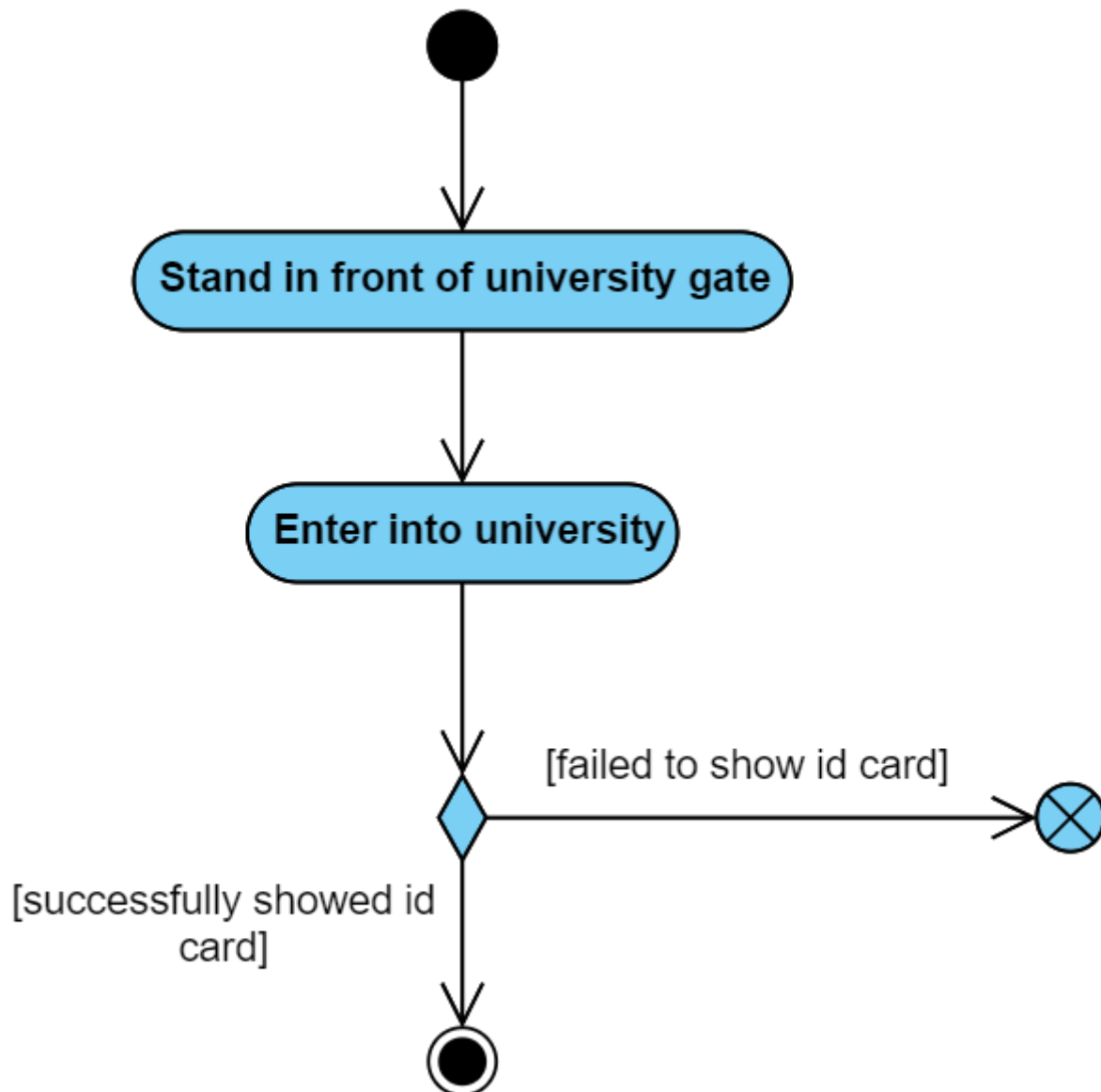


Figure 3.17: Activity Diagram for 'Enter into university'

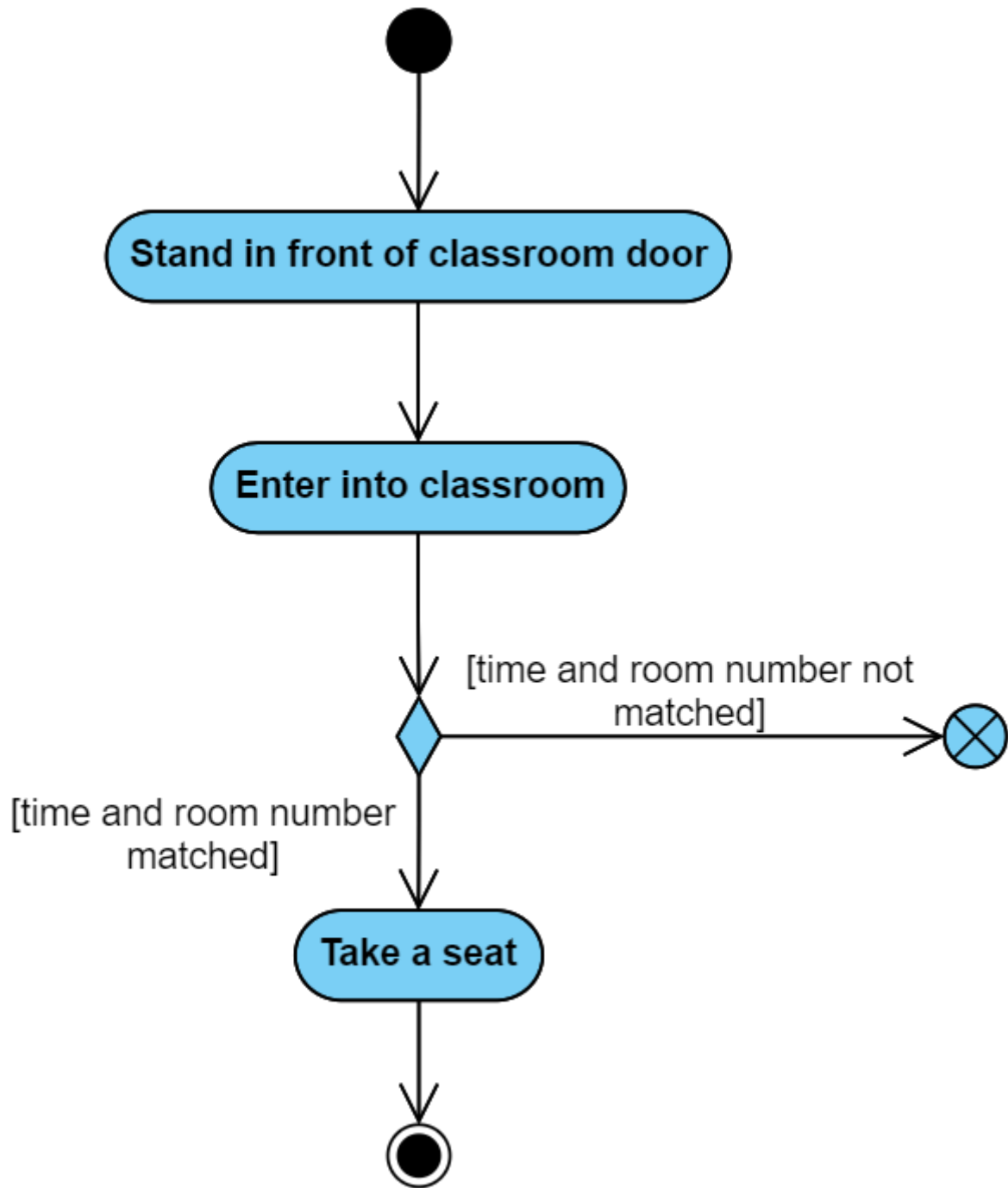


Figure 3.18: Activity Diagram for 'Enter into classroom'

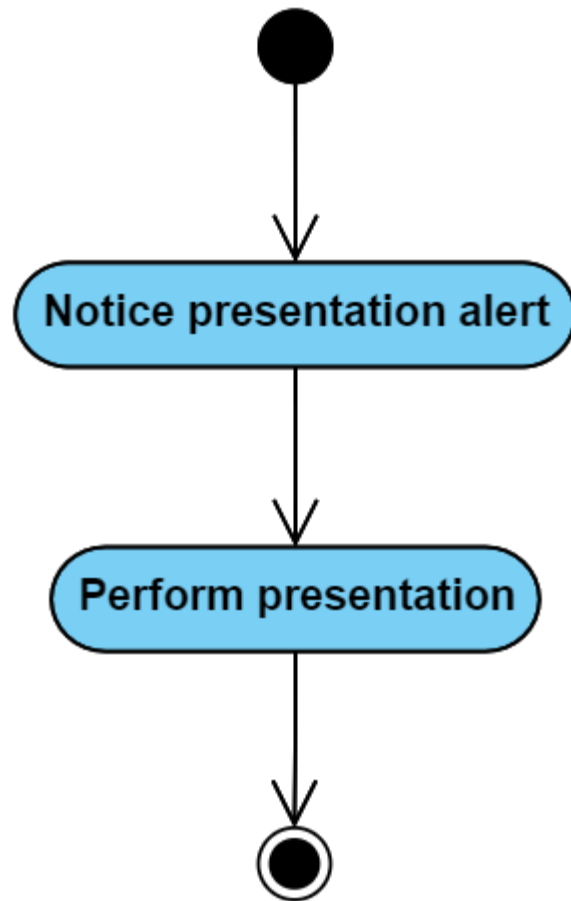


Figure 3.19: Activity Diagram for 'Perform presentation'

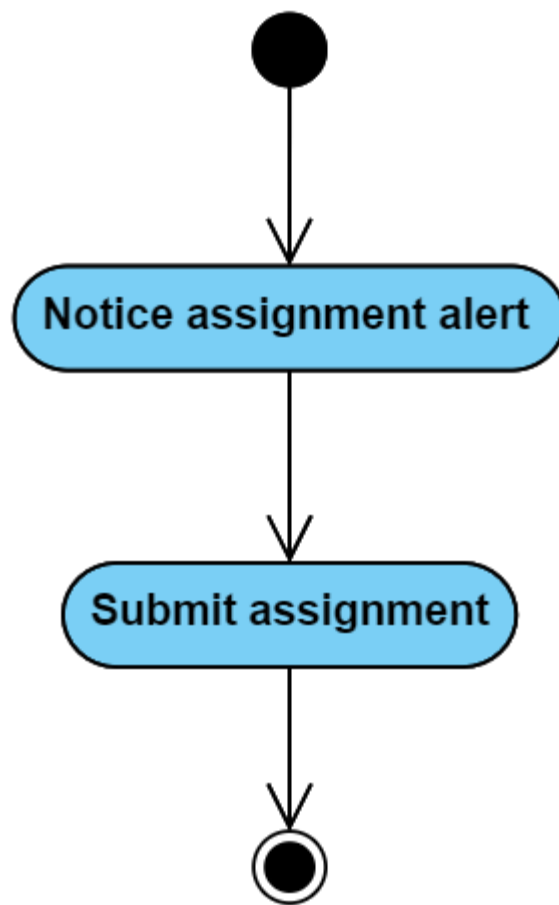


Figure 3.20: Activity Diagram for 'Submit assignment'

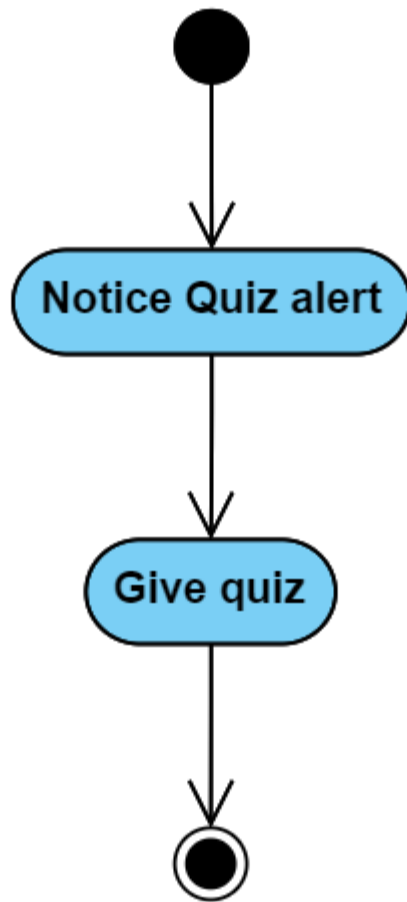


Figure 3.21: Activity Diagram for ‘Give quiz’

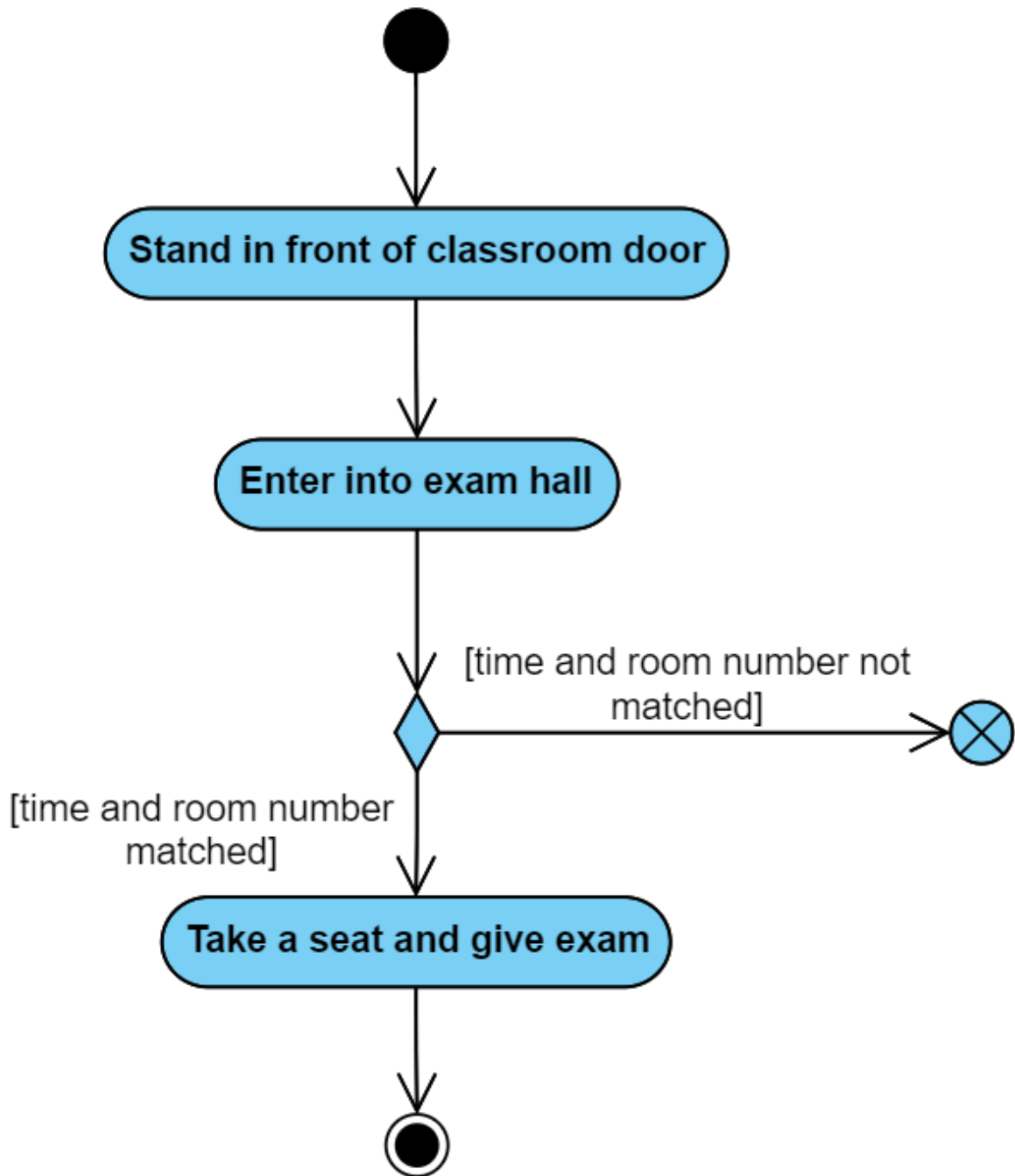


Figure 3.22: Activity Diagram for 'Give mid/final exam'

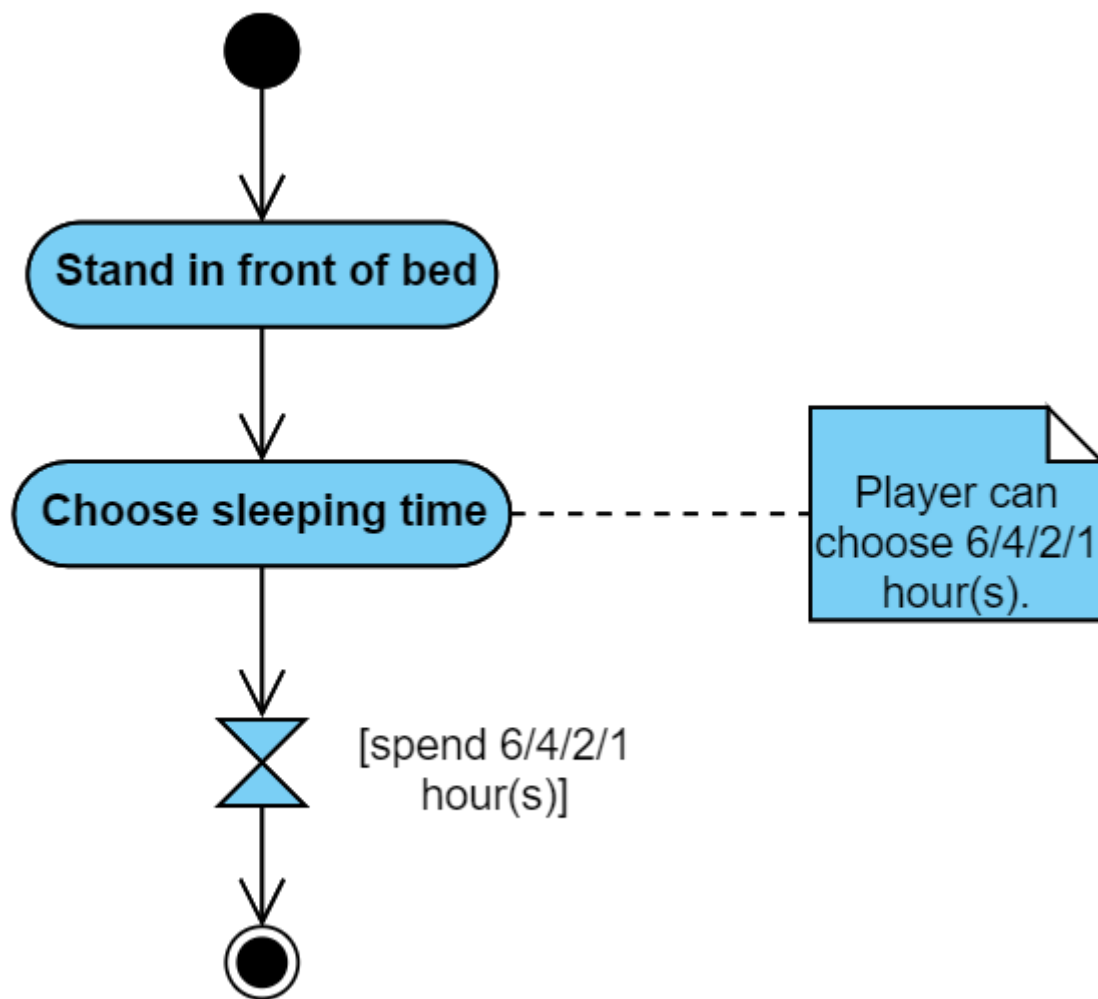


Figure 3.23: Activity Diagram for 'Go to sleep'

3.4 Sequence Diagram

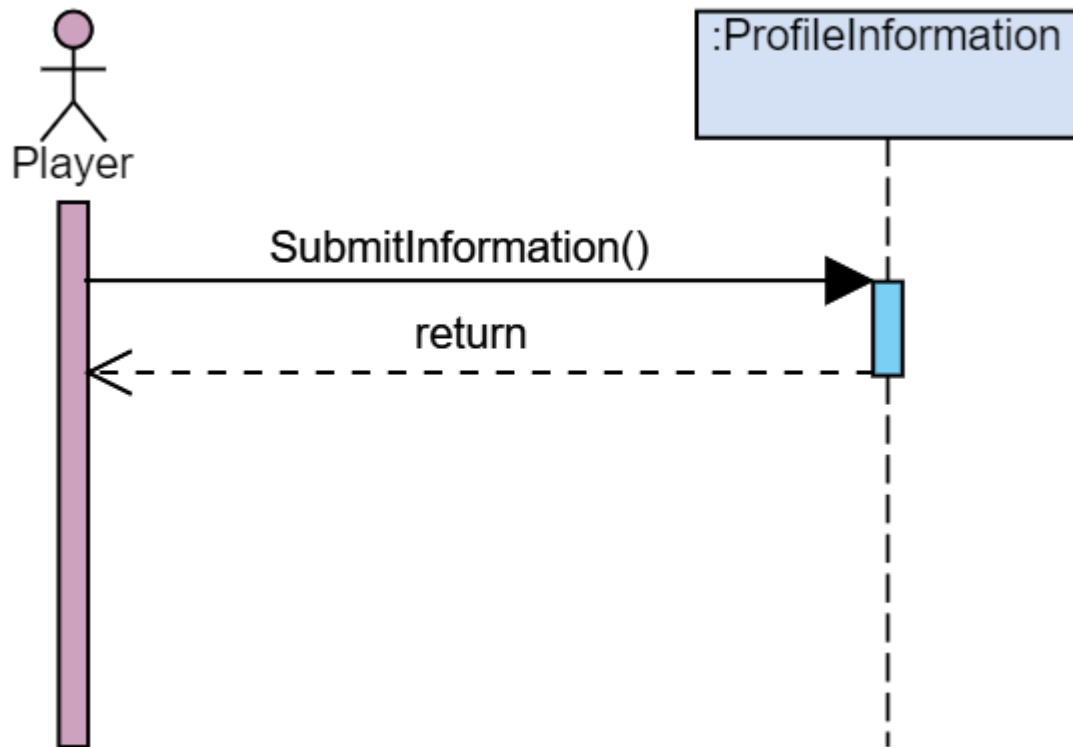


Figure 3.24: Sequence Diagram for 'Enter player's name'

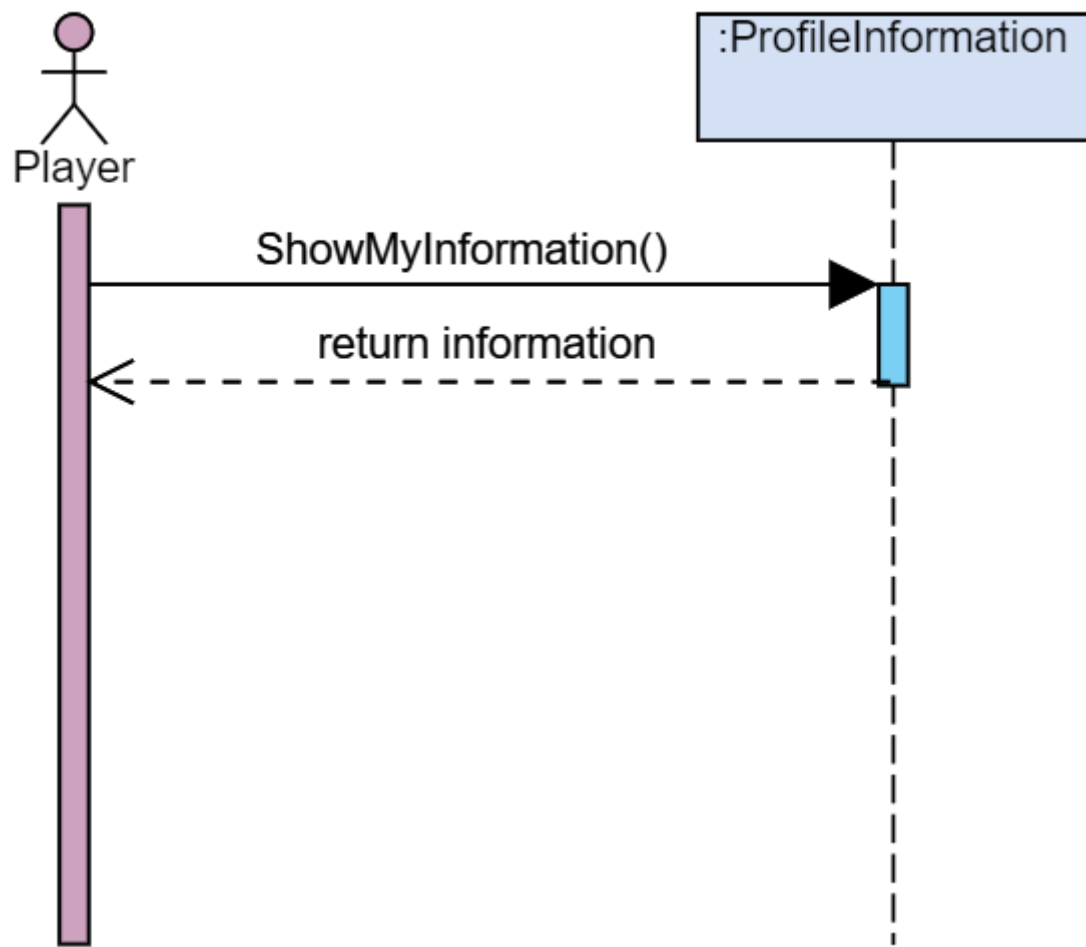


Figure 3.25: Sequence Diagram for 'Check player's information'

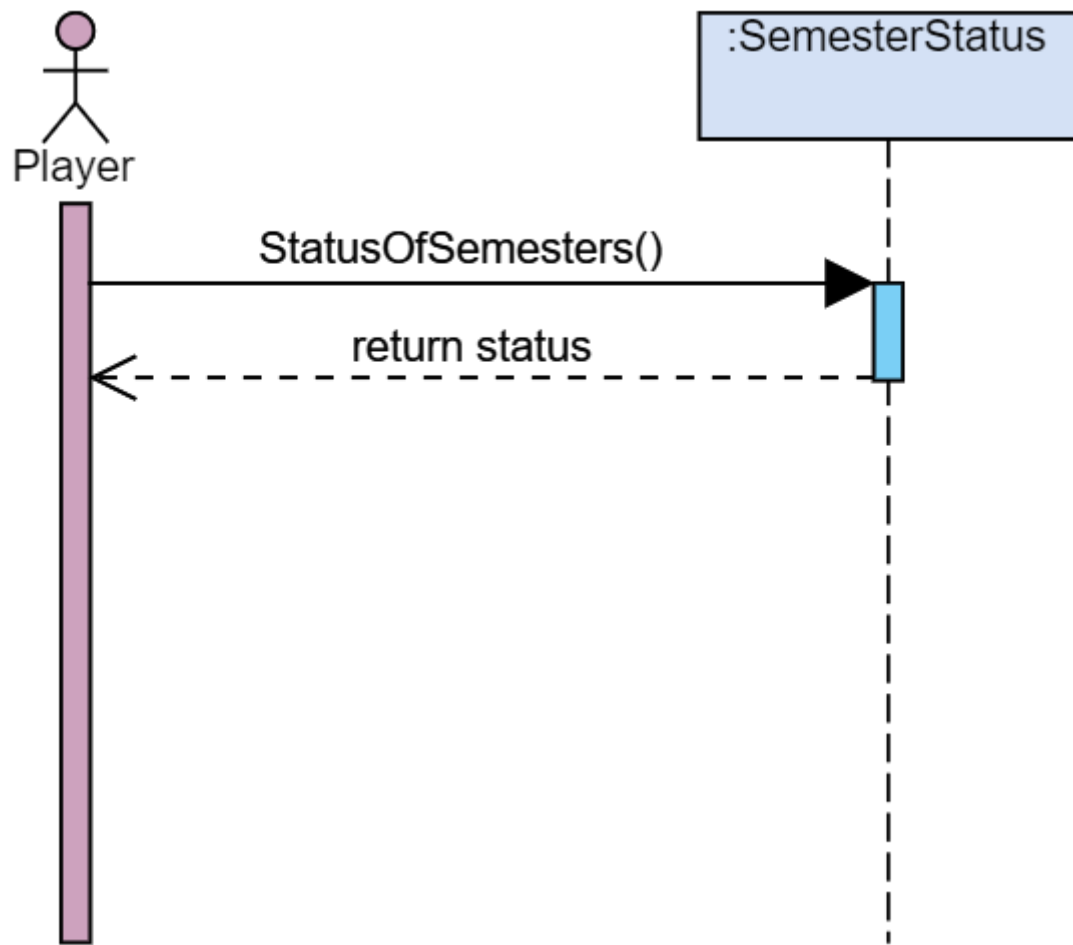


Figure 3.26: Sequence Diagram for 'View semester status'

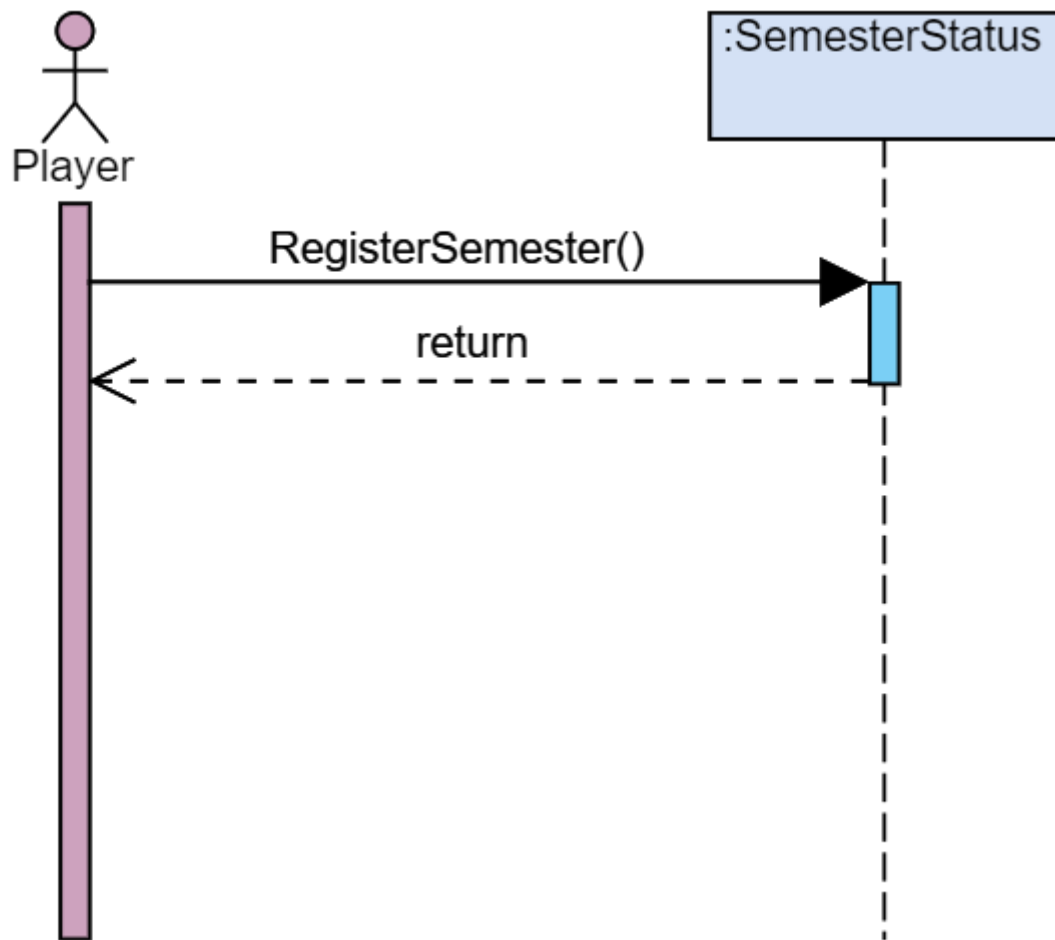


Figure 3.27: Sequence Diagram for 'Register semester'

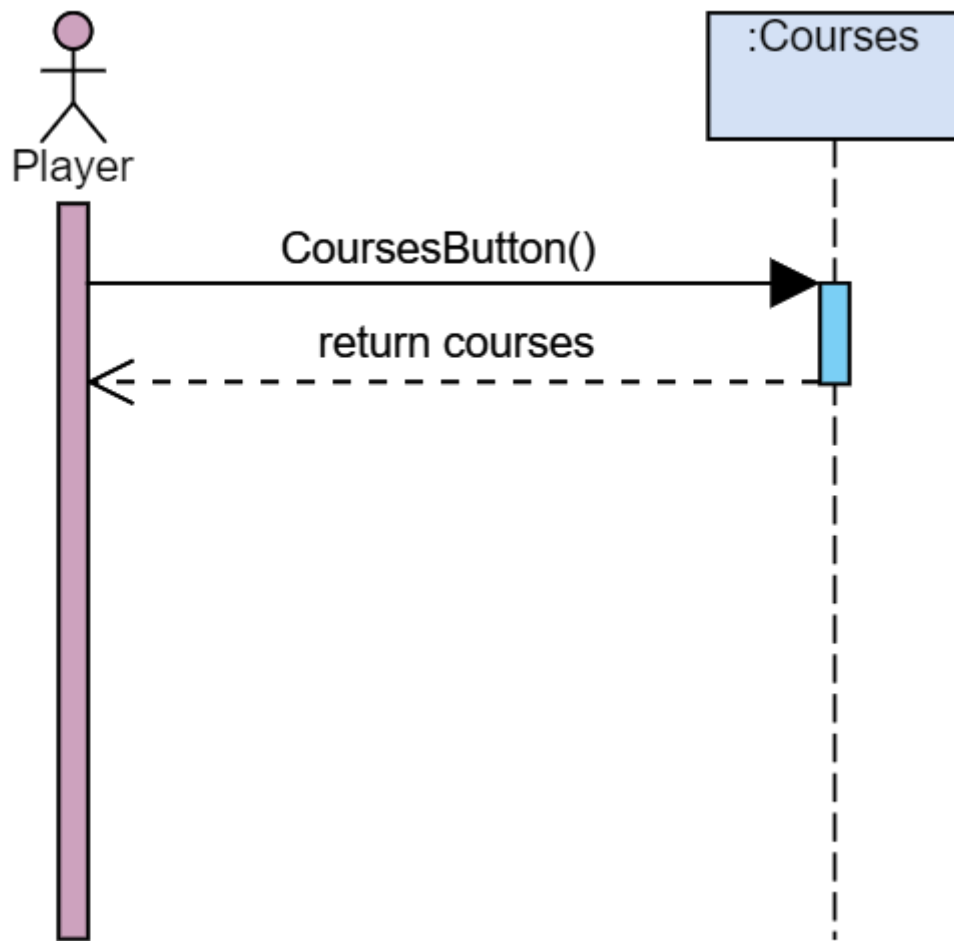


Figure 3.28: Sequence Diagram for ‘View semester courses’

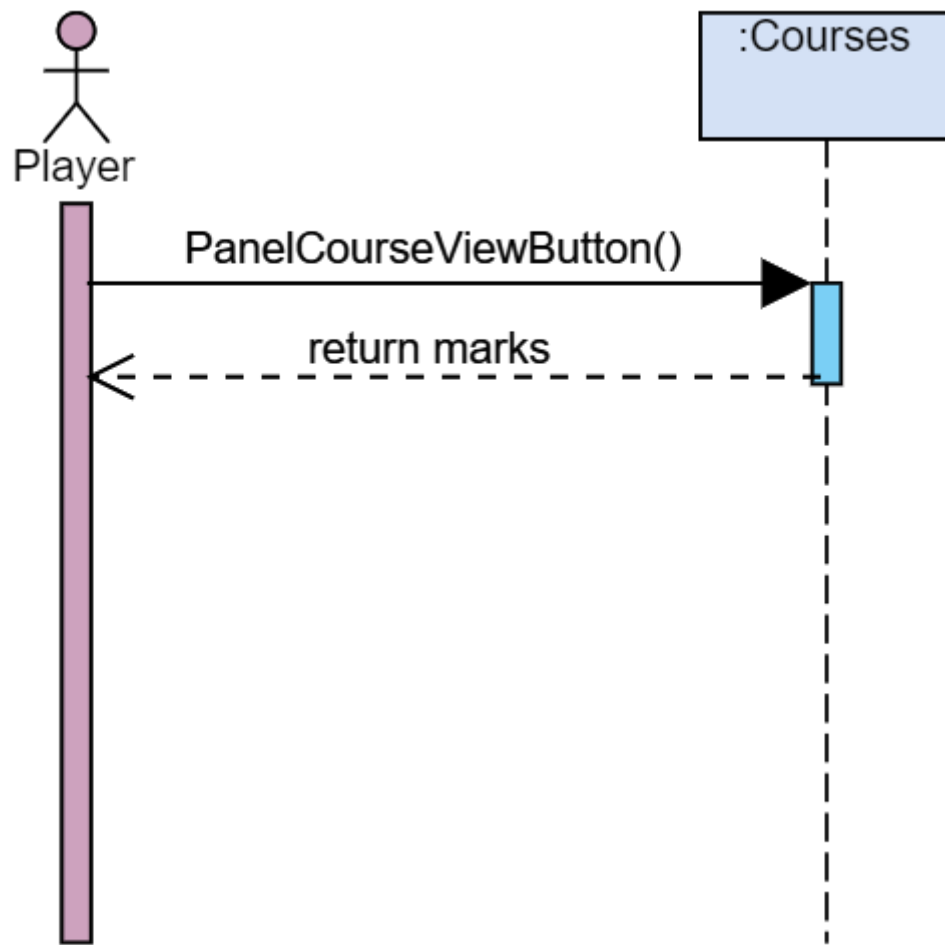


Figure 3.29: Sequence Diagram for 'View course marks'

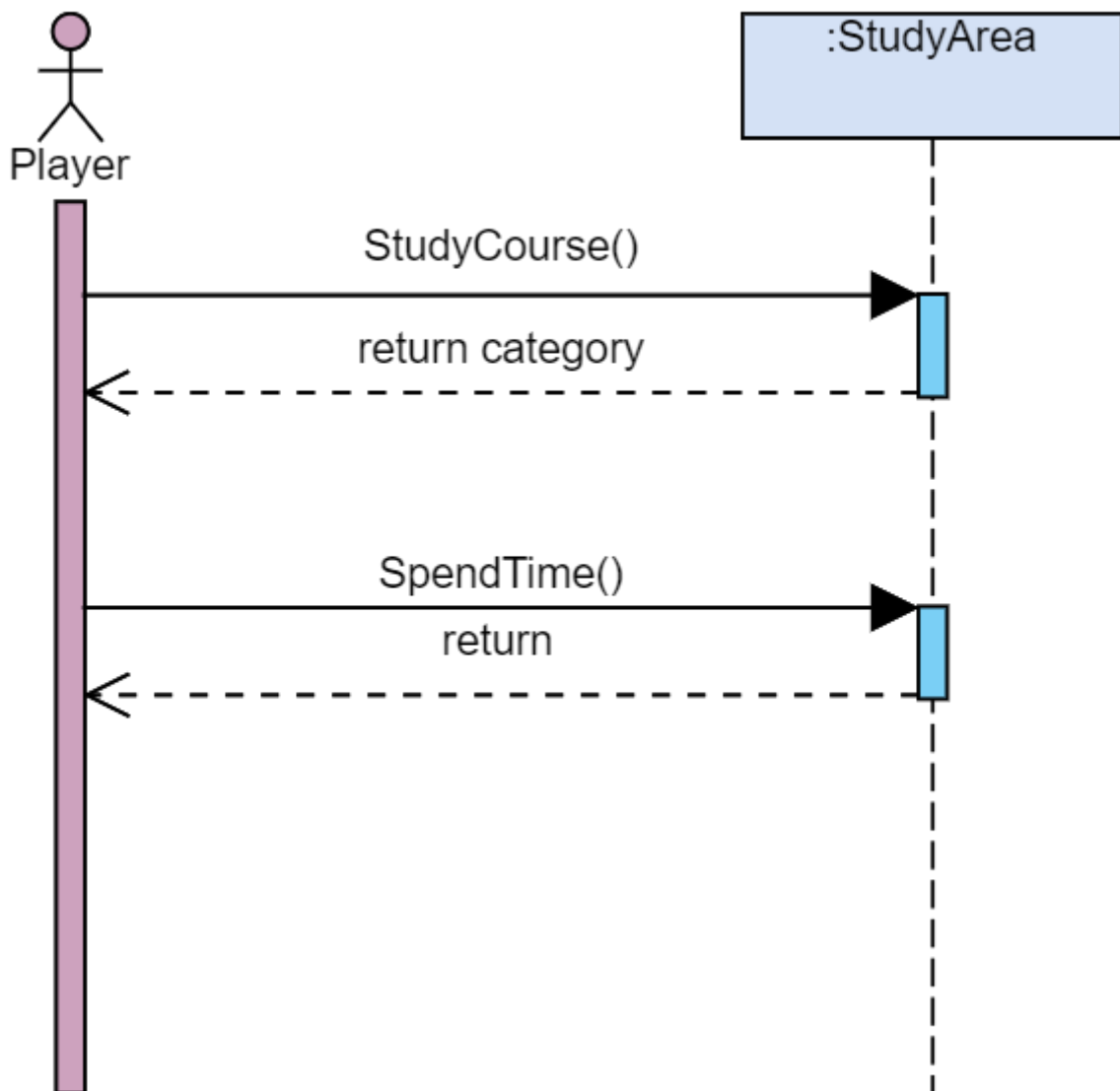


Figure 3.30: Sequence Diagram for 'Study courses'

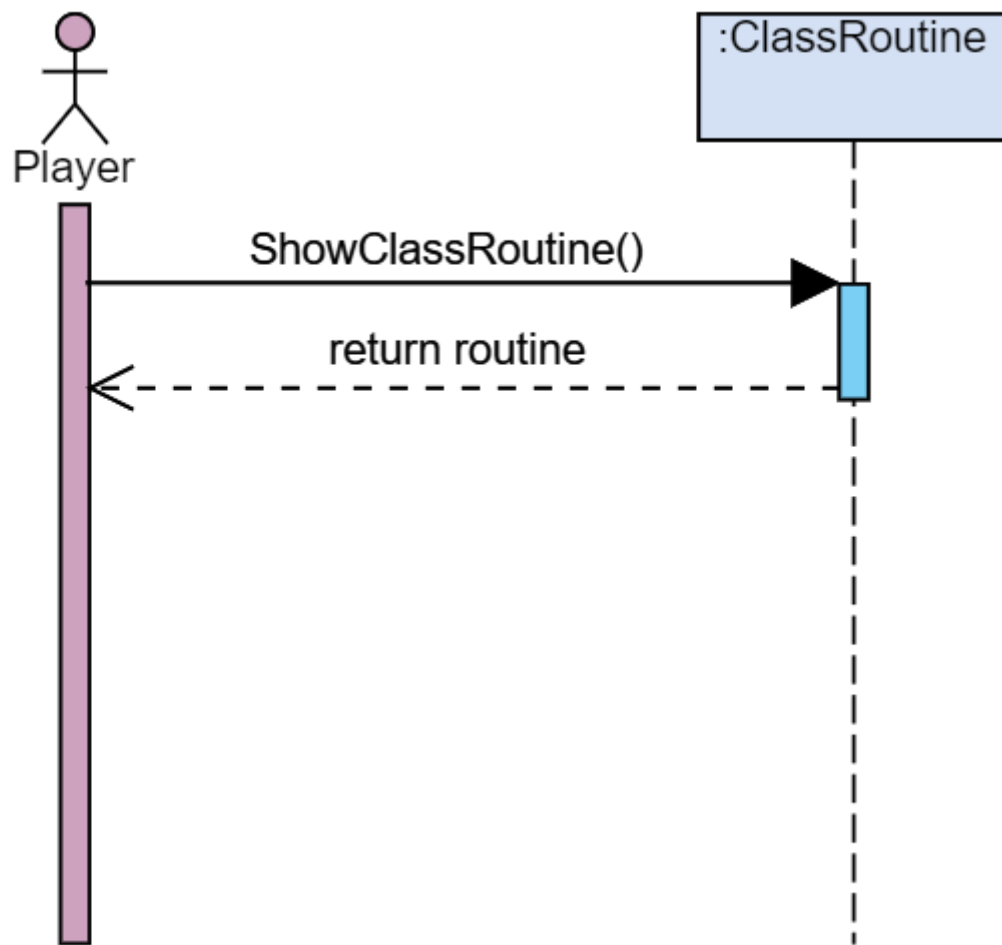


Figure 3.31: Sequence Diagram for ‘View class routine’

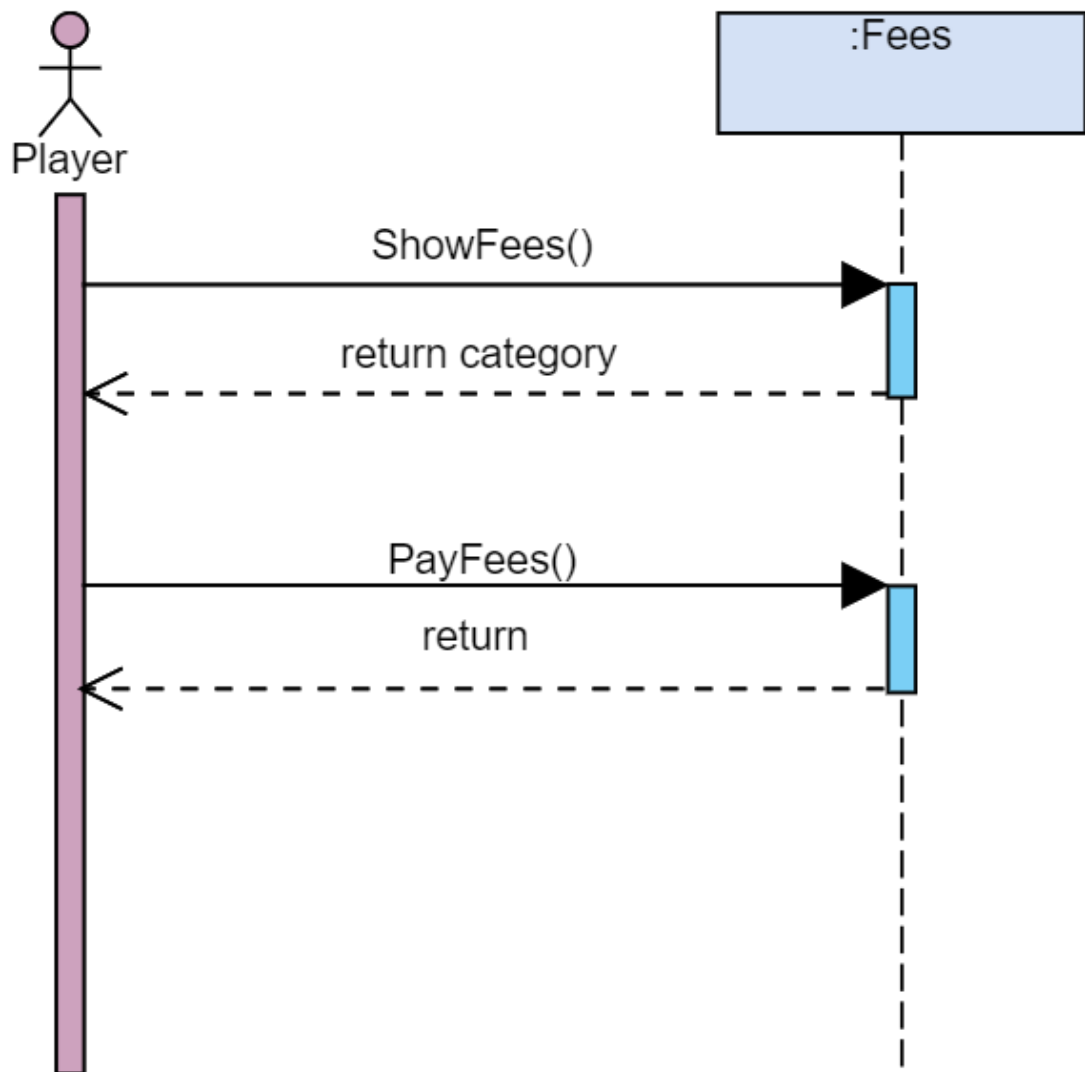


Figure 3.32: Sequence Diagram for 'Pay fees'

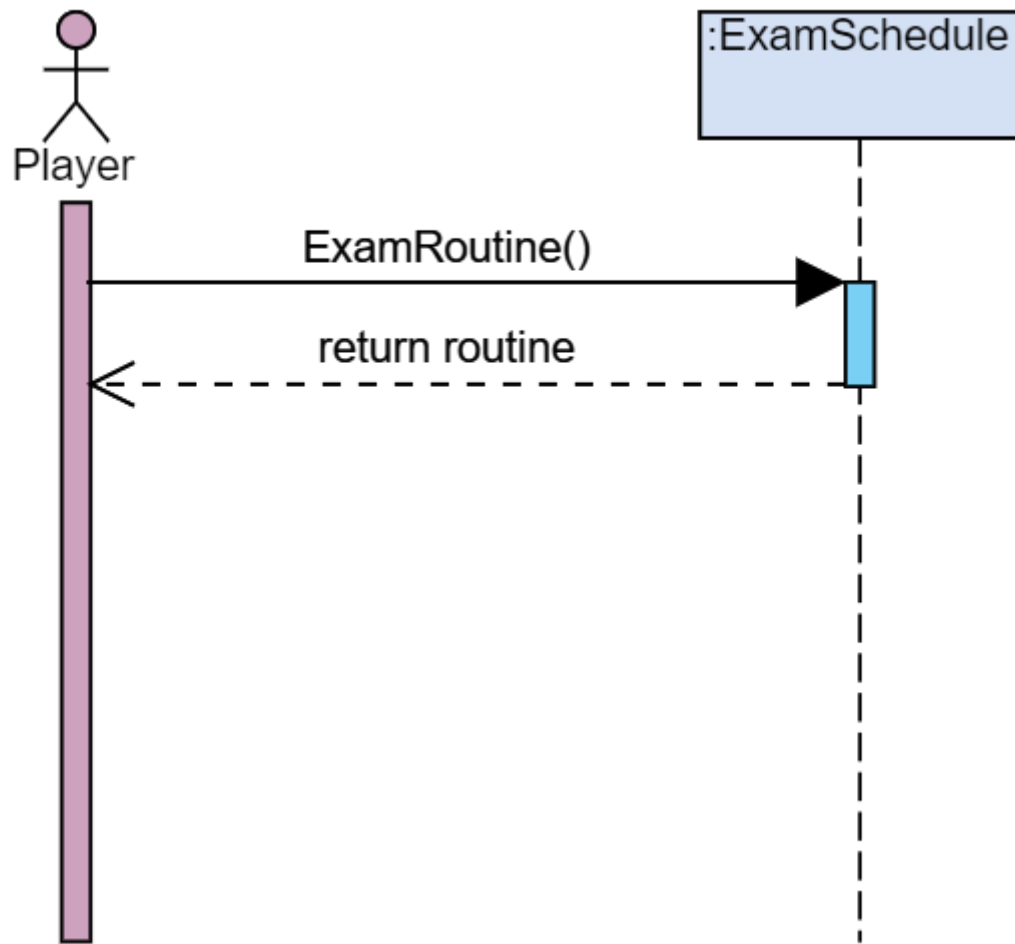


Figure 3.33: Sequence Diagram for 'View exam schedules'

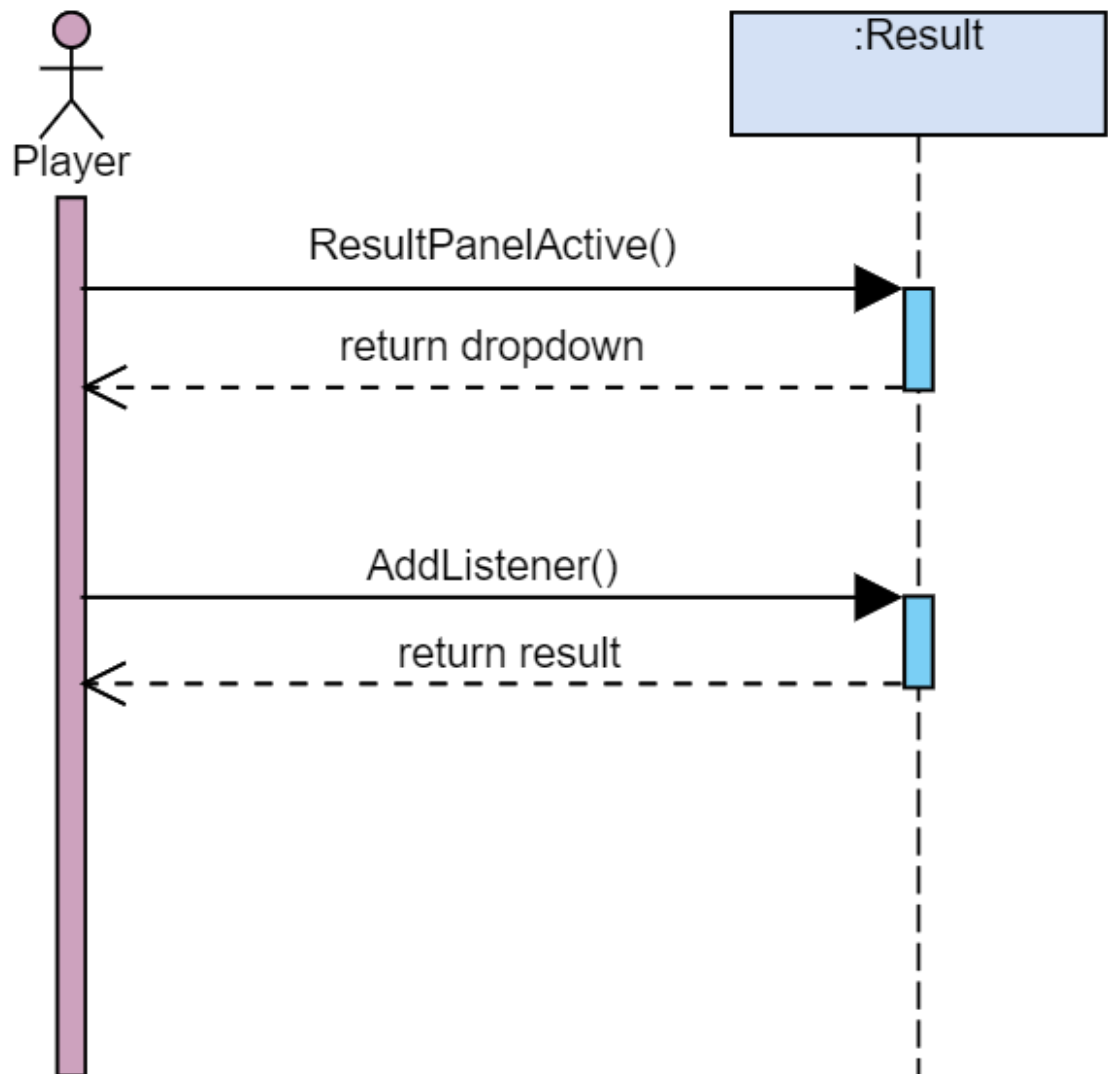


Figure 3.34: Sequence Diagram for 'View semester result'

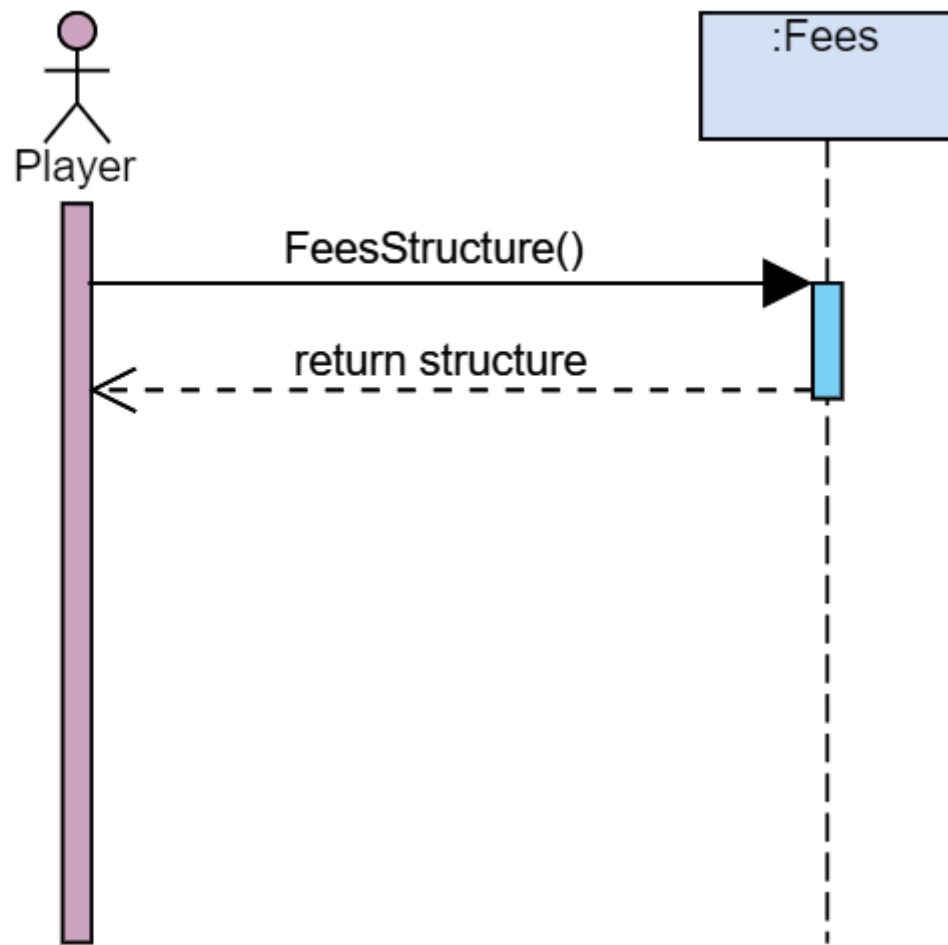


Figure 3.35: Sequence Diagram for 'View fees structure'

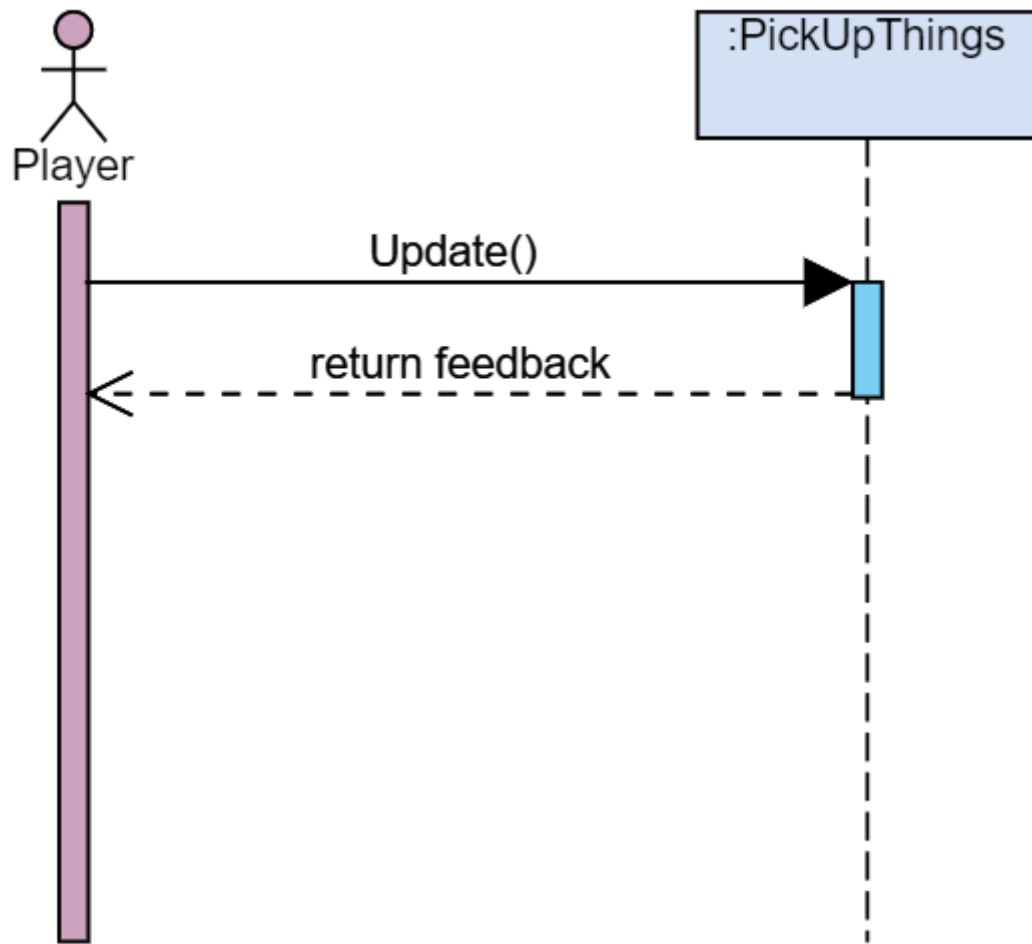


Figure 3.36: Sequence Diagram for 'Pick up id/bag'

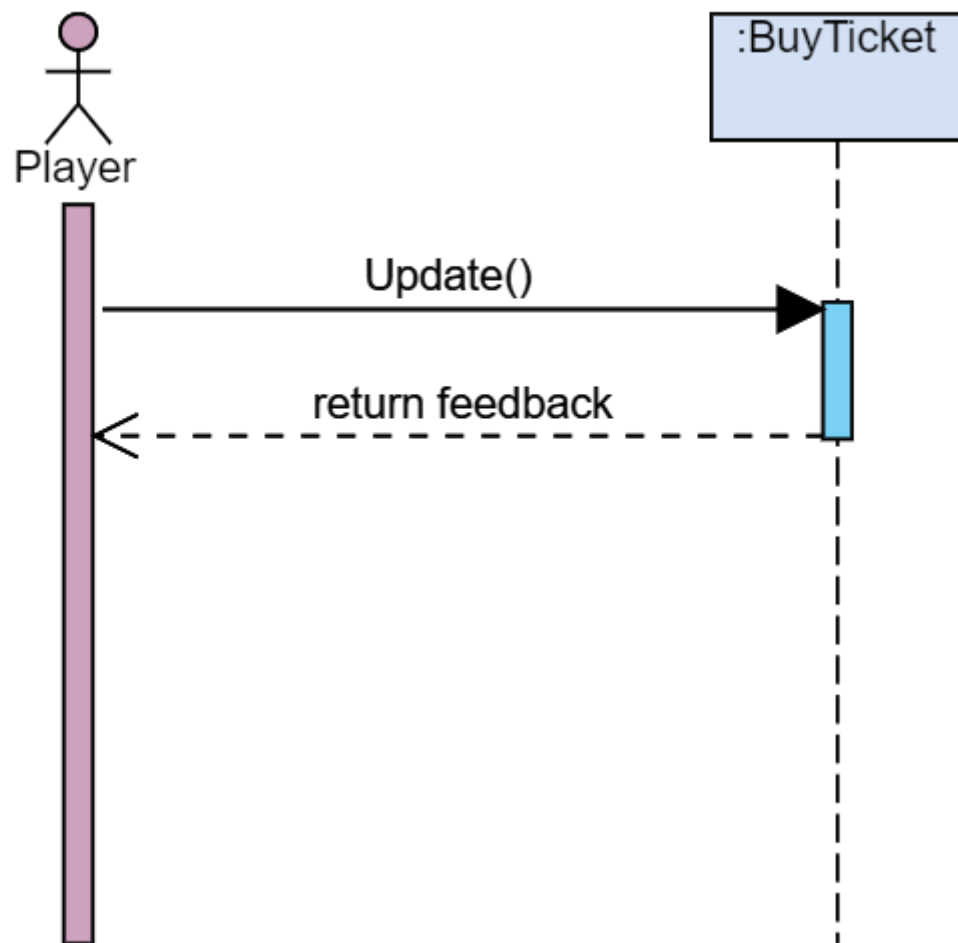


Figure 3.37: Sequence Diagram for 'Buy bus ticket'

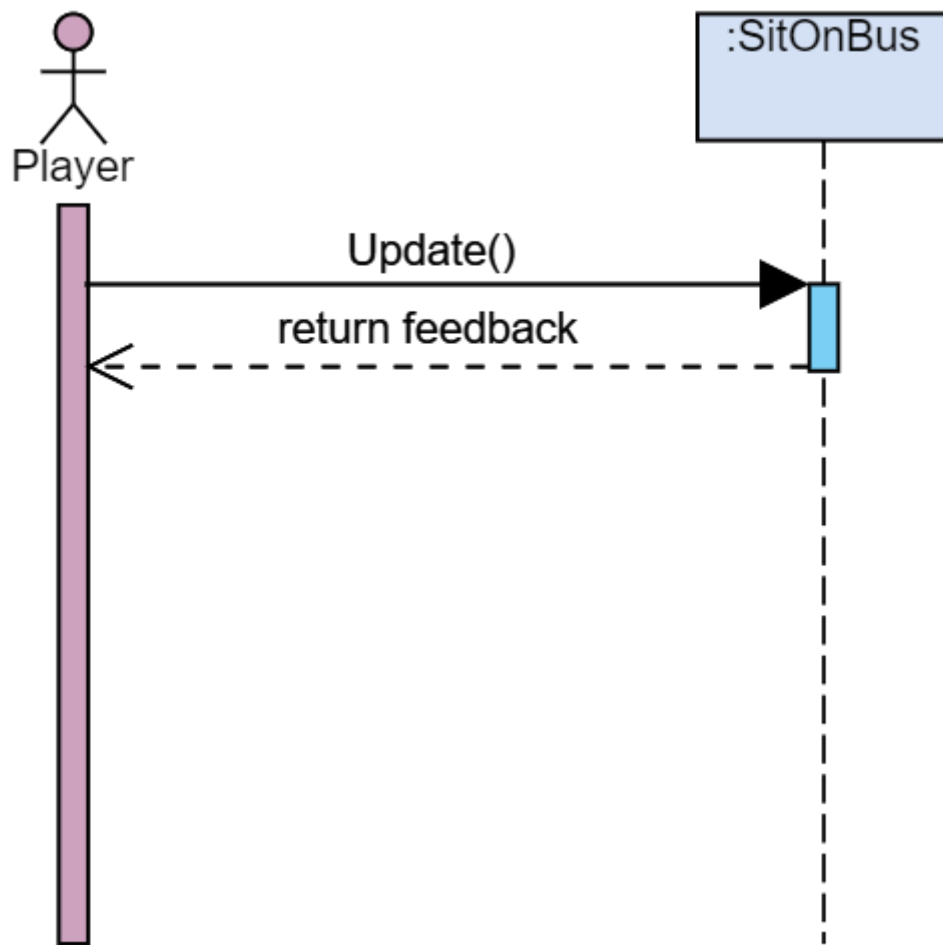


Figure 3.38: Sequence Diagram for 'Take a seat on bus'

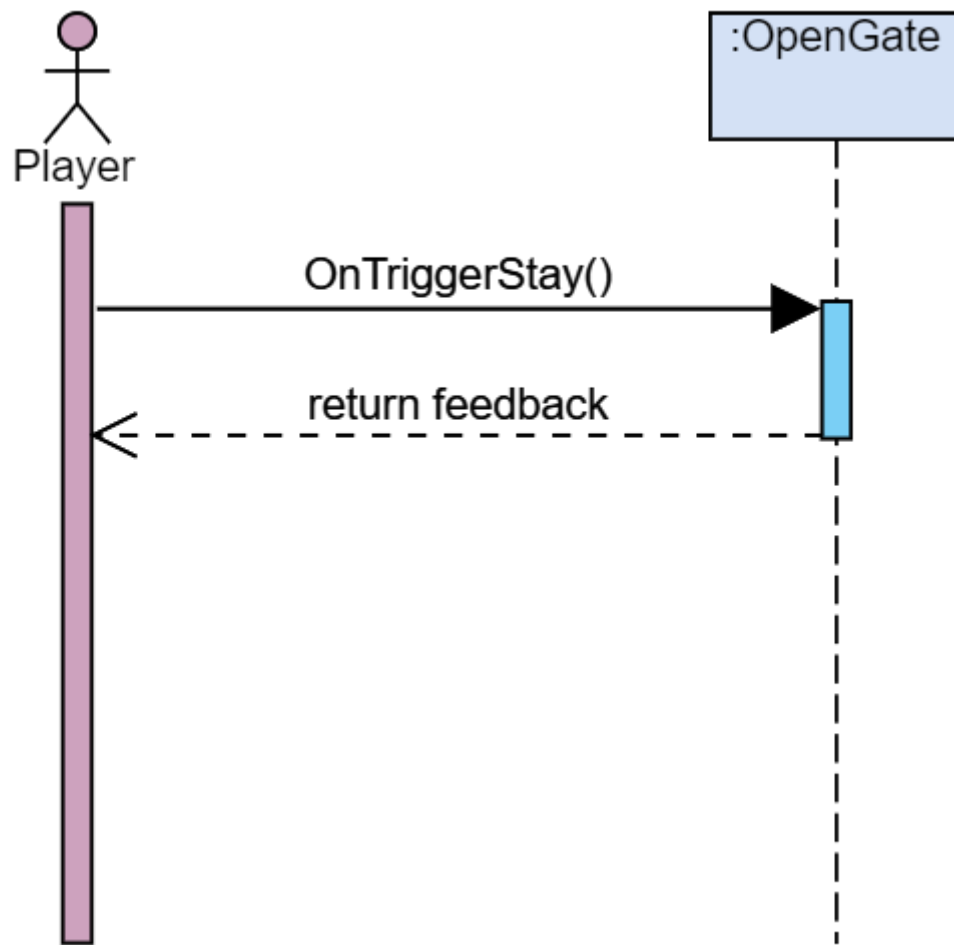


Figure 3.39: Sequence Diagram for 'Enter into university'

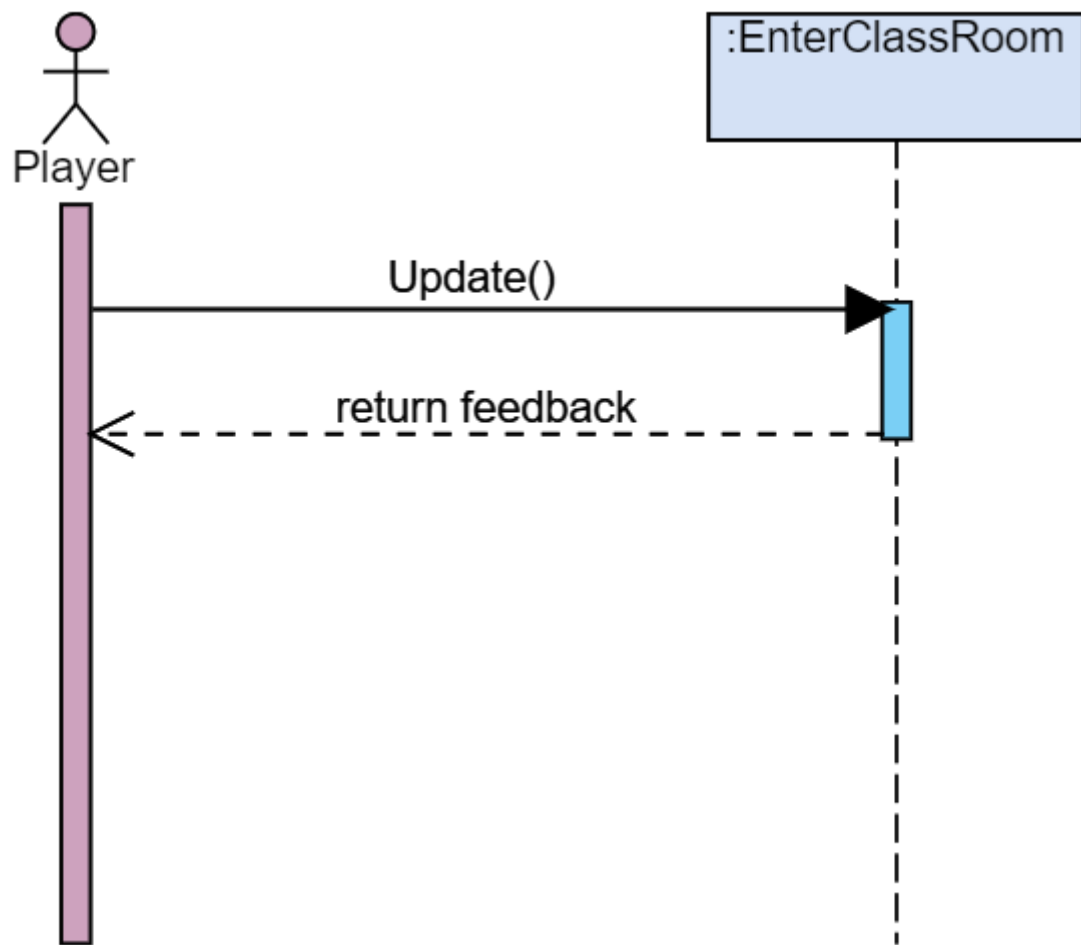


Figure 3.40: Sequence Diagram for 'Enter into classroom'

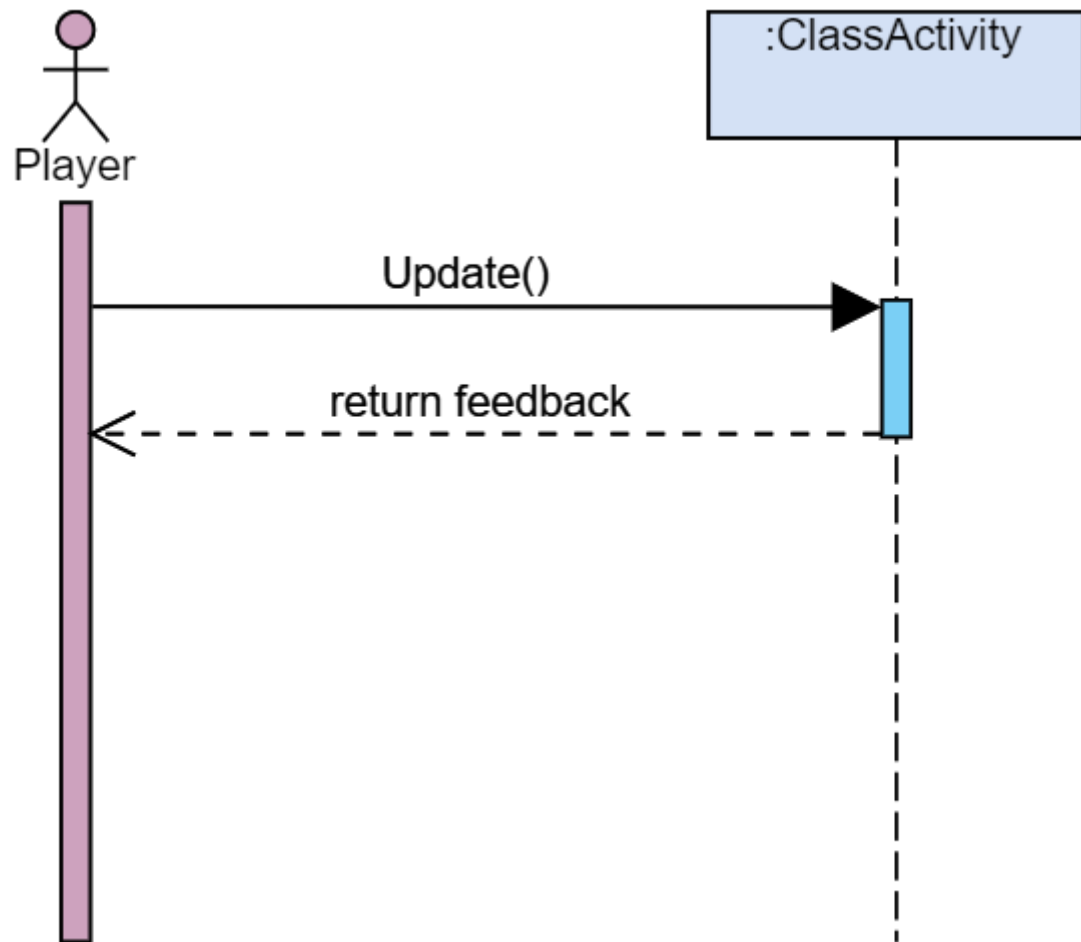


Figure 3.41: Sequence Diagram for 'Perform presentation'

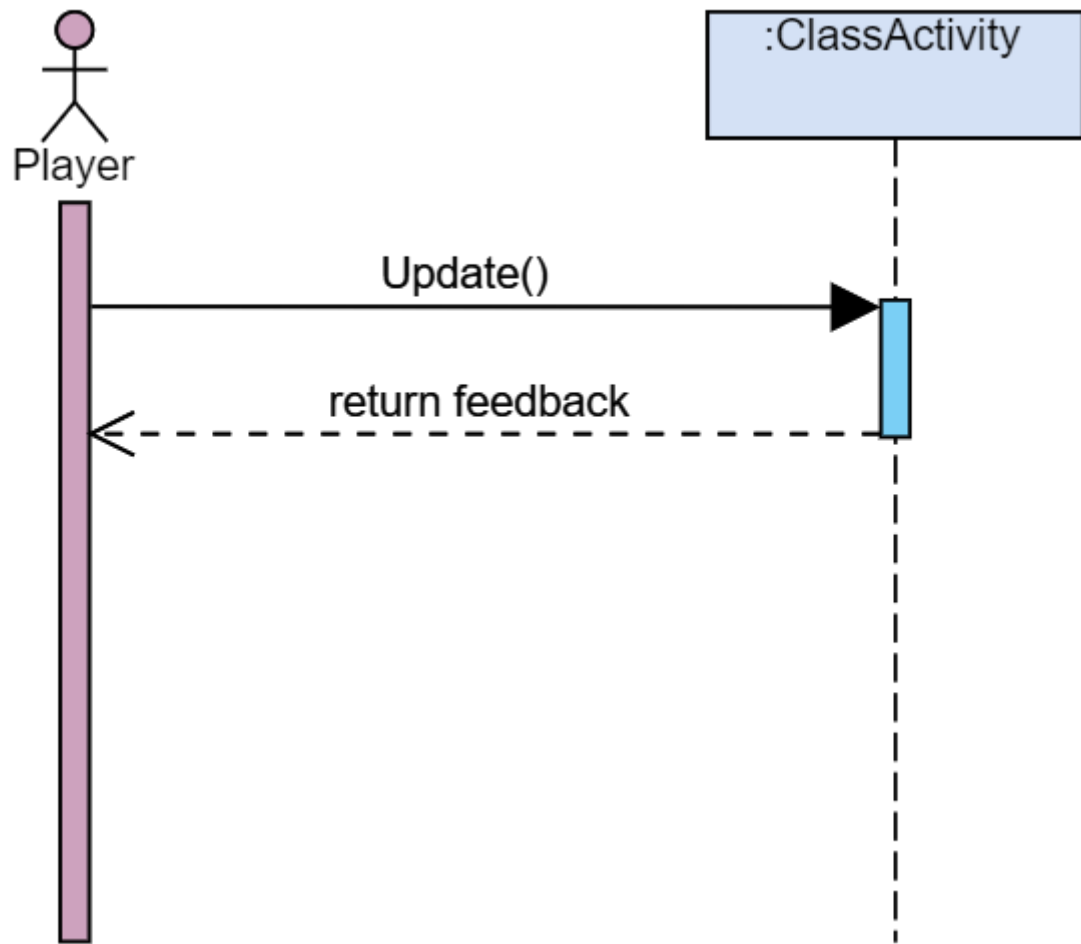


Figure 3.42: Sequence Diagram for 'Submit assignment'

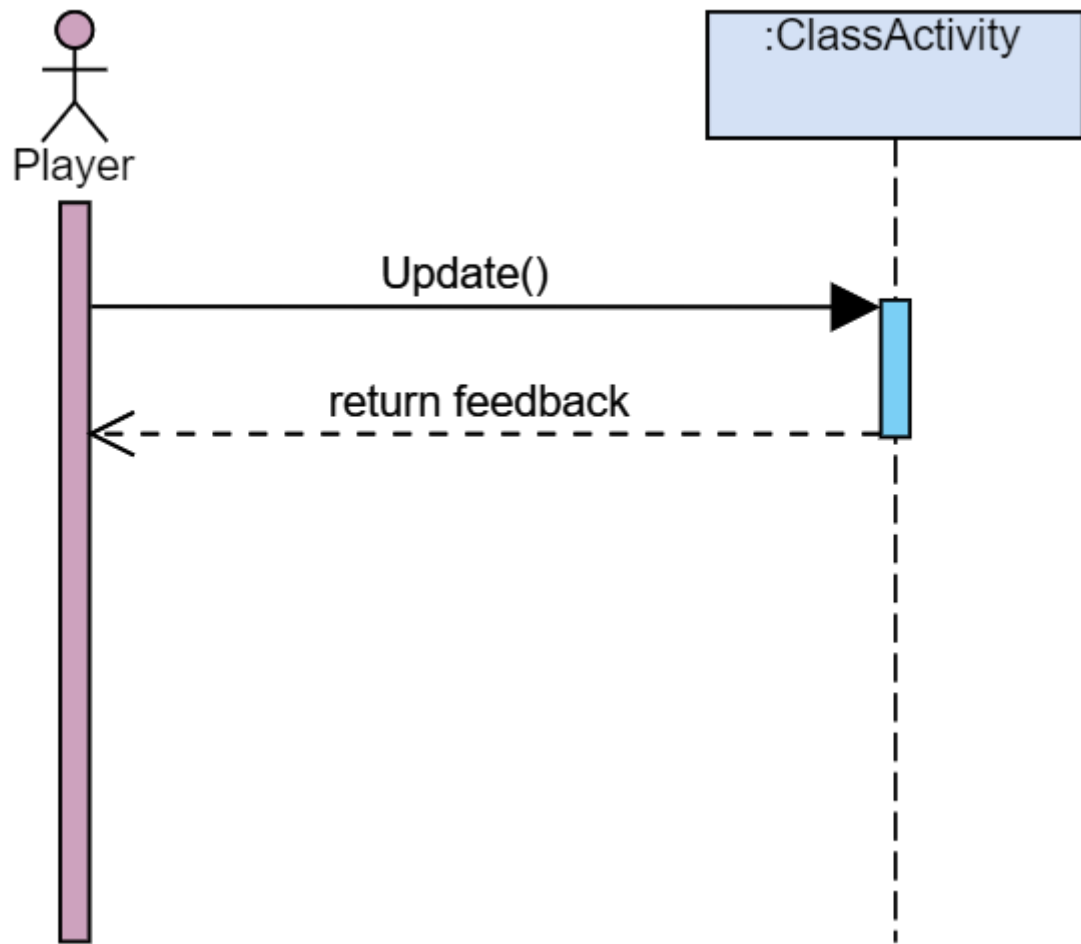


Figure 3.43: Sequence Diagram for 'Give quiz'

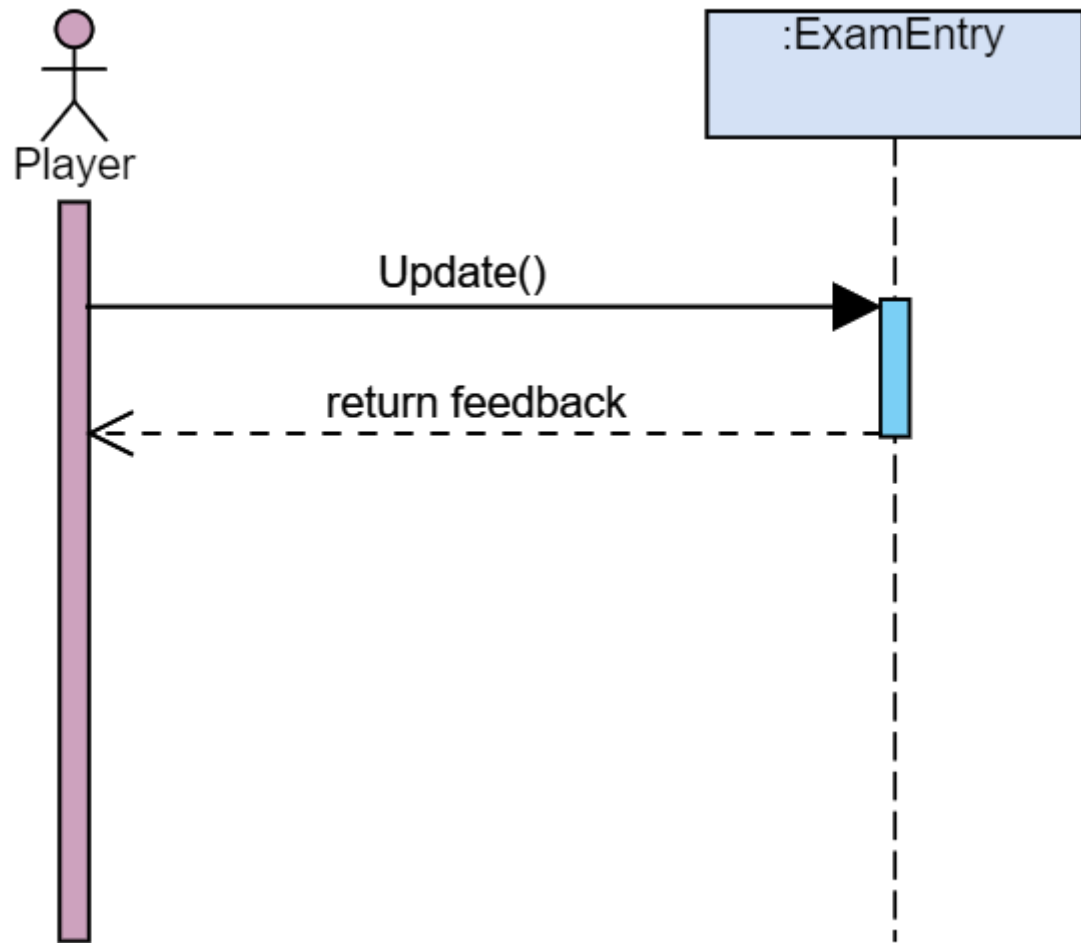


Figure 3.44: Sequence Diagram for ‘Give mid/final exam’

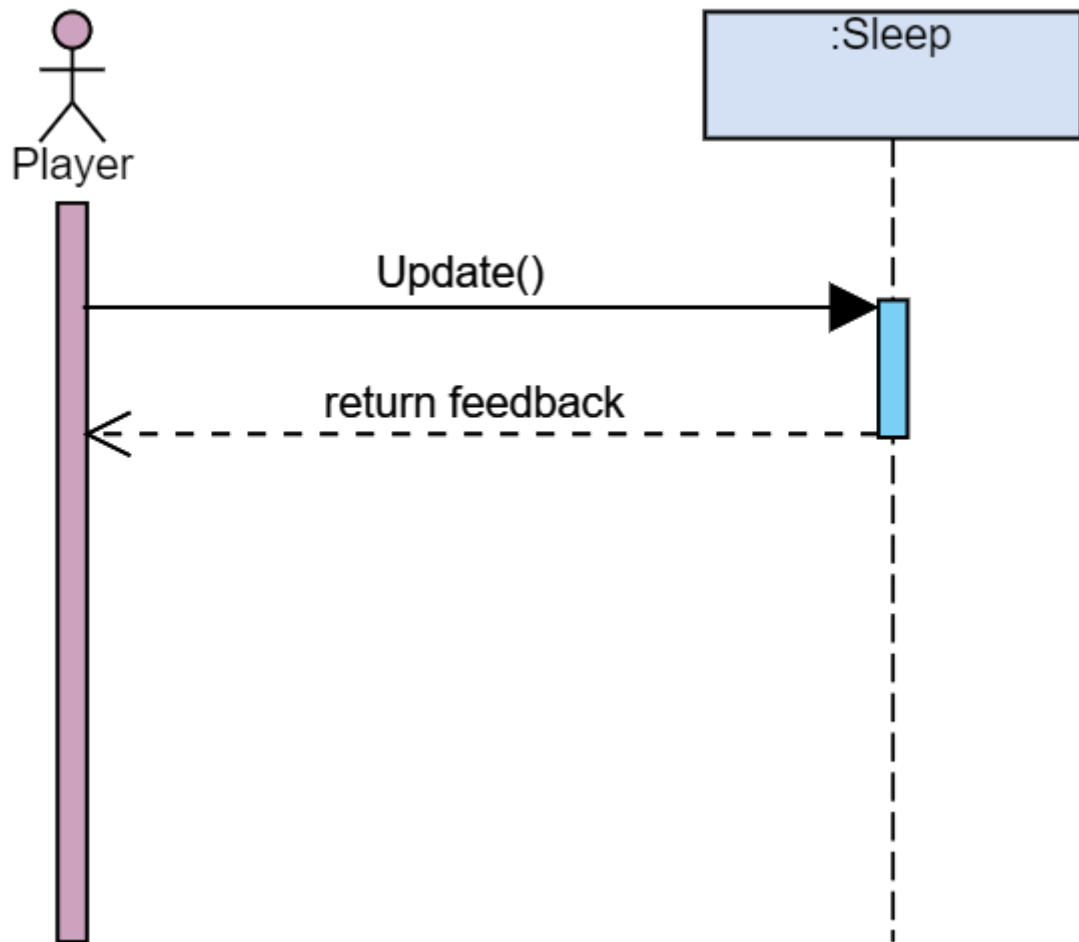


Figure 3.45: Sequence Diagram for 'Go to sleep'

CHAPTER 4

SYSTEM DESIGN SPECIFICATION

4.1 Game Scripts

There are more than forty scripts in the game. Here, I am providing some scripts.

StudyArea	TimeDate			
- dBConnection : DBConnection - courses : Courses - timeDate : TimeDate - PerSpendText : Text - MaxSpendDayText : Text - SpentTodayText : Text - TotalSpendText : Text - RecommendedTimeRegularStudyText : Text - SpentTimeRegularStudyText : Text - SpentTodayRegularStudyText : Text - SpentTodayAssignmentText : Text - RecommendedTimeAssignment1Text : Text - SpentTimeAssignment1Text : Text - RecommendedTimeAssignment2Text : Text - SpentTimeAssignment2Text : Text - RecommendedTimeAssignment3Text : Text - SpentTimeAssignment3Text : Text - SpentTodayPresentationText : Text - RecommendedTimePresentation1Text : Text - SpentTimePresentation1Text : Text - RecommendedTimePresentation2Text : Text - SpentTimePresentation2Text : Text - SpentTodayQuizText : Text - RecommendedTimeQuiz1Text : Text - SpentTimeQuiz1Text : Text - RecommendedTimeQuiz2Text : Text - SpentTimeQuiz2Text : Text - RecommendedTimeQuiz3Text : Text - SpentTimeQuiz3Text : Text - SpentTodayMidText : Text - RecommendedTimeMidText : Text - SpentTimeMidText : Text - SpentTodayFinalText : Text - RecommendedTimeFinalText : Text - SpentTimeFinalText : Text - panel1 : GameObject - panel2 : GameObject - panel3 : GameObject - panel4 : GameObject - panel5 : GameObject - panel6 : GameObject - panel7 : GameObject - Course1Text : Text - Course2Text : Text - Course3Text : Text - Course4Text : Text - Course5Text : Text - Course6Text : Text - Course7Text : Text + studyAreaPanel : GameObject + courseStudyIndividualPanel : GameObject + MaxSpendAlertPanel : GameObject + whichCourseClicked : int - sqlQuery1 : string - sqlQuery2 : string + semNo : int - FeesPanel : GameObject	+ timer : float - seconds : float + minutes : float + hours : float - timeShower : Text - dayCountShower : Text - dayShower : Text - timeSpeed : float - day : float - runningDay : float - dayTrack : int + dayName : string + stopTime : bool + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader + time : int + currentDay : int - runningSession : string + semesterDay : int - SessionText : Text - SemesterDayText : Text - entered : bool - MaxSpendAlertPanel : GameObject - SpentTodayText : Text			
	Awake() Update() OnApplicationQuit() TimeShower() IncreaseTime() ZeroSpentToday()			
	<table><tr><th>Sleep</th></tr><tr><td> - onceClick : bool - sleepPanel : GameObject - blackSleepImage : GameObject - timeDate : TimeDate - playerTouched : bool - pickedIDText : GameObject - bag : GameObject - id : GameObject - interaction : bool + overSleep : bool - presentDay : int </td></tr><tr><td> Update() OnTriggerEnter() sleepGap() OnTriggerExit() </td></tr></table>	Sleep	- onceClick : bool - sleepPanel : GameObject - blackSleepImage : GameObject - timeDate : TimeDate - playerTouched : bool - pickedIDText : GameObject - bag : GameObject - id : GameObject - interaction : bool + overSleep : bool - presentDay : int	Update() OnTriggerEnter() sleepGap() OnTriggerExit()
Sleep				
- onceClick : bool - sleepPanel : GameObject - blackSleepImage : GameObject - timeDate : TimeDate - playerTouched : bool - pickedIDText : GameObject - bag : GameObject - id : GameObject - interaction : bool + overSleep : bool - presentDay : int				
Update() OnTriggerEnter() sleepGap() OnTriggerExit()				

ThikanaBusTimeRoutine1
+ firstStation : bool + secondStation : bool + secondStation : bool + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - BusInterval()

ThikanaBusTimeRoutine2
+ firstStation : bool + secondStation : bool + secondStation : bool + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - BusInterval()

SitOnBus
- enterBusPanel : GameObject - busSeatTracker : Transform - player : GameObject - playerCamera : GameObject - counter : int + thikanaSit1 : bool + thikanaSit2 : bool + thikanaSit3 : bool + thikanaSit4 : bool + thikanaPassengersPoint1 : bool + thikanaPassengersPoint2 : bool + thikanaPassengersPoint3 : bool + thikanaPassengersPoint4 : bool + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - playerTouched : bool
- Start() - OnTriggerEnter() - Update() - OnTriggerExit() - PlayerMoveLookScriptsDisable()

ThikanaBusTimeRoutine3
+ firstStation : bool + secondStation : bool + secondStation : bool + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - BusInterval()

ThikanaBusTimeRoutine4
+ firstStation : bool + secondStation : bool + secondStation : bool + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - BusInterval()

ThikanaBusMove1
- _speed : float + busShouldMove : bool + sideChange : bool + returnWay : bool + bustype : string - thikanaBusStation1 : GameObject - thikanaReturnBusStation1 : GameObject + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - OnApplicationQuit()

ThikanaBusMove2
- _speed : float + busShouldMove : bool + sideChange : bool + returnWay : bool + bustype : string - thikanaBusStation1 : GameObject - thikanaReturnBusStation1 : GameObject + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - OnApplicationQuit()

PlayerMove
- horizontalInputName : string - verticalInputName : string - movementSpeed : float - runSpeed : float - walkSpeed : float - runBuildUpSpeed : float - runKey : KeyCode - slopeForceRayLength : float - jumpFallOff : AnimationCurve - jumpMultiplier : float - jumpKey : KeyCode - isJumping : bool - controller : CharacterController
- Awake() - Update() - PlayerMovement() - SetMovementSpeed() - OnSlope() - JumpInput() - JumpEvent()

ThikanaBusMove3
- _speed : float + busShouldMove : bool + sideChange : bool + returnWay : bool + bustype : string - thikanaBusStation1 : GameObject - thikanaReturnBusStation1 : GameObject + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - OnApplicationQuit()

ThikanaBusMove4
- _speed : float + busShouldMove : bool + sideChange : bool + returnWay : bool + bustype : string - thikanaBusStation1 : GameObject - thikanaReturnBusStation1 : GameObject + dBConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader
- Start() - Update() - OnApplicationQuit()

PlayerLook
- mouseXInputName : string - mouseYInputName : string + mouseSensitivity : float - playerBody : Transform - xAxisClamp : float
- Awake() - Update() - CameraRotation() - ClampXAxisRotationToValue()

ProfileInformation
+ dbConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - yourNameString : string - varsityNameString : string - departmentNameString : string - idNumberString : string - status : string - yourNameText : Text - profileNameText : Text - showNameText : Text - showVarsityNameText : Text - showDepartmentNameText : Text - showIdNumberText : Text - entryPanel : GameObject - player : GameObject - playerCamera : GameObject - informationProfilePanel : GameObject - semestersPanel : GameObject - semesterOverviewPanel : GameObject - semesterStatus : SemesterStatus + SubmitInformation() - ShowMyInformation() - LockCursor()

PickUpThings
+ items : List<string> - itemsPanel : GameObject - pickedIDText : GameObject - pickedBagText : GameObject - id : GameObject - bag : GameObject + pickedID : bool + pickedBag : bool + playerTouched : bool - StudyAreaButton : Button - semesterStatus : SemesterStatus - Update() - OnTriggerEnter() - OnTriggerExit()

MarksCalculation
+ dbConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - attendanceMarks : float - updateAttendanceMarks : bool - percentage : float - score : float - marks : float - assignmentAvg : float - presentationAvg : float - quizAvg : float + AttendanceMarksCalculation() + AssignmentMarksCalculation() + PresentationMarksCalculation() + QuizMarksCalculation() + MidAndFinalMarksCalculation() + MarksCalculationIndividual() - AssignmentMarksDistribution() - PresentationMarksDistribution() - QuizMarksDistribution() - MidMarksDistribution() - FinalMarksDistribution()

Semesters
+ courses : Courses + studyArea : StudyArea - semestersPanel : GameObject + whichSemesterClicked : int - semesterOverviewPanel : GameObject - SemesterCoursesPanel : GameObject - CourseViewPanel : GameObject + OpenSemester()

OpenGate
- entryGate : Animator - OnTriggerStay()

CursorControl
- player : GameObject - MenuPanel : GameObject + FreezePlayer() + FreePlayer()

Buy Ticket
- buyTicketPanel : GameObject - ticketBoughtPanel : GameObject - fare : float + hasTicket : bool + first : bool - panelOpen : bool - ticketPriceText : Text + ticketType : string - ticketTypeUpDown : string - transaction : Transaction + dbConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - Start() - OnTriggerEnter() - Update() - OnTriggerExit() - DisplayMessage()

DBConnection
+ courses : Courses - profileNameText : Text - PerSpendText : Text - MaxSpendDayText : Text - SpentTodayText : Text - TotalSpendText : Text - RecommendedTimeRegularStudyText : Text - SpentTimeRegularStudyText : Text - SpentTodayRegularStudyText : Text - SpentTodayAssignmentText : Text - RecommendedTimeAssignment1Text : Text - SpentTimeAssignment1Text : Text - RecommendedTimeAssignment2Text : Text - SpentTimeAssignment2Text : Text - RecommendedTimeAssignment3Text : Text - SpentTimeAssignment3Text : Text - SpentTodayPresentationText : Text - RecommendedTimePresentation1Text : Text - SpentTimePresentation1Text : Text - RecommendedTimePresentation2Text : Text - SpentTimePresentation2Text : Text - SpentTodayQuizText : Text - RecommendedTimeQuiz1Text : Text - SpentTimeQuiz1Text : Text - RecommendedTimeQuiz2Text : Text - SpentTimeQuiz2Text : Text - RecommendedTimeQuiz3Text : Text - SpentTimeQuiz3Text : Text - SpentTodayMidText : Text - RecommendedTimeMidText : Text - SpentTimeMidText : Text - SpentTodayFinalText : Text - RecommendedTimeFinalText : Text - SpentTimeFinalText : Text - AttendanceText : Text - PresentationAvgText : Text - Presentation1Text : Text - Presentation2Text : Text - AssignmentAvgText : Text - Assignment1Text : Text - Assignment2Text : Text - Assignment3Text : Text - QuizAvgText : Text - Quiz1Text : Text - Quiz2Text : Text - Quiz3Text : Text + MidText : Text - FinalText : Text - TotalText : Text - PointText : Text - GradeText : Text - Course1Text : Text - Code1Text : Text - Credit1Text : Text - Course2Text : Text - Code2Text : Text - Credit2Text : Text - Course3Text : Text - Code3Text : Text - Credit3Text : Text - Course4Text : Text - Code4Text : Text - Credit4Text : Text - Course5Text : Text - Code5Text : Text - Credit5Text : Text - Course6Text : Text - Code6Text : Text - Credit6Text : Text - Course7Text : Text - Code7Text : Text - Credit7Text : Text - Course1StudyAreaText : Text - Course2StudyAreaText : Text - Course3StudyAreaText : Text - Course4StudyAreaText : Text - Course5StudyAreaText : Text - Course6StudyAreaText : Text - Course7StudyAreaText : Text - perSpend : float - maxSpendDay : float - spentToday : float - totalSpend : float - spentTimeRegularStudy : float - spentTimeAssignment1 : float - spentTimeAssignment2 : float - spentTimeAssignment3 : float - spentTimePresentation1 : float - spentTimePresentation2 : float - spentTimeQuiz1 : float - spentTimeQuiz2 : float - spentTimeQuiz3 : float - spentTimeMid : float - spentTimeFinal : float - spentTodayRegularStudy : float - spentTodayAssignment : float - spentTodayPresentation : float - spentTodayQuiz : float - spentTodayFinal : float - semTotalCourses : int + id_card : string + path : string - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - entryPanel : GameObject - player : GameObject - playerCamera : GameObject - Start() + ReadDatabase() + UpdateDatabase() + InsertToDatabase()

Courses
+ studyArea : StudyArea - dbConnection : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - path : string - GENERAL : float - CORE : float - LAB : float - panel : GameObject - panel1 : GameObject - panel2 : GameObject - panel3 : GameObject - panel4 : GameObject - panel5 : GameObject - panel6 : GameObject - panel7 : GameObject - courseViewPanel : GameObject - SemesterCoursesPanel : GameObject - classRoutinePanel : GameObject - FeesPanel : GameObject - examPanel : GameObject + semesterTotalCourses : int + DetermineSemesterCourses() + AppearCoursesPanel() - HideCoursePanel() + PanelCourseViewButton() + CoursesButton() + CourseCode() + CourseName() + CourseCredit() + CalculateFees()

BusStartPoint
- thikanaBusStation2 : Transform - thikanaBus1 : GameObject - thikanaBus2 : GameObject - thikanaBus3 : GameObject - thikanaBus4 : GameObject + touched1 : bool + touched2 : bool + touched3 : bool + touched4 : bool + dbConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - Start() - OnTriggerEnter()

BusEndPoint
- player : GameObject - playerCamera : GameObject - getDownFromBusPoint : Transform - downFromBus : bool - thikanaBus1 : Transform - thikanaBus2 : Transform - thikanaBus3 : Transform - thikanaBus4 : Transform - thikanaBusStation3 : Transform + touched1 : bool + touched2 : bool + touched3 : bool + touched4 : bool + dbConnectionObject : DBConnection - dbConnection : IDbConnection - dbCommand : IDbCommand - reader : IDataReader - Start() - Update() - OnTriggerEnter() - PassengerGetDownFromBus()

BusFinalEndPoint
- thikanaReturnBusStation1 : GameObject - thikanaBus1 : GameObject - thikanaBus2 : GameObject - thikanaBus3 : GameObject - thikanaBus4 : GameObject + dbConnectionObject : DBConnection - OnTriggerEnter()

BusFinalEndPointReturn
- thikanaBusStation1 : Transform - thikanaBus1 : GameObject - thikanaBus2 : GameObject - thikanaBus3 : GameObject - thikanaBus4 : GameObject + dbConnectionObject : DBConnection - OnTriggerEnter()

Fees
+ dBConnectionObject : DBConnection - dbConnection: IDbConnection - dbCommand : IDbCommand - reader : IDataReader - path : string - MidAmountText : Text - FinalAmountText : Text - MidPayButtonText : Text - FinalPayButtonText : Text - MidPayButton : Button - FinalPayButton : Button - FeesButton : Button - midAmount : float - finalAmount : float - FeesPanel : GameObject - transaction : Transaction - classRoutinePanel : GameObject - SemesterCoursesPanel : GameObject - courseViewPanel : GameObject - studyArea : StudyArea + PayFees() + ShowFees()

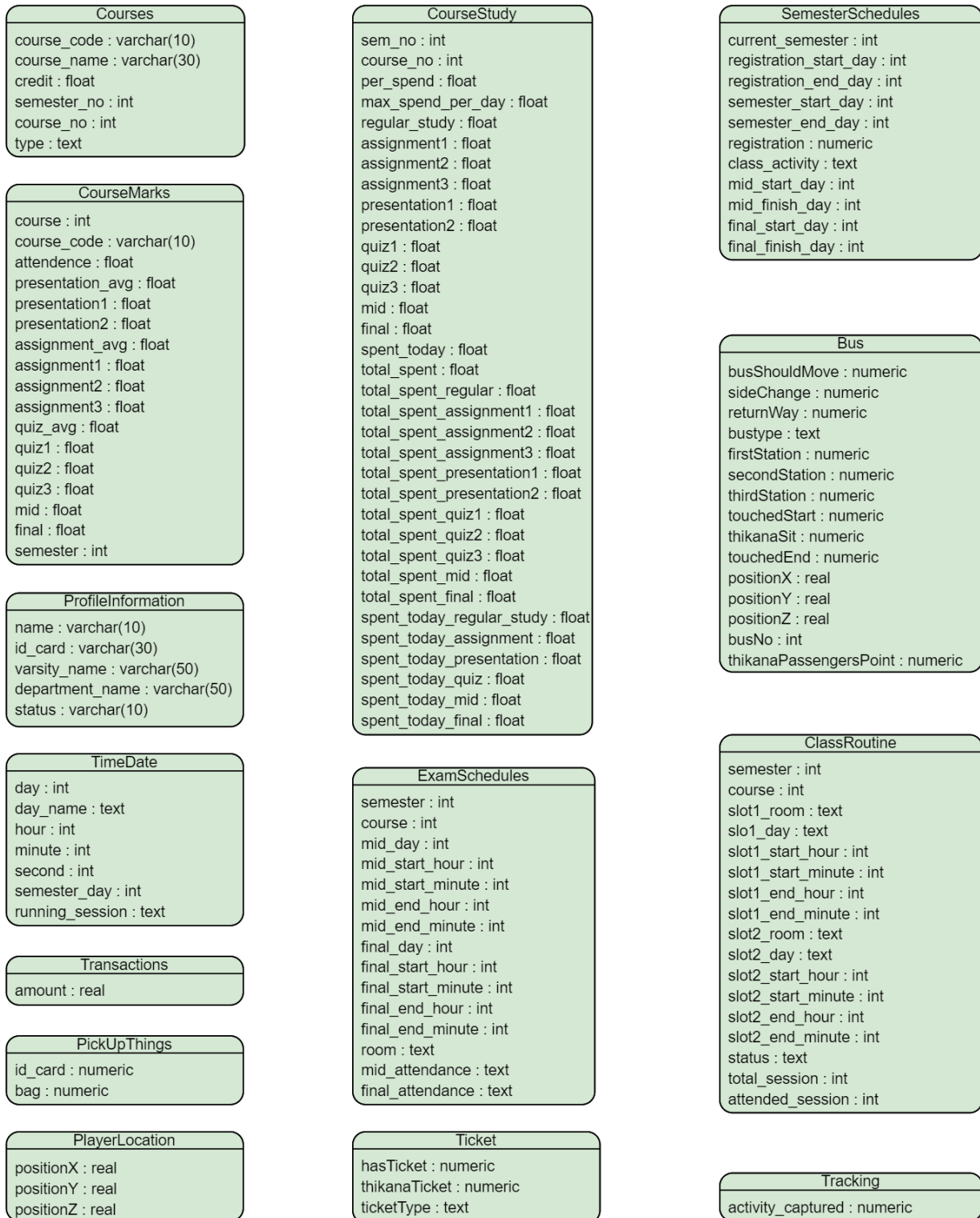
ClassActivities
+ dBConnectionObject : DBConnection - dbConnection: IDbConnection - dbCommand : IDbCommand - reader : IDataReader - TodayClassActivityPanel : GameObject - ActivityTypeText : Text - FToDoActivityText : Text - semester : int - course : int - activityAvailable : bool - activityCaptured : bool - marksCalculation : MarksCalculation - courses : Courses - semesterSchedules : SemesterSchedules - assignment1Day : int - assignment2Day : int - quiz1Day : int - presentation1Day : int - assignment3Day : int - quiz2Day : int - presentation2Day : int - quiz3Day : int - Update() - ActivitySchedule()

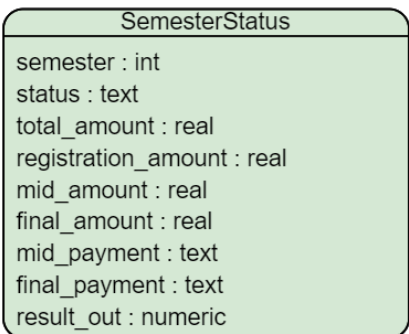
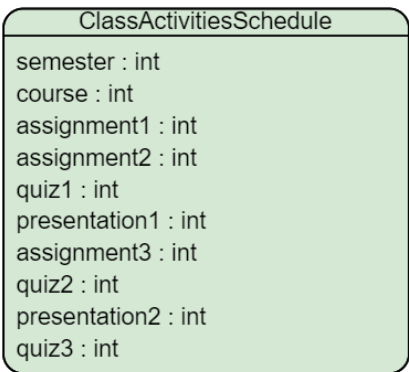
EnterHome
- interactionNoticePanel : GameObject - playerTouched : bool - entryPosition : GameObject - player : GameObject - cursorControl : CursorControl + OnTriggerEnter() + OnTriggerExit() + Update() + WaitFreeze()

ClassRoutine
- classRoutinePanel : GameObject - SemesterCoursesPanel : GameObject - courseViewPanel : GameObject - FeesPanel : GameObject - examPanel : GameObject - studyArea : StudyArea - Course1RoutinePanel : GameObject - Course2RoutinePanel : GameObject - Course3RoutinePanel : GameObject - Course4RoutinePanel : GameObject - Course5RoutinePanel : GameObject - Course6RoutinePanel : GameObject - Course1Code : Text - Course2Code : Text - Course3Code : Text - Course4Code : Text - Course5Code : Text - Course6Code : Text - Course1Day1Room : Text - Course2Day1Room : Text - Course3Day1Room : Text - Course4Day1Room : Text - Course5Day1Room : Text - Course6Day1Room : Text - Course1Day1Name : Text - Course2Day1Name : Text - Course3Day1Name : Text - Course4Day1Name : Text - Course5Day1Name : Text - Course6Day1Name : Text - Course1Day1Time : Text - Course2Day1Time : Text - Course3Day1Time : Text - Course4Day1Time : Text - Course5Day1Time : Text - Course6Day1Time : Text - Course1Day2Room : Text - Course2Day2Room : Text - Course3Day2Room : Text - Course4Day2Room : Text - Course5Day2Room : Text - Course6Day2Room : Text - Course1Day2Name : Text - Course2Day2Name : Text - Course3Day2Name : Text - Course4Day2Name : Text - Course5Day2Name : Text - Course6Day2Name : Text - Course1Day2Time : Text - Course2Day2Time : Text - Course3Day2Time : Text - Course4Day2Time : Text - Course5Day2Time : Text - Course6Day2Time : Text - courses : Courses - dbConnection: IDbConnection - dbCommand : IDbCommand - reader : IDataReader - path: string - code1 : string + ShowClassRoutine() + PanelAppearance()

Result
- dbConnection: IDbConnection - dbCommand : IDbCommand - reader : IDataReader - courses : Courses - semesterDropdown : Dropdown - total : float - sgpa : float - gpa : float - gpa1 : float - gpa2 : float - gpa3 : float - gpa4 : float - gpa5 : float - gpa6 : float - sgrade : string - grade : string - code : string - Course1Panel : GameObject - Course2Panel : GameObject - Course3Panel : GameObject - Course4Panel : GameObject - Course5Panel : GameObject - Course6Panel : GameObject - FinalPanel : GameObject - ResultPanel : GameObject - Code1Text : Text - Name1Text : Text - GPA1Text : Text - Grade1Text : Text - Credit1Text : Text - Code2Text : Text - Name2Text : Text - GPA2Text : Text - Grade2Text : Text - Credit2Text : Text - Code3Text : Text - Name3Text : Text - GPA3Text : Text - Grade3Text : Text - Credit3Text : Text - Code4Text : Text - Name4Text : Text - GPA4Text : Text - Grade4Text : Text - Credit4Text : Text - Code5Text : Text - Name5Text : Text - GPA5Text : Text - Grade5Text : Text - Code6Text : Text - Name6Text : Text - GPA6Text : Text - Grade6Text : Text - Credit6Text : Text - SGPAtext : Text - GradeText : Text - countCourse : int - Start() + ResultPublicationCheck() + SemesterResult() + GPACount() + Grade() + PanelAppearance() + HideCoursePanel() + ResultPanelActive()

4.2 Game Data Storage





4.3 Development Tools and Technologies

- Unity Game Engine (2019.3.0f6)
- C# Programming Language
- Visual Studio 2019
- DB Browser for SQLite

CHAPTER 5

SYSTEM TESTING

5.1 Testing Features

Testing is much needed approach to find any kind of bugs, functionality errors or development failures. In this project, there are different kinds of functionalities to be tested. I have tested all functionalities to ensure whether there exists any kind of errors or not. Testing result became positive. Here I will present half of the functionality tests only.

5.1.1 Features to be Tested

Features	Description
Enter player's name	Player can enroll himself/herself as a student of the department of Software Engineering at Daffodil International University.
View semester status	Player can check semester status whether it remains locked, registered or completed.
Register semester	Player can register semester.
View semester courses	Player can view semester courses.
Study courses	Player can study courses.
Pay fees	Player pays required mid-term or final-term fees for attending exam.
View semester result	Player can check semester result.
Pick up id/bag	Player can pick up his/her university id card or bag.
Enter into classroom	Player can enter into classroom.
Perform presentation	Player can perform presentation.
Give mid/final exam	Player can give exam.

5.2 Test Case

Test Case No- 1

Test case #1			Test case name: Enter player's name	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 1-1-2021	
Executed by: Antor			Execution date: 2-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player inputs his name in the input box.	Player is registered to the game to play.	Pass	Input box must not be null when registering.

Test Case No- 2

Test case #2			Test case name: View semester status	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 1-1-2021	
Executed by: Antor			Execution date: 2-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player clicks on 'Education Corner' button.	Player can check semester status whether it remains locked, registered or completed.	Pass	N/A

Test Case No- 3

Test case #3			Test case name: Register semester	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 1-1-2021	
Executed by: Antor			Execution date: 2-1-2021	

Step	Action	Response	Pass/Fail	Comment
1	Player clicks on 'Register' button.	Player can register himself for the semester.	Pass	Sufficient amount should remain in the balance.

Test Case No- 4

Test case #4		Test case name: View semester courses		
System: Game: Graduation		Sub-system: N/A		
Designed by: Antor		Design date: 1-1-2021		
Executed by: Antor		Execution date: 2-1-2021		
Step	Action	Response	Pass/Fail	Comment
1	Player clicks on 'Courses' button.	Player can check registered courses.	Pass	N/A

Test Case No- 5

Test case #5		Test case name: Study courses		
System: Game: Graduation		Sub-system: N/A		
Designed by: Antor		Design date: 3-1-2021		
Executed by: Antor		Execution date: 4-1-2021		
Step	Action	Response	Pass/Fail	Comment
1	Player clicks on 'Study' button.	Player is redirected to study courses area and can spend time.	Pass	To study courses player must be in front of reading table.

Test Case No- 6

Test case #6			Test case name: Pay fees	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 3-1-2021	
Executed by: Antor			Execution date: 4-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player clicks on 'Fees' button.	Player is redirected to payment area where he can pay mid or final fees.	Pass	Sufficient amount should be in the balance.

Test Case No- 7

Test case #7			Test case name: View semester result	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 3-1-2021	
Executed by: Antor			Execution date: 4-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player clicks on 'Result' button.	Player is redirected to result area where he can choose a semester and get result.	Pass	Result must be published.

Test Case No- 8

Test case #8			Test case name: Pick up id/bag	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 3-1-2021	
Executed by: Antor			Execution date: 4-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player presses down 'F' from keyboard.	Player picks up id card.	Pass	Id must be in the table.
2	Player presses down 'G' from keyboard.	Player picks up bag.	Pass	Bag must be in the table.

Test Case No- 9

Test case #9			Test case name: Enter into classroom	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 5-1-2021	
Executed by: Antor			Execution date: 6-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player presses down 'F' from keyboard.	Player enters into classroom if routine matches.	Pass	Player must be in front of door.

Test Case No- 10

Test case #10			Test case name: Perform presentation	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 5-1-2021	
Executed by: Antor			Execution date: 6-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player presses down 'F' from keyboard.	Player performs presentation.	Pass	Player must be in the classroom.

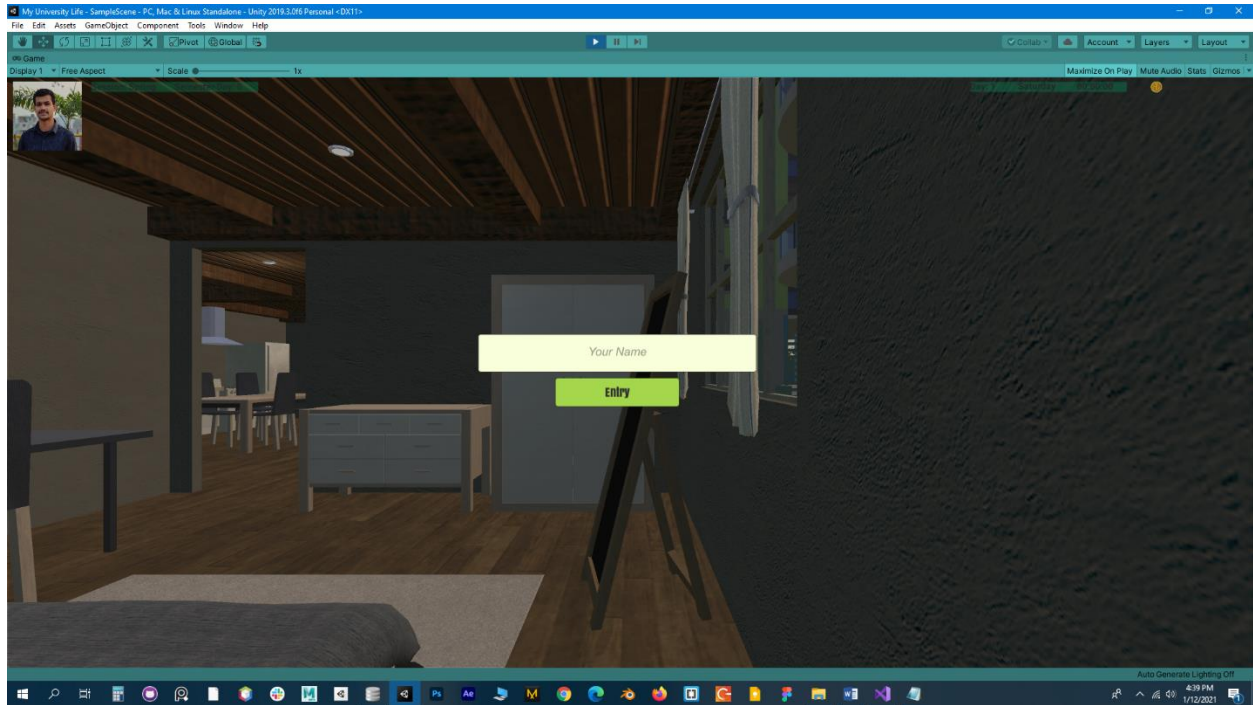
Test Case No- 11

Test case #11			Test case name: Give mid/final exam	
System: Game: Graduation			Sub-system: N/A	
Designed by: Antor			Design date: 5-1-2021	
Executed by: Antor			Execution date: 6-1-2021	
Step	Action	Response	Pass/Fail	Comment
1	Player presses down 'F' from keyboard.	Player enters into the exam hall.	Pass	Player must be in front of door.

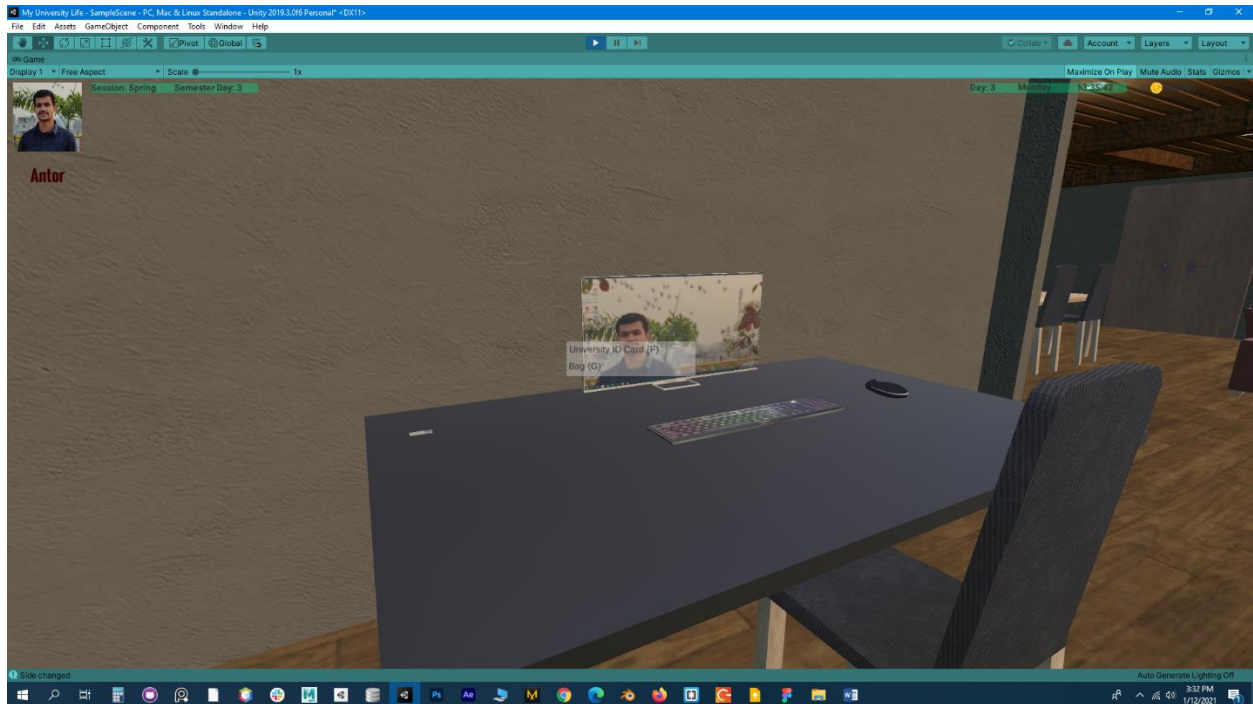
CHAPTER 6

USER MANUAL

6.1 Player Name Entry



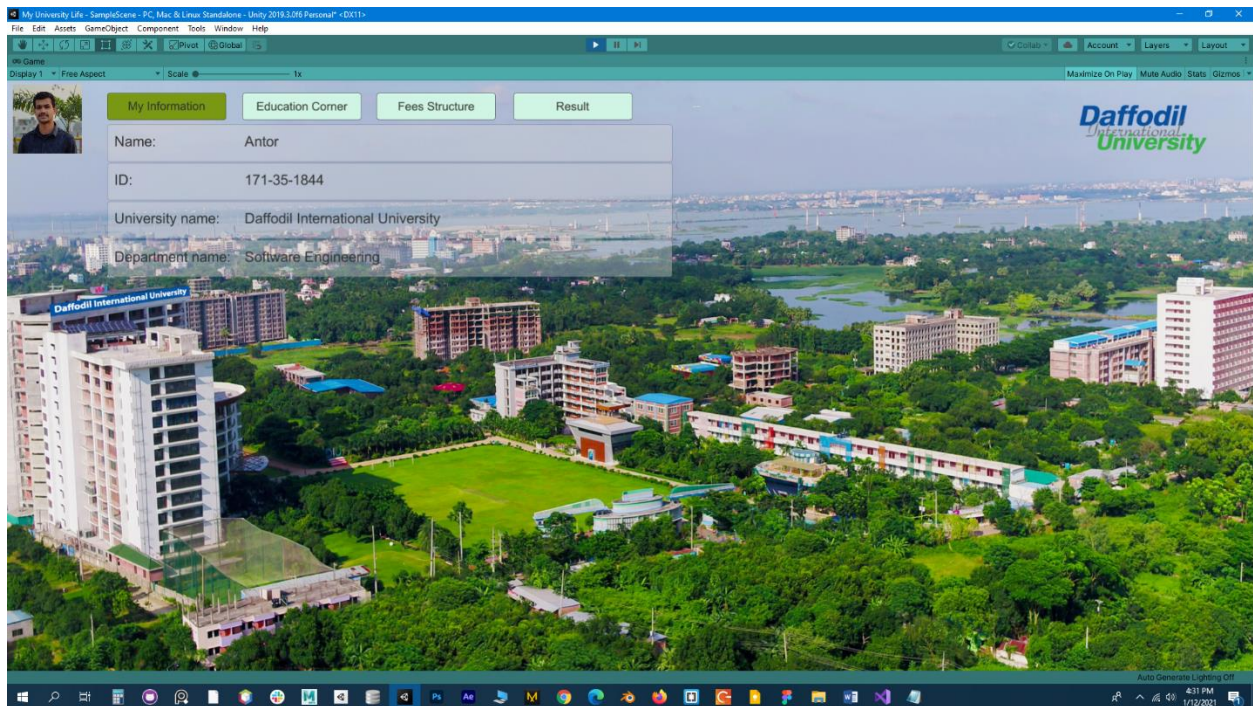
6.2 Pick Up ID/Bag



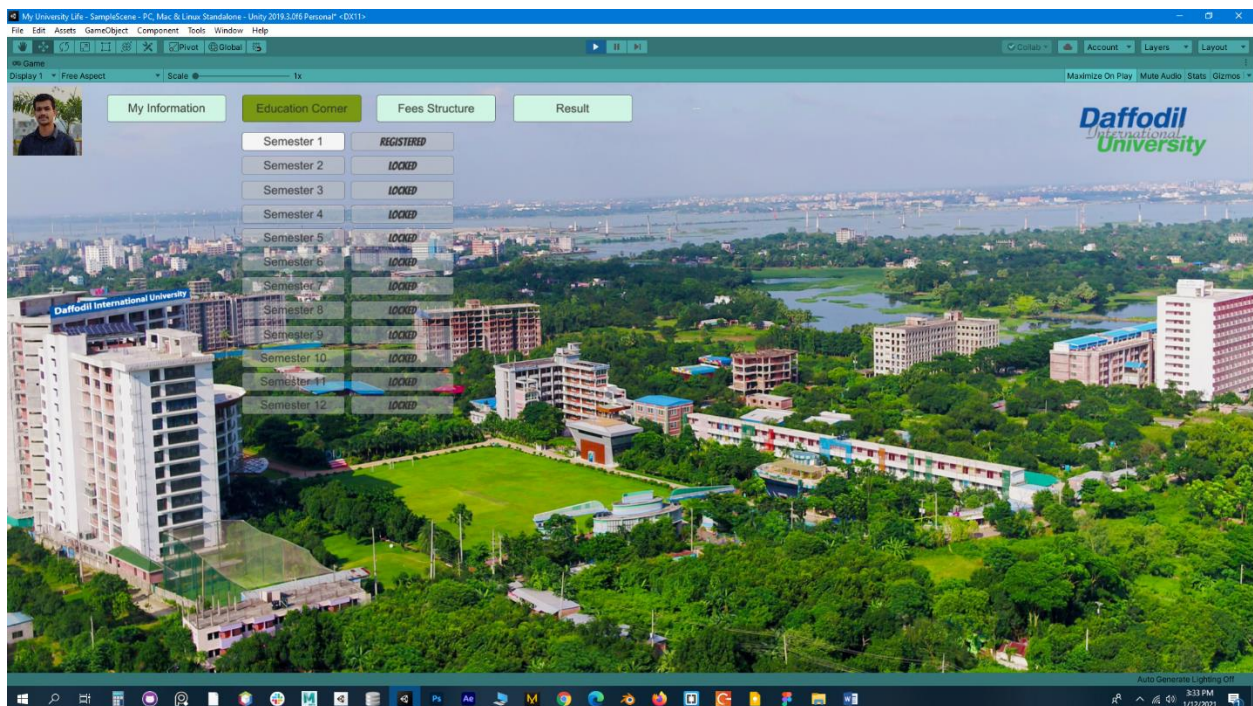
6.3 Sleeping Area



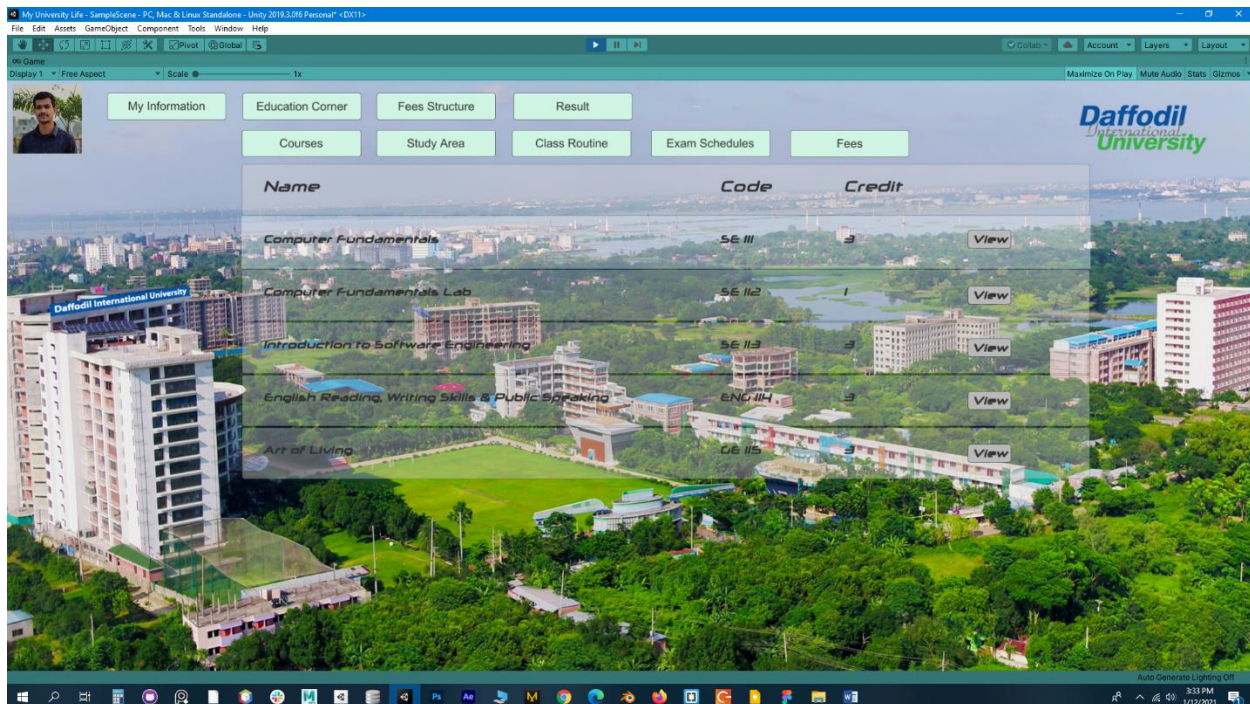
6.4 Player Information



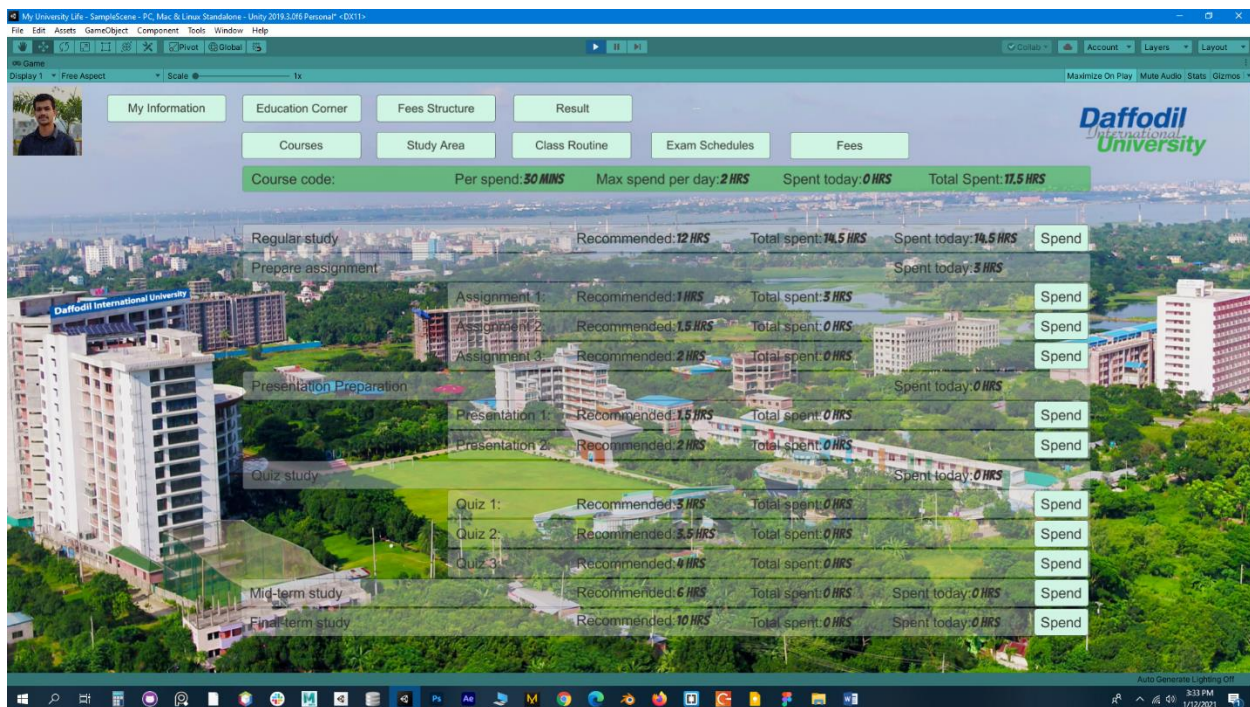
6.5 Semester Status



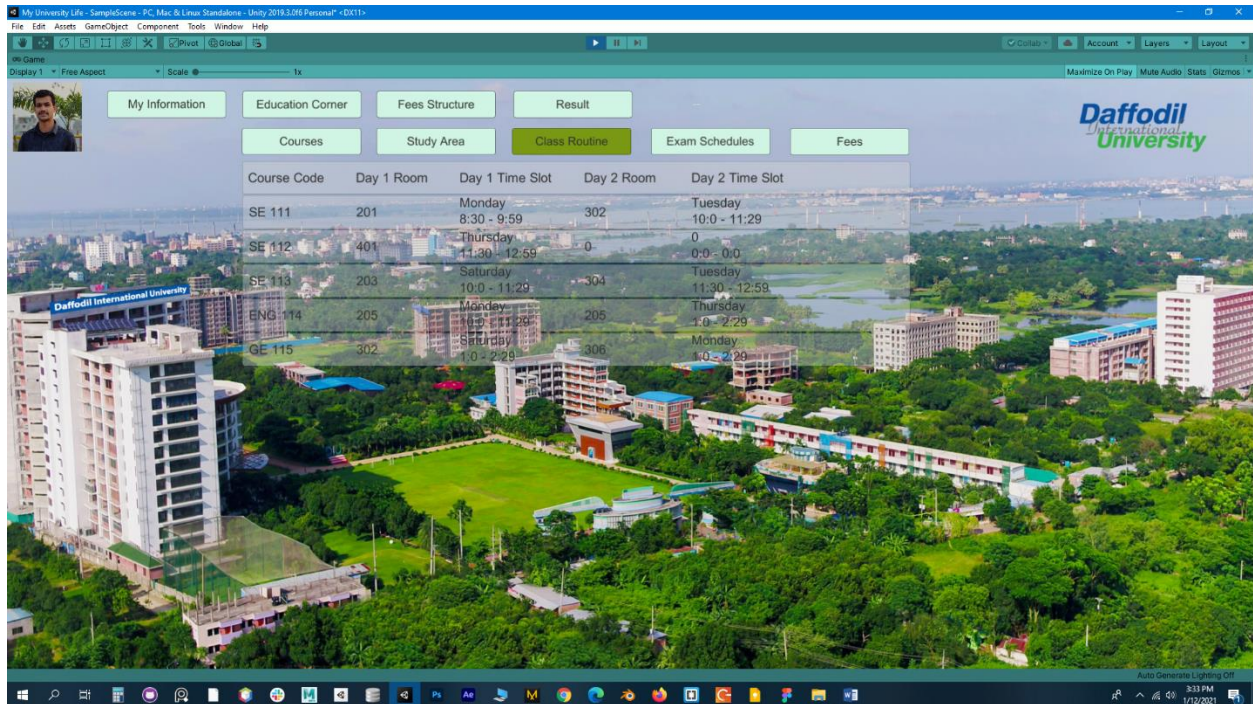
6.6 Registered Courses



6.7 Study Area

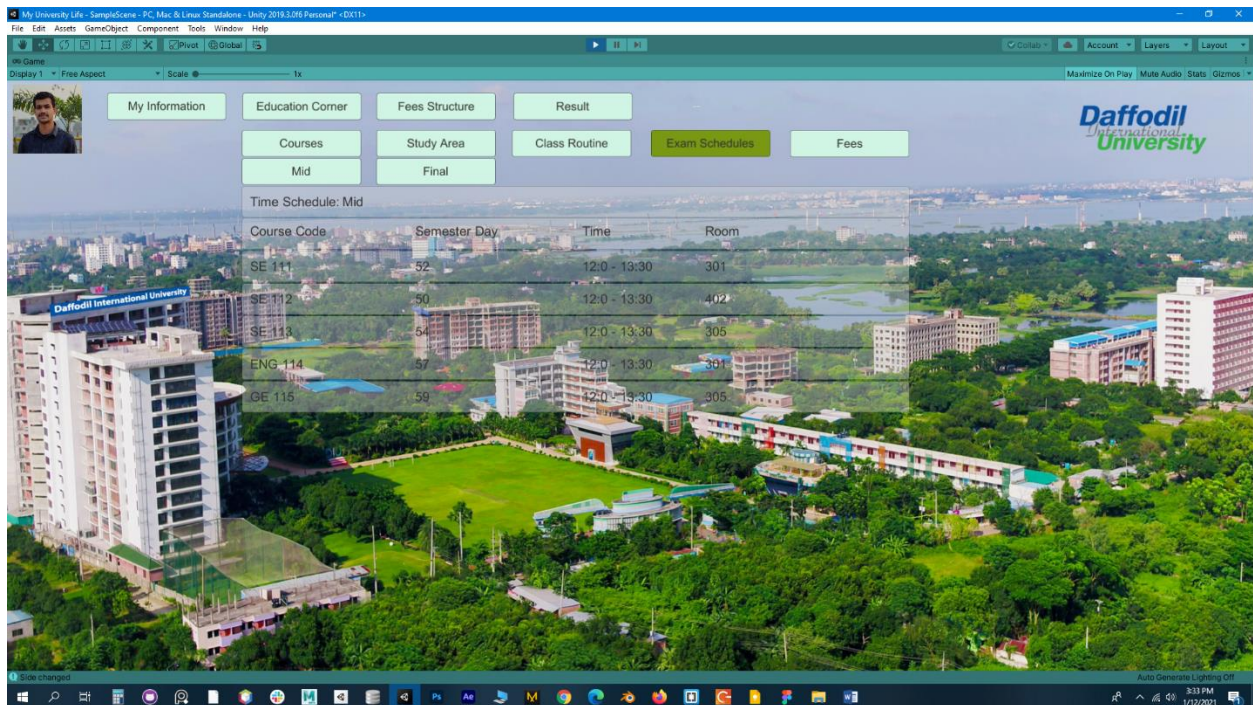


6.8 Class Routine



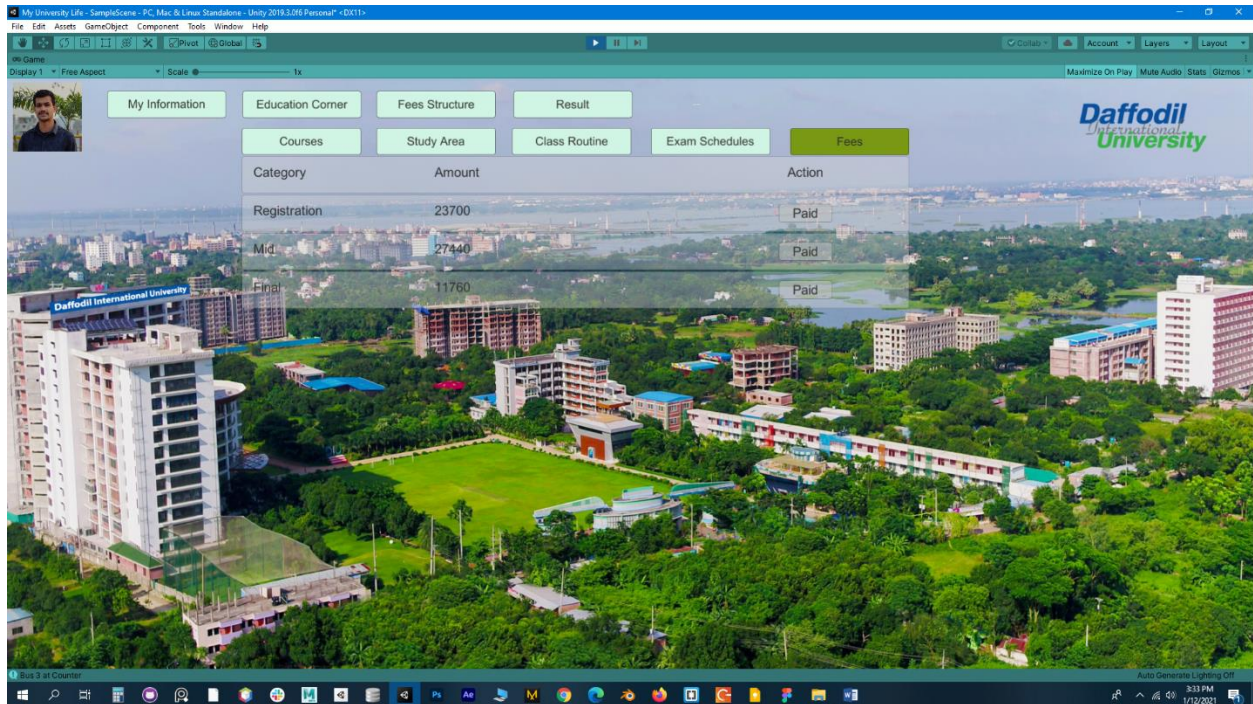
Course Code	Day 1 Room	Day 1 Time Slot	Day 2 Room	Day 2 Time Slot
SE 111	201	Monday 8:30 - 9:59	302	Tuesday 10:0 - 11:29
SE 112	401	Thursday 11:30 - 12:59	0	0 0:0 - 0:0
SE 113	203	Saturday 10:0 - 11:29	304	Tuesday 11:30 - 12:59
ENG 114	205	Monday 10:0 - 11:29	205	Thursday 1:0 - 2:29
GE 115	302	Saturday 1:0 - 2:29	306	Monday 1:0 - 2:29

6.9 Exam Schedules

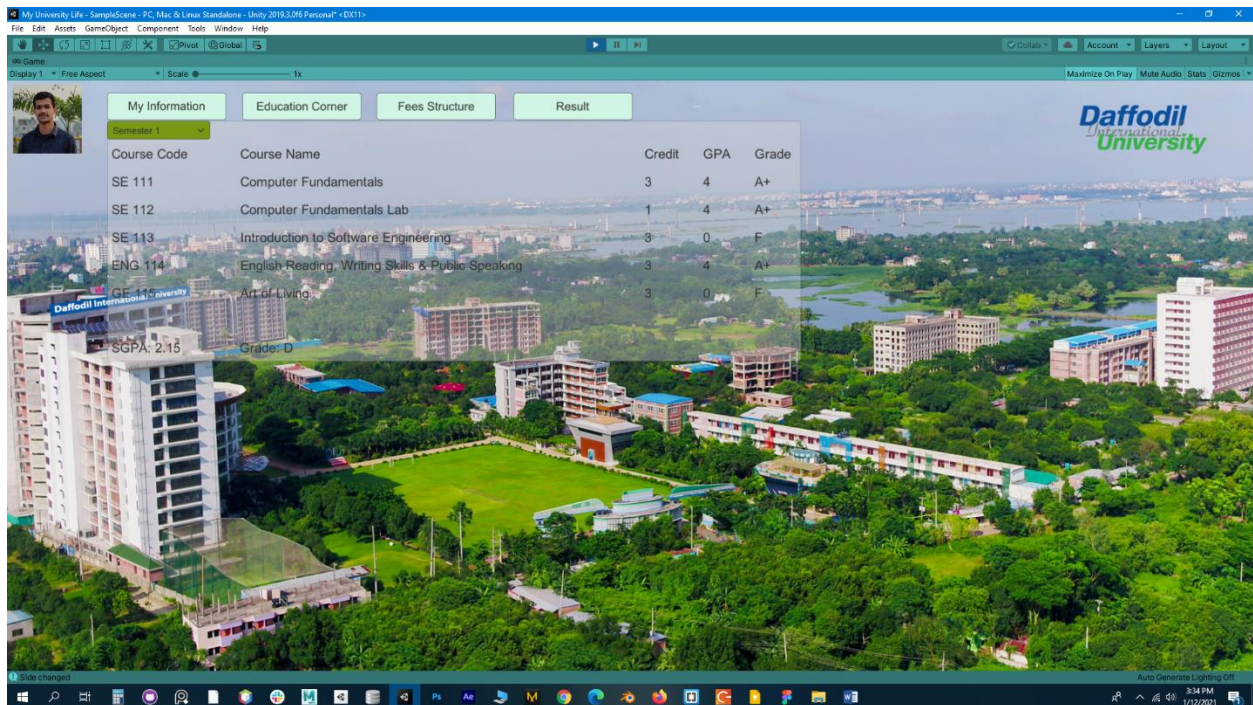


Course Code	Semester Day	Time	Room
SE 111	52	12:0 - 13:30	301
SE 112	50	12:0 - 13:30	402
SE 113	54	12:0 - 13:30	305
ENG 114	57	12:0 - 13:30	301
GE 115	59	12:0 - 13:30	305

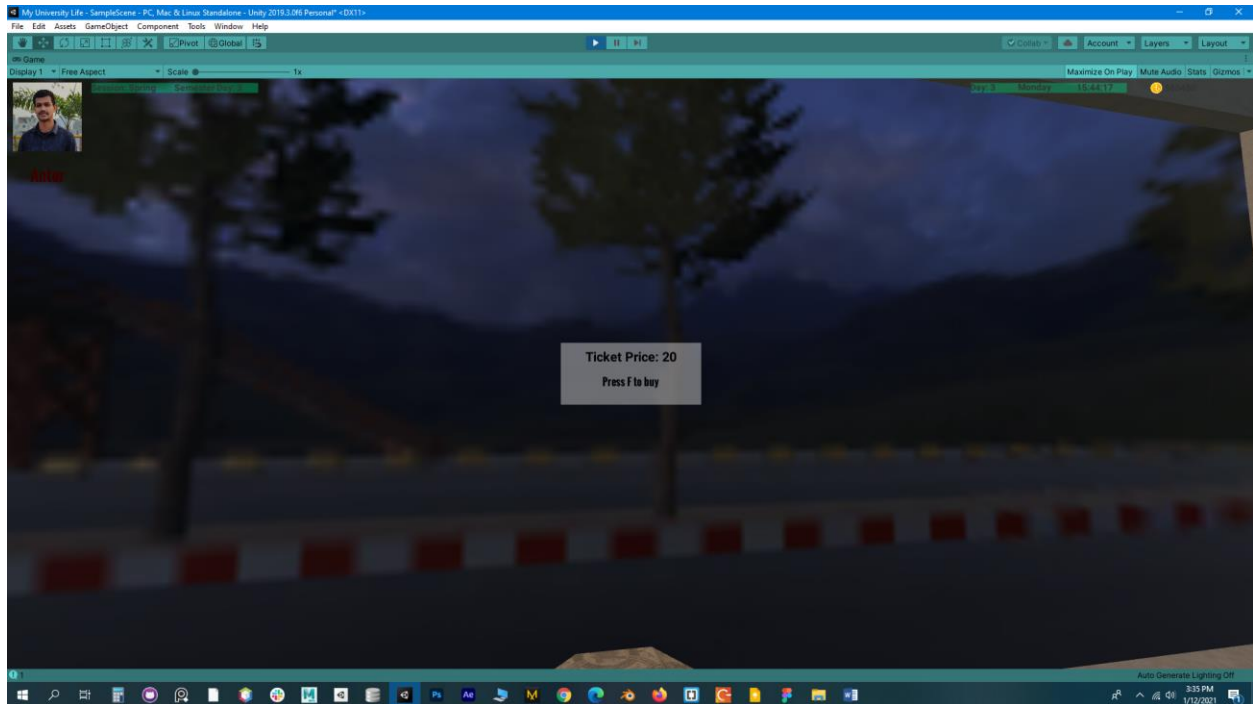
6.10 Fees



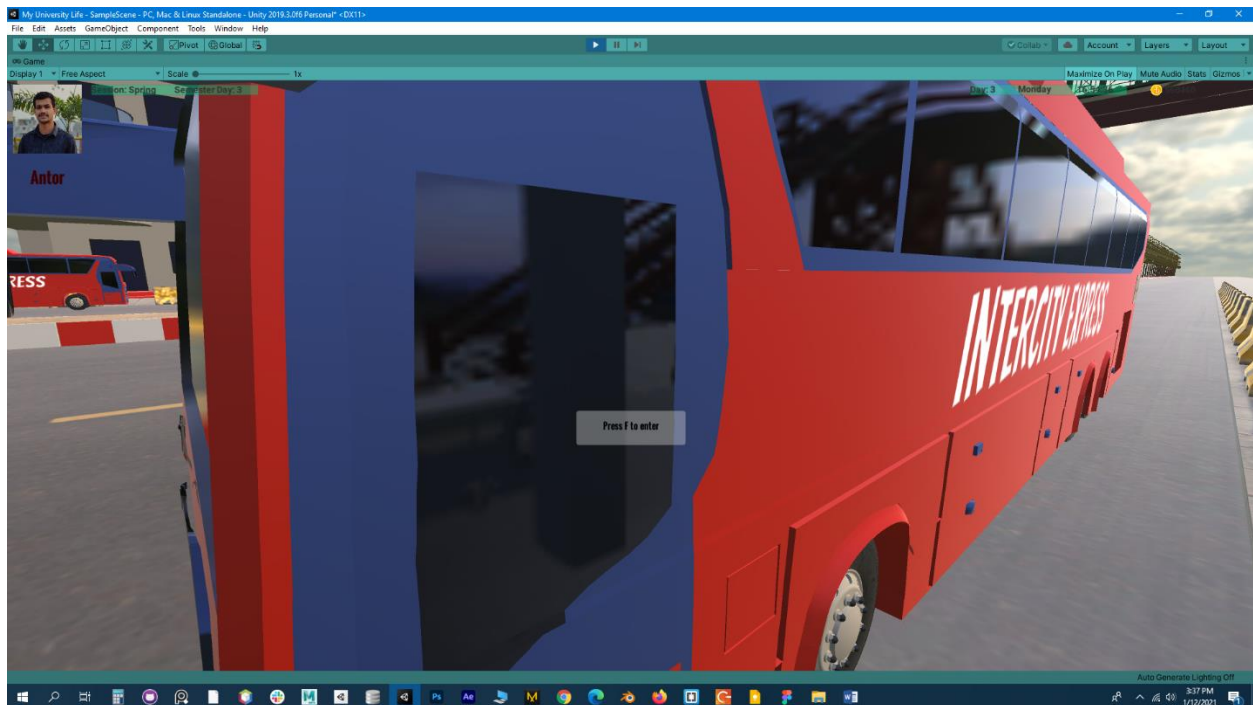
6.11 Semester Result



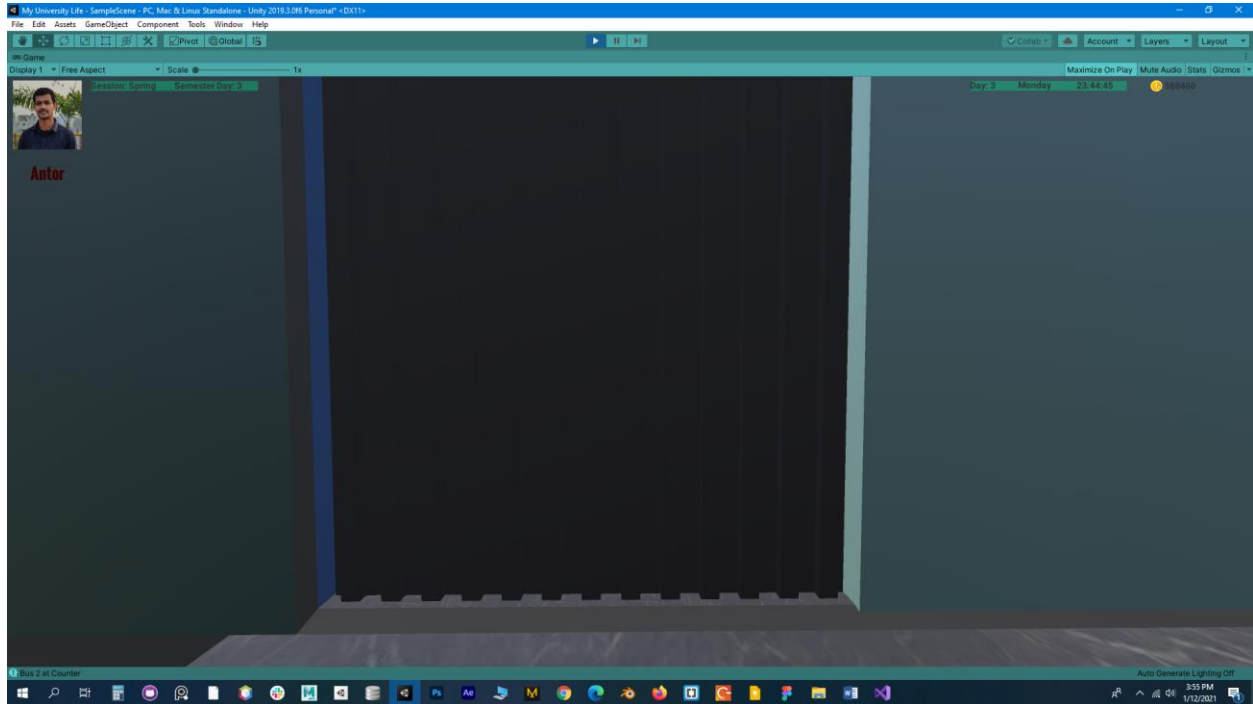
6.12 Buy Bus Ticket



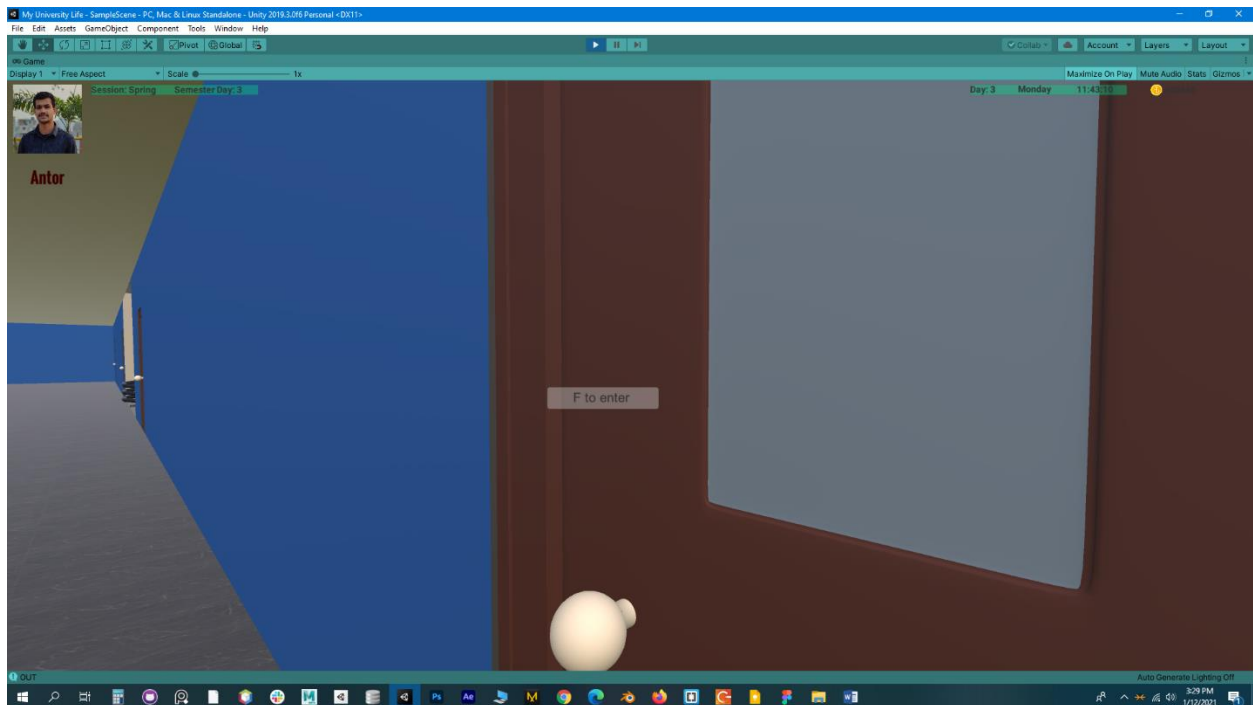
6.13 Enter into Bus



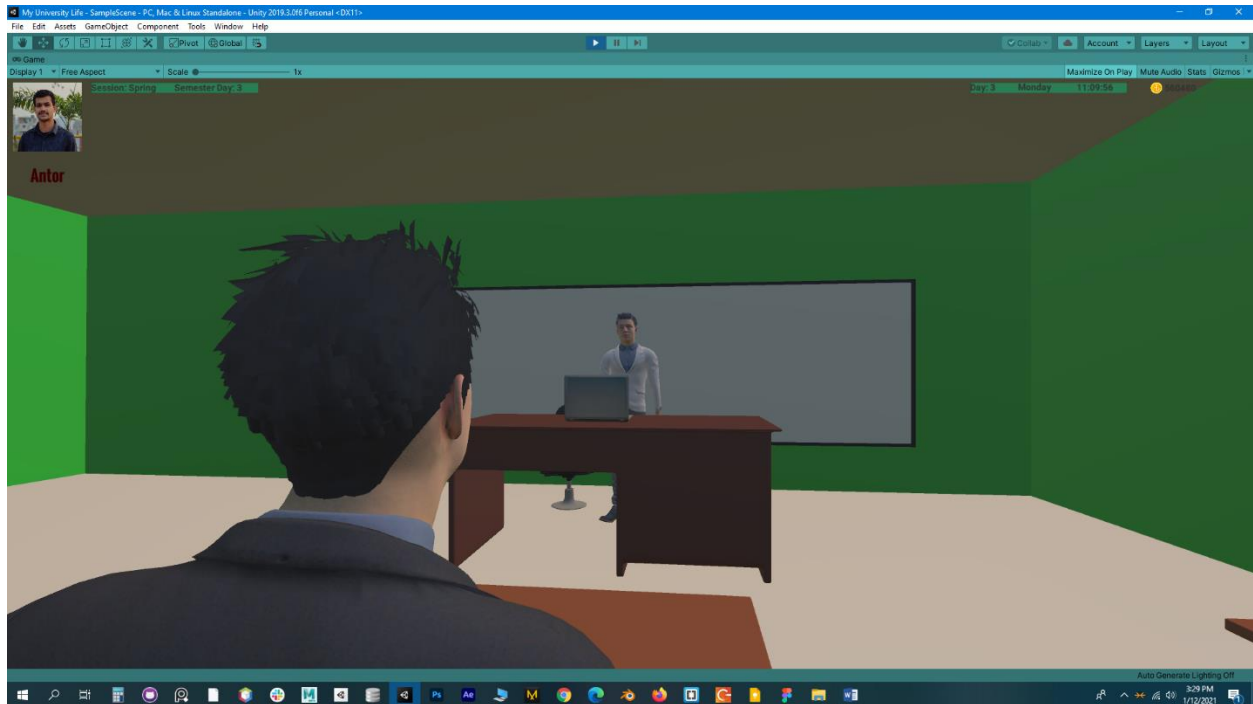
6.14 Enter into University



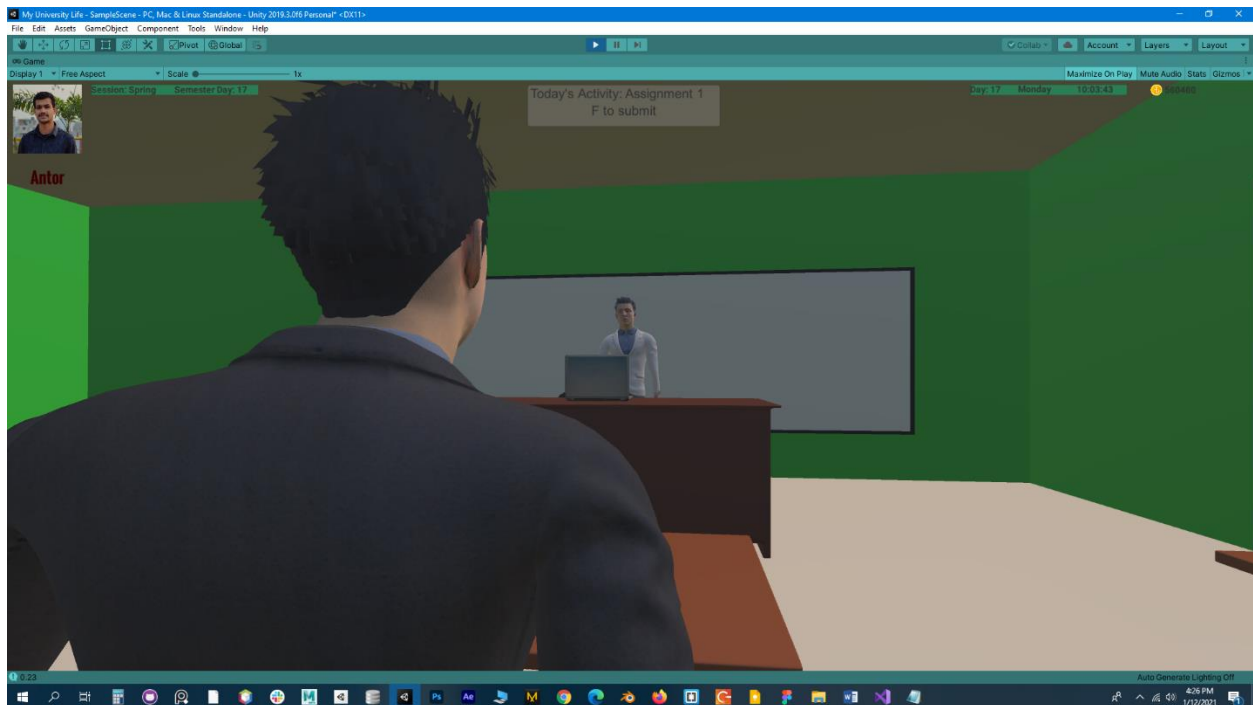
6.15 Enter into Classroom



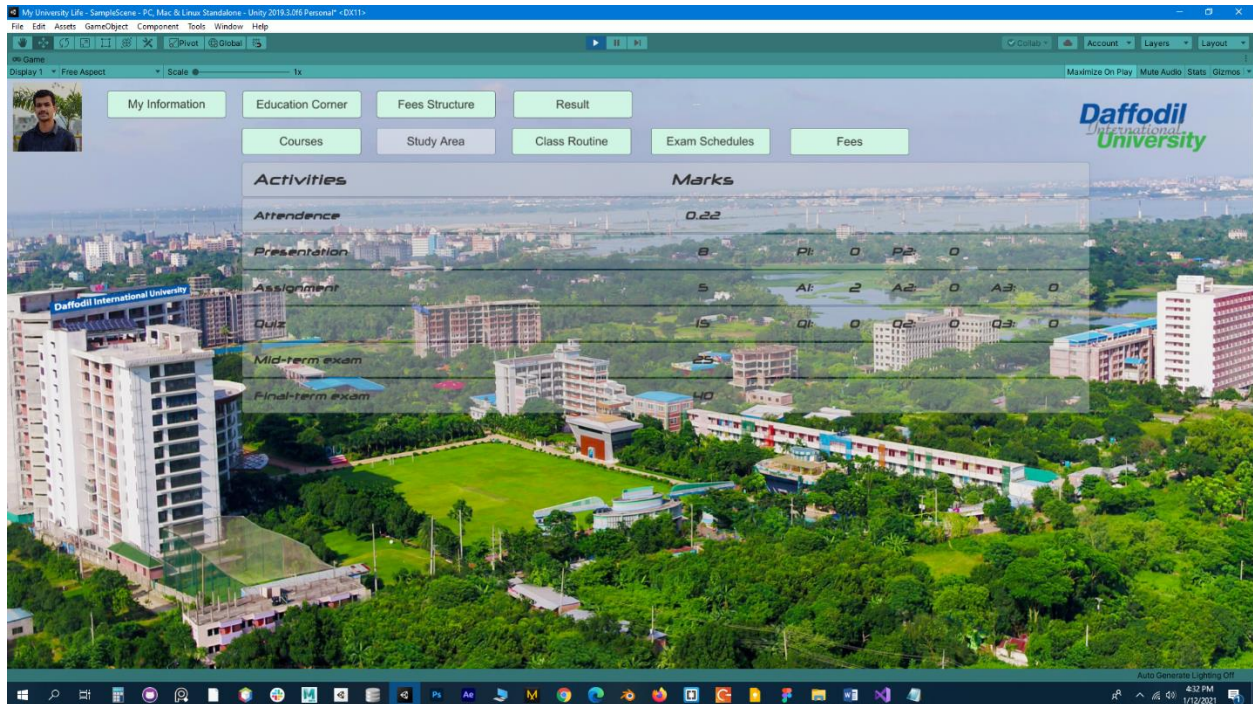
6.16 Class Moments



6.17 Class Activity Available



6.18 Course Marks



CHAPTER 7

PROJECT SUMMARY

7.1 Limitations

- Teacher names are not assigned under courses.
- Faculty information are not included.
- Player cannot retake any course individually. If player fails in any course of a semester, he/she has to register the semester again.
- Major in Cyber Security part is not included.
- Programming contests are not included.
- This game only runs on Windows only, not made for mobile users.

7.2 Obstacles & Achievement

Obstacles:

- As I am not so well in design part, 3D model designs was tough from me.
- I had to collect right information as the game is designed for knowing about the Software Engineering department of Daffodil International University.

Achievement:

- Implemented all the requirements successfully.
- Pleased with the game design.

7.3 Future Improvement

- Android version of this game will be revealed.
- Faculty information will be included.
- Retake systems will be added.
- Teacher names will be assigned under each course.
- Major in cyber security part will be included.
- Events will be added such as programming contests.

7.4 GitHub Link

<https://github.com/MazharulAntor/My-University-Life.git>

7.5 References

- [Unity Asset Store - The Best Assets for Game Making](#)
- [Mixamo](#)
- [Textures for 3D, graphic design and Photoshop!](#)
- [Free 3D Models | CGTrader](#)
- [Unity - Manual: Unity User Manual \(2019.3\) \(unity3d.com\)](#)

7.6 Plagiarism Report

1/30/2021

Turnitin

Document Viewer

Turnitin Originality Report

Processed on: 30-Jan-2021 12:51 +06
ID: 1497607387
Word Count: 6246
Submitted: 1

171-35-1844 By Md. Mazharul
Islam Antor

Similarity Index

14%

Similarity by Source

Internet Sources: 8%
Publications: 0%
Student Papers: 8%

[include quoted](#) [include bibliography](#) [excluding matches < 10 words](#) mode:

quickview (classic) report [print](#) [refresh](#) [download](#)

4% match (Internet from 01-Apr-2020)

<https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report>

2% match (student papers from 16-Dec-2019)

[Submitted to Higher Education Commission Pakistan on 2019-12-16](#)

1% match (student papers from 16-Feb-2018)

[Submitted to Higher Education Commission Pakistan on 2018-02-16](#)

1% match (student papers from 05-Nov-2018)

[Submitted to Sim University on 2018-11-05](#)

1% match (student papers from 31-Oct-2006)

[Submitted to Sim University on 2006-10-31](#)

1% match (student papers from 21-Oct-2012)

[Submitted to Higher Education Commission Pakistan on 2012-10-21](#)

1% match (Internet from 12-Aug-2020)

https://guido-dev.tk/reports/analysis_report.pdf

<1% match (Internet from 20-Dec-2019)

<http://dspace.daffodilvarsity.edu.bd:8080>

<1% match (Internet from 16-Jan-2021)

<https://www.coursehero.com/file/75006055/Final-Exampdf/>

<1% match (Internet from 10-Jan-2020)

<http://dspace.daffodilvarsity.edu.bd:8080>

<1% match (Internet from 20-Feb-2020)

<http://dspace.daffodilvarsity.edu.bd:8080>

<1% match (Internet from 28-Jul-2019)

<http://dspace.daffodilvarsity.edu.bd:8080>

https://www.turnitin.com/newreport_classic.asp?lang=en_us&oid=1497607387&ft=1&bypass_cv=1

1/12

<1% match (student papers from 26-Sep-2014) Submitted to University of Hertfordshire on 2014-09-26	✕
<1% match () http://ethesis.nitrkl.ac.in	✕
<1% match (student papers from 19-May-2016) Submitted to Higher Education Commission Pakistan on 2016-05-19	✕
<1% match (student papers from 30-May-2006) Submitted to University of Newcastle upon Tyne on 2006-05-30	✕
<1% match (Internet from 11-Oct-2008) http://www.eqr.uh.edu	✕
<1% match (Internet from 23-Aug-2012) http://www.timban.info	✕
<1% match (student papers from 27-May-2016) Submitted to Higher Education Commission Pakistan on 2016-05-27	✕
<1% match (student papers from 30-Nov-2020) Submitted to University of Portsmouth on 2020-11-30	✕

Game: Graduation Submitted By: Md. Mazharul Islam Antor ID: 171-35-1844
[Faculty of Science and Information Technology Department of Software Engineering Daffodil International University](#) Supervised By: Asif Khan Shakir
 Lecturer (Senior Scale) [Faculty of Science and Information Technology Department of Software Engineering Daffodil International University](#) Fall - 2020
 APPROVAL OF PROJECT I DECLARATION OF PROJECT II ACKNOWLEDGEMENTS At first, I want to give special thanks to my creator almighty Allah for giving me sufficient patience and of course His mercy on me for completing this project successfully in time. No plan can ever succeed without the permission of Allah. Next, I would like to give thanks to my supervisor Mr. Asif Khan Shakir for providing me the idea of this project and having trusts on me. I had to follow some game developers on YouTube to keep my project running. Big thanks to them. III ABSTRACT This game is designed for those newcomer students who are going to admit [in Software Engineering department of Daffodil International University](#). When playing this game, they can easily understand the curriculum activities, payment structures, marks distribution, etc., of the department. In addition, they will be familiar with semester courses. After starting official classes, they can utilize the learning knowledge from the game. There are 12 levels/semesters in the game. If a player fails in a semester, he/she cannot progress to further levels until he/she passes the failed level/semester. A player has to complete all the levels successfully to achieve the graduation certificate. IV TABLE OF CONTENTS APPROVAL OF PROJECT.....
I DECLARATION OF PROJECT
II
 ACKNOWLEDGEMENTS.....
III
 ABSTRACT.....
IV LIST OF FIGURES
VII
1.
INTRODUCTION.....
1.1.1 Project Overview

.....	1
<u>1.2 Project</u>	
<u>Purpose.....</u>	
1 <u>1.2.1</u>	
<u>Background.....</u>	
1 <u>1.2.2 Benefits and Beneficiaries</u>	
..... <u>2 1.2.3 Goal</u>	
.....	2
<u>1.3 Stakeholders</u>	
.....	2
<u>1.4 Proposed System</u>	
<u>Model.....</u>	2
<u>1.5 Project Schedule</u>	
.....	4
<u>1.5.1 Gantt</u>	
<u>Chart.....</u>	
<u>4.2. SOFTWARE REQUIREMENTS SPECIFICATION</u>	
..... <u>5 2.1 Software Requirements</u>	
Specification <u>..... 5 2.2</u>	
Functional Requirements	
..... <u>5 2.3 Non-</u>	
<u>Functional</u>	
<u>Requirements.....</u>	8
<u>2.3.1 Frames rate</u>	
.....	8
<u>2.3.2 Usability</u>	
.....	8
<u>2.3.3 Response time</u>	
.....	8
<u>2.3.4 Required resources</u>	
..... <u>9 2.3.5</u>	
Platform	
.....	9
<u>2.3.6 Maintainability</u>	
.....	9
<u>2.3.7 Safety</u>	
.....	9
<u>3. SYSTEM ANALYSIS</u>	
.....	10
<u>3.1 Use Case Diagram</u>	
.....	10
<u>3. 2 Use Case</u>	
<u>Description.....</u>	
12 <u>3.3 Activity Diagram</u>	
.....	23
<u>3. 4 Sequence Diagram</u>	
.....	45
<u>4. SYSTEM DESIGN</u>	
SPECIFICATION.....	67
Game Scripts	
.....	
67 <u>4.2 Game Data Storage</u>	
.....	71
<u>4.2 Development Tools & Technologies</u>	
.....	72
<u>5. SYSTEM TESTING</u>	
.....	73
<u>5.1 Testing</u>	
<u>Features.....</u>	
73 <u>5.1.1 Features to be Tested</u>	

.....	73	5.2
Test Case		
74 6. USER MANUAL		
79 6.1 Player Name Entry		
6.2 Pick Up ID/Bag	79	
80 6.3 Sleeping Area		
80 6.4 Player Information		
6.5 Semester	81	
Status.....		
81 6.6 Registered Courses		
6.7 Study Area	82	
82 6.8 Class Routine		
83 6.9 Exam Schedules		
83 6.10 Fees		
84 6.11 Semester Result		
6.12 Buy Bus Ticket	84	
85 6.13 Enter into Bus		
85 6.14 Enter into University		
6.15 Enter into	86	
Classroom.....		
86 6.16 Class Moments		
87 6.17 Class Activity		
Available.....		
87 6.18 Course		
Marks.....		
88 7. PROJECT SUMMARY		
Limitations.....	89	7.1
89 7.2 Obstacles & Achievement		
Future	89	7.3
Improvement.....		
89 7.4 GitHub Link		
90 7.5 References		
90 7.6 Plagiarism Report		
VI LIST OF FIGURES	91	
Point Name of Figure	Page No.	
Figure 1.1 Spiral	model 3	
Figure 1.2 Gantt chart	4 Figure 3. 1 Use Case Diagram	10
Figure 3.2 Activity	Diagram for 'Enter player's name'	23
Figure 3.3 Activity	Diagram for 'Check player's information'	24
Figure 3.4 Activity	Diagram for 'View semester status'	25
Figure 3. 5 Activity	Diagram for 'Register semester'	26
Figure 3. 6 Activity		

Diagram for 'View semester courses' 27 Figure 3. 7 Activity Diagram for 'View course marks' 28 Figure 3. 8 Activity Diagram for 'Study courses' 29 Figure 3. 9 Activity Diagram for 'View class routine' 30 Figure 3. 10 Activity Diagram for 'Pay fees' 31 Figure 3. 11 Activity Diagram for 'View exam schedules' 32 Figure 3. 12 Activity Diagram for 'View semester result' 33 Figure 3. 13 Activity Diagram for 'View fees structure' 34 Figure 3. 14 Activity Diagram for 'Pick up id/bag' 35 Figure 3.15 Activity Diagram for 'Buy bus ticket' 36 Figure 3.16 Activity Diagram for 'Take a seat on bus' 37 Figure 3.17 Activity Diagram for 'Enter into university' 38 Figure 3.18 Activity Diagram for 'Enter into classroom' 39 Figure 3. 19 Activity Diagram for 'Perform presentation' 40 Figure 3. 20 Activity Diagram for 'Submit assignment' 41 VII Figure 3. 21 Activity Diagram for 'Give quiz' 42 Figure 3. 22 Activity Diagram for 'Give mid/final exam' 43 Figure 3.23 Activity Diagram for 'Go to sleep' 44 Figure 3.24 Sequence Diagram for 'Enter player's name' 45 Figure 3.25 Sequence Diagram for 'Check player's information' 46 Figure 3.26 Sequence Diagram for 'View semester status' 47 Figure 3.27 Sequence Diagram for 'Register semester' 48 Figure 3.28 Sequence Diagram for 'View semester courses' 49 Figure 3.29 Sequence Diagram for 'View course marks' 50 Figure 3. 30 Sequence Diagram for 'Study courses' 51 Figure 3. 31 Sequence Diagram for 'View class routine' 52 Figure 3. 32 Sequence Diagram for 'Pay fees' 53 Figure 3. 33 Sequence Diagram for 'View exam schedules' 54 Figure 3. 34 Sequence Diagram for 'View semester result' 55 Figure 3. 35 Sequence Diagram for 'View fees structure' 56 Figure 3. 36 Sequence Diagram for 'Pick up id/bag' 57 Figure 3.37 Sequence Diagram for 'Buy bus ticket' 58 Figure 3.38 Sequence Diagram for 'Take a seat on bus' 59 Figure 3. 39 Sequence Diagram for 'Enter into university' 60 Figure 3. 40 Sequence Diagram for 'Enter into classroom' 61 Figure 3. 41 Sequence Diagram for 'Perform presentation' 62 Figure 3. 42 Sequence Diagram for 'Submit assignment' 63 Figure 3. 43 Sequence Diagram for 'Give quiz' 64 VIII Figure 3. 44 Sequence Diagram for 'Give mid/final exam' 65 Figure 3.45 Sequence Diagram for 'Go to sleep' 66 IX CHAPTER 1 INTRODUCTION 1.1 Project Overview

"Game: Graduation" is a game runs on windows platform. When entering the game for the first time game player will notice an input box to register his/her name. He will be registered as a student of Software Engineering department at Daffodil International University. There are twelve levels/semester in the game. You have to pass a semester to enter into next semester. Under each semester player can view his/her registered courses, semester class routine, exam schedule. For attending mid/final player has to pay required fees from fees section. In study section, player can study courses. Course studies will affect his/her semester results. Player has to attend classes to submit assignment, perform presentation or give quiz. Player can sleep for given hours to pass his/her game time quickly. University is far away from home, so player can use bus as transportation medium. Also player can overview the fees structure of the department. While playing the game, player can feel the graduation experiences of his/her department. 1.2 Project Purpose This game provides the best graduation experiences. User can get overall idea about the curriculum systems of Software Engineering department. He/she can gather fees related idea, which courses are taught in which semester. He will be able to get an idea about course activities, marks distribution. He/she will be able to know about class time allotment and class frequencies in a week. Furthermore, it is a game of learning with enjoyment for a newcomer student. 1.2.1 Background Students should know about his/her department's curriculum activities, fees structure, class time allotment, class frequencies per week, marks distribution, assigned courses, etc. Therefore, there should exist such a game by playing which he/she can gather enough knowledge about these things. Before admitting to the department, a newcomer can gather overall knowledge easily. He/she is playing the game and learning everything with a fun mode. It is the main reason behind this game project. 1.2.2 Benefits and Beneficiaries By playing this game, a newcomer student will be able to get an overall idea about the department's teaching procedures. In addition, those students who are planning to admit in Software

Engineering department will be benefited mostly. 1.2.3 Goal Our main goal is to provide the newcomer student proper knowledge about the Software Engineering department with an enjoyable way while he/she is playing the game. 1.3 Stakeholders In this project stakeholders are: 1. Project supervisor 2. Project developer 3. Users (playing the game) 1.4 Proposed System Model When developing a game, a model should be followed by the developer. In my case, I have followed spiral model to fulfill the project demand. I have to design the game scenario at first. After putting the scenario in the game, I must need the player to interact with the environment or other game functions. After completing coding part, there comes testing part whether the game is progressing or not. Then Evaluation matters most whether I will go with the game progress whatever I have done till now. This cycle happens repeatedly because when developing a game you need to go back for any design or coding correction. After final evaluation, you can declare the end of the project. [Figure 1.1: Spiral model 1.5 Project Schedule Project](#) schedule means an overall idea about the project processing time from start to end. To complete the project effectively there should exist a time schedule otherwise progressing will be halt. 1.5.1 Gantt Chart Weeks→ Works| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 Project Planning Requirement Collection Requirement Analysis Feasibility Study UML Diagrams Design Implementation Testing Evaluation Documentation Figure 1.2: Gantt chart [CHAPTER 2 SOFTWARE REQUIREMENTS SPECIFICATION 2.1 Software Requirements](#) Specification Software [requirement](#) specification describes how a software system needs to be developed. In many cases, it prevents project failure. It provides a detailed overview of the mentioned product and targeted audience, goals. It includes all functionalities and features of a product which need to be fulfilled as per as stakeholder needs. 2.2 Functional Requirements FR-01 Enter player's name Description Player can register his/her game name when first entering the game. Stakeholders Player Priority Medium FR-02 Check player's information Description Player can check his/her information such as his/her name, id number, department and university name. Stakeholders Player Priority Low FR-03 View semester status Description Player can check locked semester, or which semester he/she has completed or registered. Stakeholders Player Priority High FR-04 Register semester Description Player can register semester within deadline. Stakeholders Player Priority High FR-05 View semester courses Description Player can view semester courses. Stakeholders Player Priority Medium FR-06 View course marks Description Player can view course marks. Stakeholders Player Priority Medium FR-07 Study courses Description Player can study assigned courses for running semester. Stakeholders Player Priority High FR-08 View class routine Description Player can check class routine. Stakeholders Player Priority High FR-09 Pay fees Description Player can pay mid-term and final-term fees. Stakeholders Player Priority Medium FR-10 View exam schedules Description Player can check mid and final exam schedules to attend the exam in time. Stakeholders Player Priority High FR-11 View semester result Description Player can check published semester result. Stakeholders Player Priority Low FR-12 View fees structure Description Player can check fees structure determined by the department. Stakeholders Player Priority Low FR-13 Pick up id/bag Description Player can pick up his/her id and bag from his/her reading table. Stakeholders Player Priority Low FR-14 Buy bus ticket Description Player can buy bus ticket from bus counter. Stakeholders Player Priority Low FR-15 Take a seat on bus Description Player can enter into bus and take a seat if he/she has a ticket. Stakeholders Player Priority Low FR-16 Enter into university Description Player can enter into university if he/she has id card. Stakeholders Player Priority Low FR-17 Enter into classroom Description Player can enter into classroom matching class routine. Stakeholders Player Priority Medium FR-18 Perform presentation Description Player can perform presentation if available on that day. Stakeholders Player Priority High FR-19 Submit assignment Description Player can submit assignment if available on that day. Stakeholders Player Priority High FR-20 Give quiz Description Player can give quiz if available on that day. Stakeholders Player Priority High FR-21 Give mid/final exam Description

Player can give mid/final exam according to given schedule. Stakeholders Player Priority High FR-22 Go to sleep Description Player can sleep for given hours such as six, four, two or one hour. Stakeholders Player Priority Low [2.3 Non-Functional Requirements 2.3.1](#) Frames rate ? The game is easily playable at 30 FPS (frame per second). ? Frames dropping are not occurring when playing the game. 2.3.2 Usability ? Every functions are workable when player interacts. ? User experiences 3D graphics in the game. 2.3.3 Response time ? It takes less than 0.5 seconds for response when player interacts with a function. ? Player can play the game with a relax mode with great response time. 2.3.4 Required resources ? The game is playable with minimum of one GB RAM. ? This game does not take more than five to six GB. ? This game does not take too much graphics space. 2.3.5 Platform ? This game is made for windows platform only. 2.3.6 Maintainability ? The game is overlooked twice per week to find any kind of bugs or functional failures. 2.3.7 Safety ? The game does not harm user's health and eyes. ? This game does not show such things, which can have negative impacts to society. [CHAPTER 3 SYSTEM ANALYSIS 3.1 Use Case Diagram Figure 3.1: Use Case Diagram 3. 2 Use Case Description Use Case No. 01 Use Case Name](#) Enter player's name Goal Player can register his/her name. Preconditions Player has to open the game. Success End Condition Player can enroll himself/herself as a student of [the department of Software Engineering, at Daffodil International University](#). Failed End Condition Player cannot enroll himself/herself as a student. Primary Actors Player Secondary Actors N/A Trigger Player opens the game at first time. Description/Main Step Action Success Scenario 1 Player opens the game at first time. 2 Player gets the input box for entering his/her name. 3 Player puts his/her name and register himself/herself. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A](#) Use Case No. 02 Use Case Name Check player's information Goal Player can check his/her information. Preconditions Player has to play the game. Success End Condition Player can check his/her information. Failed End Condition Player cannot check his/her information. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'My Information' button. Description/Main Step Action Success Scenario 1 Player clicks on 'My Information' button. 2 Player views his/her information. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A](#) Use Case No. 03 Use Case Name View semester status Goal Player can check locked semester, or which semester he/she has completed or registered. Preconditions Player has to play the game. Success End Condition Player can check semester status whether it remains locked, registered or completed. Failed End Condition Player cannot check semester status. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Education Corner' button. Description/Main Step Action Success Scenario 1 Player clicks on 'Education Corner' button. 2 Player views semester status Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A](#) Use Case No. 04 Use Case Name Register semester Goal Player can continue his/her studies. Preconditions Registration date must be opened. Success End Condition Player can register semester. Failed End Condition Player cannot register semester. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Register' named button. Description/Main Step Action Success Scenario 1 Player clicks on 'Register' named button. 2 Player is registered for this semester. 3 Player can continue his/her studies. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A](#) Use Case No. 05 Use Case Name View semester courses Goal Player can check which courses are assigned for the semester. Preconditions Semester must be registered. Success End Condition Player can view semester courses. Failed End Condition Player fails to view semester courses. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Courses' button. Description/Main Step Action Success Scenario 1 Player clicks on 'Courses' button. 2 Player views semester courses. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A](#) Use Case No. 06 Use Case Name View course marks Goal Player can check course marks. Preconditions Semester must be

registered. Success End Condition Player views course marks. Failed End Condition Player fails to view course marks. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'View' button inside of 'Courses' area. Description/Main Step Action Success Scenario 1 Player clicks on 'View' button inside of 'Courses' area. 2 Player views course marks such as attendance, three assignments, two presentations, three quizzes, mid, final. Attendance, assignments, presentations, mid and final marks are viewable after semester result. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A](#) Use Case No. 07 Use Case Name Study courses Goal Player can study assigned courses for running semester. Preconditions Semester must be registered and running. Success End Condition Player can study courses. Failed End Condition Player cannot study courses. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Study' button inside of 'Study Area'. Description/Main Step Action Success Scenario 1 Player clicks on 'Study' button inside of 'Study Area'. 2 Then player can see different types of category of studying named as 'Regular study', 'Prepare assignment', 'Presentation preparation', 'Quiz study', 'Mid-term study' and 'Final-term study'. 3 Player spends time on a category clicking on 'Spend' button until per day maximum spend time not crossed. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) Player cannot spend more than given time per day. Use Case No. 08 Use Case Name View class routine Goal `javascript:void(0);` class routine. Preconditions Semester must be registered. Success End Condition Player can check class routine. Failed End Condition Player fails to check class routine. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Class Routine' button. Description/Main Step Action Success Scenario 1 Player clicks on 'Class Routine' button. 2 Player checks class routine. Alternative Step Branching Action Flows [1a](#) N/A [Quality Step](#) Requirement Use Case No. 09 Use Case Name Pay fees Goal Player can pay mid-term and final-term fees. Preconditions Semester must be registered. Success End Condition Player can successfully pay required fees. Failed End Condition Player fails to pay required fees. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Fees' button. Description/Main Step Action Success Scenario 1 Player clicks on 'Fees' button. 2 Player pays required mid-term or final-term fees for attending exam. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) Player has to pay exact given amount. Use Case No. 10 Use Case Name View exam schedules Goal Player can check mid-term and final-term exam schedules. Preconditions Semester must be registered. Success End Condition Player can successfully view exam schedules. Failed End Condition Player fails to view exam schedules. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Exam Schedules' button. Description/Main Step Action Success Scenario 1 Player clicks on 'Exam Schedules' button. 2 Player checks mid-term or final-term exam schedules. Alternative Step Branching Action Flows [1a](#) N/A [Quality Step](#) Requirement Use Case No. 11 Use Case Name View semester result Goal Player can check semester result. Preconditions Semester must be completed. Success End Condition Player can successfully check result. Failed End Condition Player cannot check result. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Result' button. Description/Main Step Action Success Scenario 1 Player clicks on 'Result' button. 2 Player chooses semester from dropdown list. 3 Player gets the result. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) If semester is not completed, result will not be shown. Use Case No. 12 Use Case Name View fees structure Goal Player can check fees structure. Preconditions Player has to play the game. Success End Condition Player can successfully check fees structure. Failed End Condition Player cannot check fees structure. Primary Actors Player Secondary Actors N/A Trigger Player clicks on 'Fees Structure' button. Description/Main Step Action Success Scenario 1 Player clicks on 'Fees Structure' button. 2 Player views fees structure determined by the department. Alternative Step Branching Action Flows [1a](#) N/A [Quality Step](#) Requirement Use Case No. 13 Use Case Name Pick up id/bag Goal Player can pick up his/her university id card

or bag. Preconditions Player has to play the game. Success End Condition Player can successfully pick up his/her id/bag. Failed End Condition Player cannot pick up id/bag. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword or 'G' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F' keyword and 'G' keyword from keyboard. 2 Player picks up id and bag respectively. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) N/A Use Case No. 14 Use Case Name Buy bus ticket Goal Player can buy bus ticket from counter. Preconditions Player has to stand in front of ticket counter. Success End Condition Player can successfully buys ticket. Failed End Condition Player cannot buy ticket for not having sufficient money. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F' keyword from keyboard. 2 Player buys up/down ticket. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement Use Case No. 15 Use Case Name Take a seat on bus Goal Player can get into the bus and take seat. Preconditions Player has to stand in front of bus door. Success End Condition Player can successfully get into the bus take a seat. Failed End Condition Player cannot get into the bus take a seat. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F' keyword from keyboard. 2 Player enters into the bus and takes a seat. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) Without [a](#) bus ticket, player cannot enter into the bus. Use Case No. 16 Use Case Name Enter into university Goal Player can enter into university. Preconditions Player has to stand in front of university gate. Success End Condition Player can enter into the university. Failed End Condition Player cannot enter into the university. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F' keyword from keyboard. 2 Player enters into the university. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) Player must have id card to enter. Use Case No. 17 Use Case Name Enter into classroom Goal Player can enter into classroom. Preconditions Player has to stand before classroom door. Success End Condition Player can enter into the classroom. Failed End Condition Player cannot enter into the classroom. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F' keyword from keyboard. 2 Player enters into the classroom. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) Player must match class routine to enter. Use Case No. 18 Use Case Name Perform presentation Goal Player can perform presentation. Preconditions Player has to attend the presentation class. Success End Condition Player can successfully perform presentation. Failed End Condition Player fails to perform presentation. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F' keyword from keyboard. 2 Player performs presentation. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) N/A Use Case No. 19 Use Case Name Submit assignment Goal Player can submit assignment. Preconditions Player has to attend the assignment submission class. Success End Condition Player can successfully submit assignment. Failed End Condition Player fails to submit assignment. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F' keyword from keyboard. 2 Player submits assignment. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) N/A Use Case No. 20 Use Case Name Give quiz Goal Player can give quiz. Preconditions Player has to attend the quiz class. Success End Condition Player can successfully give quiz. Failed End Condition Player fails to give quiz. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F' keyword from keyboard. Description/Main Step Action

Success Scenario 1 Player presses down 'F' keyword from keyboard. 2 Player gives quiz. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A](#) Use Case No. 21 Use Case Name Give mid/final exam Goal Player can give exam. Preconditions Player has to attend the exam room. Success End Condition Player can successfully give exam. Failed End Condition Player fails to give exam. Primary Actors Player Secondary Actors N/A Trigger Player enters into the exam hall. Description/Main Step Action Success Scenario 1 Player enters into the exam hall. 2 Player gives exam. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1](#) Exam schedules must be matched. Use Case No. 22 Use Case Name Go to sleep Goal Player can go to sleep. Preconditions Player has to stand in front of bed. Success End Condition Player can successfully go to sleep. Failed End Condition Player cannot go to sleep. Primary Actors Player Secondary Actors N/A Trigger Player presses down 'F', 'G', 'H' or 'J' keyword from keyboard. Description/Main Step Action Success Scenario 1 Player presses down 'F', 'G', 'H' or 'J' keyword from keyboard. 2 Player sleeps for six, four, two, one hour respectively for the 'F', 'G', 'H' and 'J' keyword. Alternative [Step Branching Action](#) Flows [1a](#) N/A [Quality Step](#) Requirement [Requirements 1 N/A 3.3](#) Activity Diagram [Figure 3.2: Activity Diagram for 'Enter player's name'](#) [Figure 3.3: Activity Diagram for 'Check player's information'](#) [Figure 3.4: Activity Diagram for 'View semester status'](#) [Figure 3.5: Activity Diagram for 'Register semester'](#) [Figure 3.6: Activity Diagram for 'View semester courses'](#) [Figure 3.7: Activity Diagram for 'View course marks'](#) [Figure 3.8: Activity Diagram for 'Study courses'](#) [Figure 3.9: Activity Diagram for 'View class routine'](#) [Figure 3.10: Activity Diagram for 'Pay fees'](#) [Figure 3.11: Activity Diagram for 'View exam schedules'](#) [Figure 3.12: Activity Diagram for 'View semester result'](#) [Figure 3.13: Activity Diagram for 'View fees structure'](#) [Figure 3.14: Activity Diagram for 'Pick up id/bag'](#) [Figure 3.15: Activity Diagram for 'Buy bus ticket'](#) [Figure 3.16: Activity Diagram for 'Take a seat on bus'](#) [Figure 3.17: Activity Diagram for 'Enter into university'](#) [Figure 3.18: Activity Diagram for 'Enter into classroom'](#) [Figure 3.19: Activity Diagram for 'Perform presentation'](#) [Figure 3.20: Activity Diagram for 'Submit assignment'](#) [Figure 3.21: Activity Diagram for 'Give quiz'](#) [Figure 3.22: Activity Diagram for 'Give mid/final exam'](#) [Figure 3.23: Activity Diagram for 'Go to sleep'](#) [3.4 Sequence Diagram](#) [Figure 3.24: Sequence Diagram for 'Enter player's name'](#) [Figure 3.25: Sequence Diagram for 'Check player's information'](#) [Figure 3.26: Sequence Diagram for 'View semester status'](#) [Figure 3.27: Sequence Diagram for 'Register semester'](#) [Figure 3.28: Sequence Diagram for 'View semester courses'](#) [Figure 3.29: Sequence Diagram for 'View course marks'](#) [Figure 3.30: Sequence Diagram for 'Study courses'](#) [Figure 3.31: Sequence Diagram for 'View class routine'](#) [Figure 3.32: Sequence Diagram for 'Pay fees'](#) [Figure 3.33: Sequence Diagram for 'View exam schedules'](#) [Figure 3.34: Sequence Diagram for 'View semester result'](#) [Figure 3.35: Sequence Diagram for 'View fees structure'](#) [Figure 3.36: Sequence Diagram for 'Pick up id/bag'](#) [Figure 3.37: Sequence Diagram for 'Buy bus ticket'](#) [Figure 3.38: Sequence Diagram for 'Take a seat on bus'](#) [Figure 3.39: Sequence Diagram for 'Enter into university'](#) [Figure 3.40: Sequence Diagram for 'Enter into classroom'](#) [Figure 3.41: Sequence Diagram for 'Perform presentation'](#) [Figure 3.42: Sequence Diagram for 'Submit assignment'](#) [Figure 3.43: Sequence Diagram for 'Give quiz'](#) [Figure 3.44: Sequence Diagram for 'Give mid/final exam'](#) [Figure 3.45: Sequence Diagram for 'Go to sleep'](#) CHAPTER 4 SYSTEM DESIGN SPECIFICATION 4.1 Game Scripts There are more than forty scripts in the game. Here, I am providing some scripts. 4.2 Game Data Storage 4.3 Development Tools and Technologies ? Unity Game Engine (2019.3.0f6) ? C# Programming Language ? Visual Studio 2019 ? DB Browser for SQLite [CHAPTER 5 SYSTEM TESTING 5.1 Testing](#) Features [Testing is](#) much needed approach [to](#) find any kind [of](#) bugs, functionality errors or development failures. In this project, there are different kinds of functionalities to be tested. I have tested all functionalities to ensure whether there exists any kind of errors or not. Testing result became positive. Here I will present half of the functionality tests only. [5.1.1 Features to be Tested Features Description](#) Enter player's name Player can

enroll himself/herself as a student of [the department of Software Engineering](#) at [Daffodil International University](#). View semester status Player can check semester status whether it remains locked, registered or completed. Register semester Player can register semester. View semester courses Player can view semester courses. Study courses Player can study courses. Pay fees Player pays required mid-term or final-term fees for attending exam. View semester result Player can check semester result. Pick up id/bag Player can pick up his/her university id card or bag. Enter into classroom Player can enter into classroom. Perform presentation Player can perform presentation. Give mid/final exam Player can give exam. 5. 2 Test Case Test Case No- 1 Test case #1 Test case name: Enter player's name System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 1-1- 2021 Executed by: Antor Execution date: 2-1- 2021 Step Action Response Pass/Fail Comment 1 Player inputs his name in the `javascript:void(0);` Player is registered to the game to play. Pass Input box must not be null when registering. Test Case No- 2 Test case #2 Test case name: View semester status System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 1-1- 2021 Executed by: Antor Execution date: 2-1- 2021 Step Action Response Pass/Fail Comment 1 Player clicks on 'Education Corner' button. Player can check semester status whether it remains locked, registered or completed. Pass N/A Test Case No- 3 Test case #3 Test case name: Register semester System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 1-1- 2021 Executed by: Antor Execution date: 2-1- 2021 Step Action Response Pass/Fail Comment 1 Player clicks on 'Register' button. Player can register himself for the semester. Pass Sufficient amount should remain in the balance. Test Case No- 4 Test case #4 Test case name: View semester courses System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 1-1- 2021 Executed by: Antor Execution date: 2-1- 2021 Step Action Response Pass/Fail Comment 1 Player clicks on 'Courses' button. Player can check Pass registered courses. N/A Test Case No- 5 Test case #5 Test case name: Study courses System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 3-1-2021 Executed by: Antor Execution date: 4-1-2021 Step Action Response Pass/Fail Comment 1 Player clicks on 'Study' button. Player is redirected to study courses area and can spend time. Pass To study courses player must be in front of reading table. Test Case No- 6 Test case #6 Test case name: Pay fees System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 3-1-2021 Executed by: Antor Execution date: 4-1-2021 Step Action Response Pass/Fail Comment 1 Player clicks on 'Fees' button. Player is redirected to payment area where he can pay mid or final fees. Pass Sufficient amount should be in the balance. Test Case No- 7 Test case #7 Test case name: View semester result System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 3-1-2021 Executed by: Antor Execution date: 4-1-2021 Step Action Response Pass/Fail Comment 1 Player clicks on 'Result' button. Player is redirected to result area where he can choose a semester and get result. Pass Result must be published. Test Case No- 8 Test case #8 Test case name: Pick up id/bag System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 3-1-2021 Executed by: Antor Execution date: 4-1-2021 Step Action Response Pass/Fail Comment 1 Player presses down 'F' from keyboard. Player picks up id card. Pass Id must be in the table. 2 Player presses down 'G' from keyboard. Player picks up bag. Pass Bag must be in the table. Test Case No- 9 Test case #9 Test case name: Enter into classroom System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 5-1-2021 Executed by: Antor Execution date: 6-1-2021 Step Action Response Pass/Fail Comment 1 Player presses down 'F' from keyboard. Player enters into Pass classroom if routine matches. Player must be in front of door. Test Case No- 10 Test case #10 Test case name: Perform presentation System: Game: Graduation Sub-system: N/A Designed by: Antor Design date: 5-1-2021 Executed by: Antor Execution date: 6-1-2021 Step Action Response Pass/Fail Comment 1 Player presses down 'F' from keyboard. Player performs Pass presentation. Player must be in the classroom. Test Case No- 11 Test case #11 Test case name: Give mid/final exam System: Game: Graduation

Sub-system: N/A Designed by: Antor Design date: 5-1-2021 Executed by: Antor
 Execution date: 6-1-2021 Step Action Response Pass/Fail Comment 1 Player
 presses down 'F' from keyboard. Player enters into Pass the exam hall. Player
 must be in front of door. CHAPTER 6 USER MANUAL 6.1 Player Name Entry 6.2
 Pick Up ID/Bag 6.3 Sleeping Area 6.4 Player Information 6.5 Semester Status 6.6
 Registered Courses 6.7 Study Area 6.8 Class Routine 6.9 Exam Schedules 6.10
 Fees 6.11 Semester Result 6.12 Buy Bus Ticket 6.13 Enter into Bus 6.14 Enter
 into University 6.15 Enter into Classroom 6.16 Class Moments 6.17 Class Activity
 Availavle 6.18 Course Marks CHAPTER 7 PROJECT SUMMARY 7.1 Limitations ?
 Teacher names are not assigned under courses. ? Faculty information are not
 included. ? Player cannot retake any course individually. If player fails in any
 course of a semester, he/she has to register the semester again. ? Major in Cyber
 Security part is not included. ? Programming contests are not included. ? This
 game only runs on Windows only, not made for mobile users. 7.2 Obstacles &
 Achievement Obstacles: ? As I am not so well in design part, 3D model designs
 was tough from me. ? I had to collect right information as the game is designed
 for knowing about the Software Engineering department of Daffodil International
 University. Achievement: ? Implemented all the requirements successfully. ?
 Pleased with the game design. 7.3 Future Improvement ? Android version of this
 game will be revealed. ? Faculty information will be included. ? Retake systems
 will be added. ? Teacher names will be assigned under each course. ? Major in
 cyber security part will be included. ? Events will be added such as programming
 contests. 7.4 GitHub Link <https://github.com/MazharulAntor/My-University-Life.git> 7.5 References ? Unity Asset Store - The Best Assets for Game Making ?
 Mixamo ? Textures for 3D, graphic design and Photoshop! ? Free 3D Models |
 CGTrader ? Unity - Manual: Unity User Manual (2019.3) (unity3d.com) 7.6
 Plagiarism Report [Requirements 1 N/](#) A [Requirements 1 N/](#) A [Requirements 1 N/](#)
 A [Requirements 1 N/](#) A [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#)
[22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [46](#)
[47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#) [58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#) [67](#) [68](#) [69](#) [70](#) [71](#)
[72](#) [73](#) [74](#) [75](#) [76](#) [77](#) [78](#) [79](#) [80](#) [81](#) [82](#) [83](#) [84](#) [85](#) [86](#) [87](#) [88](#) [89](#) [90](#) [91](#)