***Online Learning System***

***Created by***

***Ryan Jenkins***

***20088363***

Table of Contents

[***SECTION 1*** 3](#_Toc79931409)

[**1.1.** **USE CASE** 3](#_Toc79931410)

[**1.2.** **USE CASE DIAGRAM** 4](#_Toc79931411)

[**1.3** **USE CASE DESCRIPTION** 5](#_Toc79931412)-11

# ***SECTION 1***

## **USE CASE**

The primary form of system/software requirements for an undeveloped software application is a UML use case diagram. The desired behavior (what) is specified in use cases, not the specific technique of achieving it (how). Once defined, use cases can be represented both textually and visually (i.e., use case diagram). A fundamental notion in use case modelling is that it assists us in designing a system from the standpoint of the end user. It's a good way to communicate system behavior to users in their own words by specifying all externally visible system behavior.

A use case diagram is often straightforward. It doesn't go into the specifics of the use cases:

* It barely touches on a few of the connections between use cases, actors, and systems.
* It doesn't reveal the sequence in which actions are completed to fulfil each use case's goals.

## **USE CASE DIAGRAM**



**Use Case Diagram**

## **USE CASE DESCRIPTION**

**Login:**

|  |  |
| --- | --- |
| Use case name | UC\_ Login |
| Actors | Student / Staff |
| Pre-Condition | Student and Staff must be on registration page |
| Flow of events | 1. Open Registration web page 2. Provide their personal information 3. The user enters their username and password 4. Click on Sign In button |
| Alternative flow | 1. If any input field is empty show error message 2. Go back to Login page |
| Failure scenario | 1. The user forgets personal information and/or password 2. The user isn’t able to sign in |
| Successful scenario | Sign In successfully |

**Successful log in diagram:**

Diagram

Description automatically generated

**Search Course**

|  |  |
| --- | --- |
| Use case name | UC\_ Search Courses |
| Actors | Student |
| Pre-Condition | Student must login on system. |
| Flow of events | 1. Click search course 2. Enter course name or code 3. Click ok |
| Alternative flow | 1. If any input field is empty show error message 2. Go back to search page |
| Failure scenario | 1. The user doesn’t find any courses that interest them 2. The user uses a different site to find a course |
| Post condition | Search course successfully |

**Successful course search diagram:**

Diagram

Description automatically generated

**Learn Course**

|  |  |
| --- | --- |
| Use case name | UC\_ Learn Courses |
| Actors | Student |
| Pre-Condition | Student must login on system. |
| Flow of events | 1. The user browses for a specific course 2. Select Course 3. Click ok 4. The user will look at all of the course requirements 5. The user will assess whether they want to do this specific course |
| Alternative flow | 1. If any input field is empty show error message 2. Go back to search page |
| Post condition | Course successfully added to profile. |

**Make Payment**

|  |  |
| --- | --- |
| Use case name | UC\_ Make Payment |
| Actors | Student |
| Pre-Condition | Student must login on system. |
| Flow of events | 1. Enter Payment Method 2. Enter Account Details 3. Enter Amount 4. Click ok 5. Payment is processed |
| Alternative flow | 1. If the account balance is less than required balance then send error message. 2. Go back to payment page |
| Failure scenario | 1. The payment doesn’t go through 2. The user has to communicate to their bank about the transaction 3. The user can’t pay for the course at this time |
| Post condition | Payment makes successfully. |

**Successful Payment diagram:**

Diagram

Description automatically generated

**Communicate Lecturer**

|  |  |
| --- | --- |
| Use case name | UC\_ Communicate Lecturer |
| Actors | Student |
| Pre-Condition | Student must login on system. |
| Flow of events | 1. Go to Communication Page 2. Select Lecturer 3. Select communication Method 4. Wright Message 5. Click send. |
| Alternative flow | 1. If message field is empty then it will show a error message 2. Go back to communication page. |
| Failure scenario | 1. The user selects the wrong lecturer to message 2. The user has sent a message to the wrong lecturer regarding specific information |
| Post condition | The user has successfully communicated with the lecturer |

**Successful Communication diagram:**

Diagram

Description automatically generated

**Rating**

|  |  |
| --- | --- |
| Use case name | UC\_ Give Rating |
| Actors | Student |
| Pre-Condition | Student must login on system. |
| Flow of events | 1. Click on rating. 2. Select Stars 3. Write Comments 4. Click Publish |
| Alternative flow | 1. If any input field is empty show error message 2. Go back to rating page |
| Failure scenario | 1. The user can’t find where to give a rating 2. The user never got to leave their opinion or feedback on the course 3. The lecturer won’t receive constructive feedback on the course |
| Post condition | The user successfully leaves their rating and feedback on the course. |

**Successful User Rating diagram:**

Diagram

Description automatically generated

**Add Course**

|  |  |
| --- | --- |
| Use case name | UC\_ Add Course |
| Actors | Staff |
| Pre-Condition | Staff must login on system. |
| Flow of events | 1. Click on add new course 2. Enter course details 3. Search promptly for the course 4. Click Add |
| Alternative flow | 1. If any input field is empty show error message 2. Go back to courses page |
| Failure scenario | 1. The user enters the wrong course details 2. The user selects and adds the wrong course |
| Post condition | A new course has been added. |

**Successful adding course diagram:**

Diagram

Description automatically generated

**Delete Courses**

|  |  |
| --- | --- |
| Use case name | UC\_ Delete Course |
| Actors | Staff |
| Pre-Condition | Staff must login on system. |
| Flow of events | 1. Search Course 2. Select Course 3. Click Delete |
| Alternative flow | 1. If no course is selected then a send error message is sent 2. Go back to courses page |
| Failure scenario | 1. The user selects the wrong course 2. The user deletes the wrong course 3. The user deletes the wrong course and must now go to recover it |
| Post condition | Delete course successfully. |

**Successful course deletion diagram:**

Diagram

Description automatically generated