

UCS1617 - MINI PROJECT

Online Course Reservation System

USE CASE DOCUMENT

VERSION 1.0

SHASHANKA VENKATESH - 185001145

SHRI CHARAN RS - 185001147 SPARSH GUPTA - 185001160

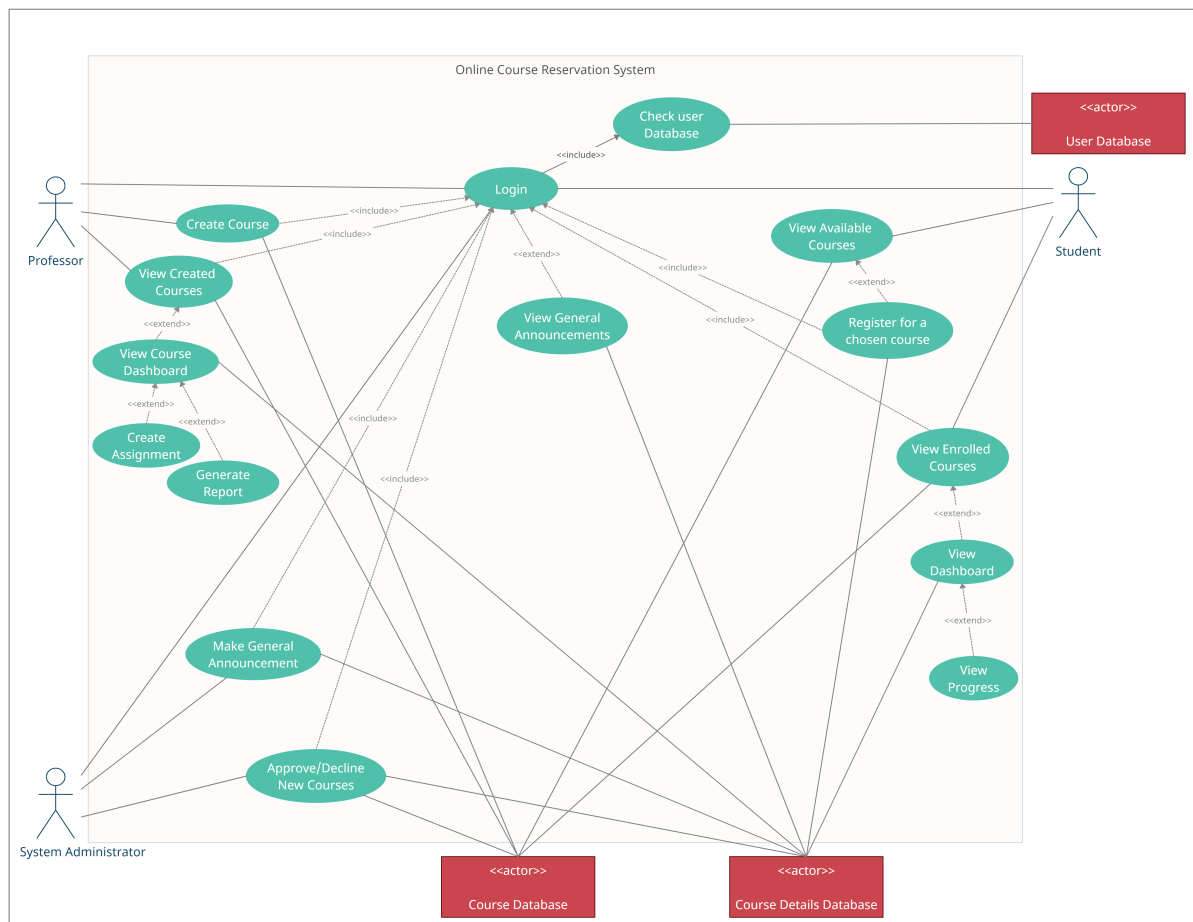
22nd February, 2021

Contents

1	Use Case Diagram	2
2	Use Case Descriptions	3
2.1	Login	3
2.2	Check User Database	4
2.3	View General Announcements	4
2.4	View Available Courses	5
2.5	Register for a Course	5
2.6	View Enrolled Courses	6
2.7	View Course Dashboard (Student)	6
2.8	View Progress	7
2.9	Create Course	7
2.10	View Created Courses	8
2.11	View Course Dashboard (Professor)	8
2.12	Create Assignment	9
2.13	Generate Report	9
2.14	Make a General Announcement	10
2.15	Approve/Decline New Course	10

Use Case Diagram

The use case diagram proposed for the Online Course Reservation System, based on the Software Requirements Specification (SRS) is shown below.



Use Case Descriptions

2.1 Login

- **Overview:** This use case describes how a student, professor and system admin logs into the Course Reservation System.
- **Primary Actors:** Student, Professor and System Administrator
- **Secondary Actors:** User Database
- **Basic Flow:** This use case starts when an actor wishes to log into the Course Reservation System.
 1. The system requests that the actor enter his/her name and password.
 2. The actor enters his/her name and password.
 3. The system validates the entered name and password and logs the actor into the system.
- **Alternative Flows:**
 1. Invalid Name / Password
If in the Basic Flow the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the Basic Flow or cancel the login, at which point the use case ends.
- **Pre-Conditions:** None
- **Post-Conditions:** If the use case was successful, the actor is now logged into the system. If not the system state is unchanged.
- **Includes:** Check User Database
- **Extends:** View General announcements.

2.2 Check User Database

- **Overview:** Use case to Check User database.
- **Basic Flow:** This use case starts when an actor wishes to log into the Course Reservation System.
 1. Give the User ID and Password
 2. Check if the user exists and the if password matches.
 3. Validate User entry
- **Alternative Flows:**
 1. Invalid User ID: Produce Error: Invalid User ID Code
 2. Incorrect Password: Produce Error: Incorrect Password
- **Pre-Conditions:** None
- **Post-Conditions:** User is validated to login if success, not validated if any error is produced.

2.3 View General Announcements

- **Overview:** Use case to View General announcements made for the entire University.
- **Primary Actors:** Student, Professor and System Administrator
- **Secondary Actors:** Course Details Page
- **Basic Flow:** This use case starts upon request from any of the primary actors after they log into the system.
 1. The system retrieves all general announcements from the Course Details Database and displays them.
- **Pre-Conditions:** The user must have logged into the system.
- **Post-Conditions:** None
- **Extends:** Login.

2.4 View Available Courses

- **Overview:** Use Case to view all available courses that the student can enroll for.
- **Primary Actor:** Student
- **Secondary Actor:** Course Database
- **Basic Flow:** This use case starts when an actor wishes to enroll in a new course.
 1. The system retrieves all available courses from the Course Database
 2. The system displays them to the student allowing them to search and sort the list
- **Pre-Conditions:** None
- **Post-Conditions:** None

2.5 Register for a Course

- **Overview:** This use case describes how a student registers for a chosen course.
- **Primary Actor:** Student
- **Secondary Actors:** Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to register for a course chosen from the available course list.
 1. The user chooses the course they want to enroll in.
 2. The system retrieves the Course details and asks for confirmation.
 3. Login to the system if not already logged in.
 4. The Student confirms enrollment.
 5. The system updates the Course Details database with the new student.
- **Alternative Flows:**
 1. The Student cancels enrollment.
 2. The System reverts back to displaying available courses.
- **Pre-Conditions:** Student must login to the system.
- **Post-Conditions:** If the use case was successful, the student is now enrolled in the system. If not the system state is unchanged.
- **Includes:** Login
- **Extends:** View Available Courses.

2.6 View Enrolled Courses

- **Overview:** Use Case to view all available courses that the student has already enrolled for.
- **Primary Actor:** Student
- **Secondary Actor:** Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to view all their enrolled courses.
 1. Login to the system if not already logged in.
 2. The system retrieves all courses that the student is enrolled in from the Course Details Database
 3. The system displays them to the student.
- **Pre-Conditions:** Student must login to the system.
- **Post-Conditions:** None
- **Includes:** Login.

2.7 View Course Dashboard (Student)

- **Overview:** Use Case to view the dashboard of a specific course the student has enrolled for.
- **Primary Actor:** Student
- **Secondary Actor:** Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to go to the Dashboard of a course they have enrolled in.
 1. The system retrieves information about the course and the student from the Course Details Database
 2. The system displays the information to the student.
- **Pre-Conditions:** Student must login to the system.
- **Post-Conditions:** None
- **Includes:** None
- **Extends:** View Enrolled Courses.

2.8 View Progress

- **Overview:** Use Case to view the progress of the student, i.e., marks, grades, attendance etc. in the chosen course.
- **Primary Actor:** Student
- **Secondary Actors:** Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to view their progress in a chosen course.
 1. The system retrieves information about the student's progress from the Course Details Database
 2. The system displays the information to the student.
- **Pre-Conditions:** Student must login to the system.
- **Post-Conditions:** None
- **Extends:** View Dashboard.

2.9 Create Course

- **Overview:** This use case describes how a professor can create a course for the upcoming semester.
- **Primary Actor:** Professor
- **Basic Flow:** This use case starts when an actor wishes to create a new course for the upcoming semester.
 1. The System requests the actor to enter the Name and syllabus of the new course.
 2. The System requests the actor to enter the intended semester when the students can take up the course
 3. The System creates a request to the system administrator to approve the course so that it can be added to the database
- **Pre-Conditions:** Professor must be logged into the system.
- **Post-Conditions:** A new course is sent for approval from the system administrator.
- **Includes:** Login

2.10 View Created Courses

- **Overview:** Use case for a professor to view all courses they have created, to choose which one to manage.
- **Primary Actor:** Professor
- **Secondary Actor:** Course Database
- **Basic Flow:** This use case starts when an actor wishes to view all the courses they have created in order to manage any of them.
 1. Gets & Validates the Professor ID.
 2. Displays the catalog of all the courses and handled by the Professor.
- **Pre-Conditions:** Professor must login to the system.
- **Post-Conditions:** None
- **Includes:** Login.

2.11 View Course Dashboard (Professor)

- **Overview:** Use Case to view the dashboard of a specific course the professor has created.
- **Primary Actor:** Professor
- **Secondary Actor:** Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to go to the Dashboard of a course they have created.
 1. The system retrieves information about the course from the Course Details Database
 2. The system displays the information to the professor.
- **Pre-Conditions:** Professor must login to the system.
- **Post-Conditions:** None
- **Includes:** None
- **Extends:** View Created Courses.

2.12 Create Assignment

- **Overview:** This use case describes how a professor can create an assignment for students taking up a specific course.
- **Primary Actor:** Professor
- **Secondary Actor:** Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to create a new assignment for the students of the chosen course.
 1. The System requests the actor to enter the Assignment Problem as a PDF.
 2. The System requests the actor to enter the deadline for the assignment.
 3. The System generates a post for the assignment given and posts it in the dashboard of the course
- **Pre-Conditions:** Professor must be logged into the system.
- **Post-Conditions:** A new assignment is issued and the details are posted in the dashboard of the course for all students to start working on it.
- **Includes:** Login
- **Extends:** View Course Dashboard (Professor).

2.13 Generate Report

- **Overview:** Use Case to view the report of a specific course the professor has created.
- **Primary Actor:** Professor
- **Secondary Actor:** Course Details Database
- **Basic Flow:** This use case starts when an actor wants to generate a report of all students that have taken the course.
 1. The System retrieves all information about the course from the Course Details Database.
 2. The System calculates several statistical values for the marks and grades obtained by the students in the course such as mean, median etc.
 3. The System generates a report and displays it to the actor.
- **Pre-Conditions:** Professor must be logged into the system.

- **Post-Conditions:** A report is generated and displayed for the Professor
- **Includes:** Login
- **Extends:** View Course Dashboard (Professor).

2.14 Make a General Announcement

- **Overview:** This use case describes how the system administrator can make a general announcement for the entire system.
- **Primary Actor:** System Administrator
- **Secondary Actor:** Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to make a new General Announcement.
 1. The system administrator writes the General Announcement
 2. The System creates a post and displays the announcement in the General Announcements page (which is treated as a course with every student enrolled in it, and under every professor)
- **Pre-Conditions:** System Administrator must be logged into the system.
- **Post-Conditions:** A new general announcement is made.
- **Includes:** Login

2.15 Approve/Decline New Course

- **Overview:** This use case describes how the system administrator can either approve or decline a new course created by the professor.
- **Primary Actor:** System Administrator
- **Secondary Actor:** Courses Database and Course Details Database
- **Basic Flow:** This use case starts when an actor wishes to approve or decline a course pending approval in the system.
 1. The System requests the system administrator to choose the pending course to take action for
 2. The System displays information of the course that the professor has entered.

3. The System Administrator approves the course
4. The System updates the Course Database and the Course Details Database with the new course.

- **Alternative Flows:**

1. The System Administrator declines the course
2. The Professor is notified that the System Administrator has declined the course, the Databases remain unchanged

- **Pre-Conditions:** System Administrator must be logged into the system, and there must be pending courses to approve/decline.

- **Post-Conditions:** If the use case was successful, the course is now added to the system and the Course Database and the Course Details Database are updated. If not the system state is unchanged.

- **Includes:** Login