Software Requirements Specification

for

ONLINE COURSE RESERVATION

Version 2.0 approved

Prepared by:

G. Pavitra - 22BD1A6728

K. Pranavi - 22BD1A6726

K. Sreeja - 22BD1A6728

G. Sujith - 22BD1A6718

Keshav Memorial Institute of Technology

24-09-2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Week-1	24-09-24	SRS creation (Introduction)	1.0
Week-2	26-09-24	SRS documentation- usecase , class diagarm	2.0

1. Introduction

1.1 Purpose

In today's fast-paced academic environment, students often face challenges in efficiently managing their course registrations. Traditional systems require manual effort and are time-consuming, leading to delays and errors in reserving courses. With increasing demands on flexibility, there is a need for a system that allows students to seamlessly browse available courses, register for them, and manage their schedules without the hassle of standing in lines or dealing with paper-based systems. The current system used by many institutions lacks integration with modern digital platforms, making it difficult for students to access courses on the go or in real-time.

1.2 Document Conventions

This document follows standard IEEE conventions for software documentation. Requirements are numbered sequentially with "REQ-" prefixes, and non-functional requirements are grouped separately.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, users, and testers of the Online Course Reservation System. It is structured to first provide an overview and then present detailed system features and requirements.

1.4 Product Scope

The Online Course Reservation System allows students to browse, reserve, and pay for courses online. Admins can manage course offerings and view reservations. This system is designed to improve the course booking experience and simplify management for educational institutions.

1.5 References

- IEEE Software Engineering Standards
- Web Design Style Guide
- API Documentation (for payment integration)

2. Overall Description

2.1 Product Perspective

The **Online Course Reservation System** is a new, self-contained platform designed to replace traditional manual course registration methods and enhance course management efficiency. While it operates independently, it can integrate with larger systems like Learning Management Systems (LMS) and payment services. The system provides real-time visibility into course availability, allowing students to make quick enrollment decisions and enabling instructors to create and manage courses without administrative assistance. Key components include an **Authentication Module** for user access, a **Course Management Module** for instructors, an **Enrollment Module** for student registrations, and a **Payment Module** for secure transactions. The product is built for scalability, offering a user-friendly web interface and secure backend database communication for storing user and course data.

2.2 Product Functions

- User registration and authentication
- Course browsing and reservation
- Payment processing
- Admin course management

2.3 User Classes and Characteristics

- **Students:** Users who browse and reserve courses.
- **Admins:** Users who manage courses and view reservations.
- **Guests:** Users who browse available courses without making reservations.

2.4 Operating Environment

The system will run on web browsers (Chrome, Firefox, Safari, and Edge). It is hosted on a cloud server, using a relational database like MySQL or PostgreSQL, and will integrate with third-party payment APIs.

2.5 Design and Implementation Constraints

- Payment gateway compliance with industry standards.
- Must follow institutional policies for data storage and privacy.

2.6 User Documentation

- User manual for students and admins.
- Online tutorials and FAQ for troubleshooting.

2.7 Assumptions and Dependencies

- Reliable internet connection.
- Integration with third-party APIs like payment gateways and institutional databases.

3. External Interface Requirements

3.1 User Interfaces

- **Student Dashboard:** Displays courses, reservation status, and payment history.
- **Admin Dashboard:** Displays all courses, enrolled students, and course management options.

3.2 Hardware Interfaces

No direct hardware interfaces, but the system must be compatible with common client hardware (laptops, desktops, smartphones).

3.3 Software Interfaces

- **Payment API:** Connects to third-party payment gateways (e.g., Stripe, PayPal).
- **Institutional Database:** For storing course and student information.

3.4 Communications Interfaces

- Email notifications for course reservations and payment confirmations.
- API calls between the system and external services.

4. System Features

4.1 User Registration and Authentication

Description and Priority:

Students must be able to create an account and log in to make course reservations. (High priority)

Stimulus/Response Sequences:

1. The student provides their email and password to register.
2. Upon successful registration, they are redirected to the dashboard.
3. For login, the system verifies the credentials and redirects the user accordingly.

Functional Requirements:

- REQ-1: The system must validate email addresses and passwords during registration.
- REQ-2: The system must support user login and logout functionality.

4.2 Course Search and Reservation

Description and Priority:

Students can search for available courses based on filters (e.g., subject, date) and reserve them. (High priority)

Stimulus/Response Sequences:

1. The student enters search criteria.
2. The system displays matching courses.
3. The student reserves a course and confirms payment.

Functional Requirements:

- REQ-3: The system must allow students to search and filter available courses.
- REQ-4: The system must allow students to reserve courses.
- REQ-5: The system must confirm the reservation via email.

4.3 Payment Integration

Description and Priority:

Students must be able to pay for course reservations securely. (High priority)

Stimulus/Response Sequences:

1. The student selects a course and proceeds to payment.
2. The system processes the payment via the payment gateway.
3. The system confirms payment and sends a receipt.

Functional Requirements:

- REQ-6: The system must securely process payments through external payment APIs.
- REQ-7: The system must confirm payments and update the reservation status.

4.4 Course Management (For Admins)

Description and Priority:

Admins must be able to add, update, or remove course listings. (Medium priority)

Functional Requirements:

- REQ-8: The system must allow admins to manage course details.
- REQ-9: The system must track student reservations for reporting purposes.

5. Other Non-Functional Requirements

5.1 Performance Requirements

- The system must handle up to 1000 concurrent users without performance degradation.

5.2 Safety Requirements

- Sensitive data, including personal and payment information, must be encrypted.

5.3 Security Requirements

- The system must implement role-based access control (RBAC) to protect admin functionalities.

5.4 Software Quality Attributes

- The system must be user-friendly, responsive, and support multi-device access.

5.5 Business Rules

- Only registered users can reserve courses.
- Admins can modify course listings.

6. Other Requirements

- The system should be scalable to accommodate additional institutions.

Appendices

Appendix A: Glossary

- **Reservation:** Booking of a course by a student.
- **Admin:** User responsible for managing course content.

Appendix B: Analysis Models

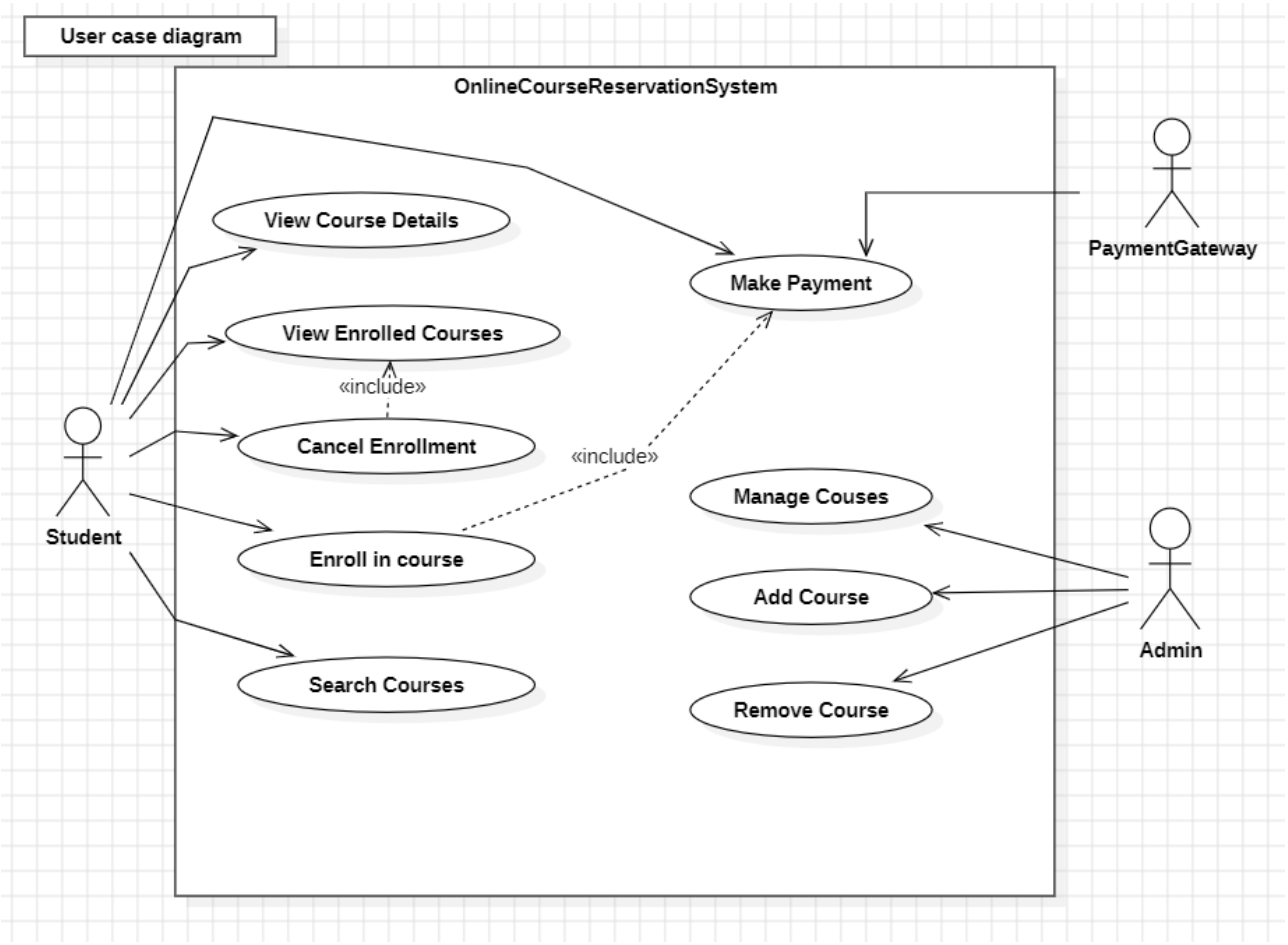
- Data flow diagrams and system architecture diagrams are provided separately.

Use case template

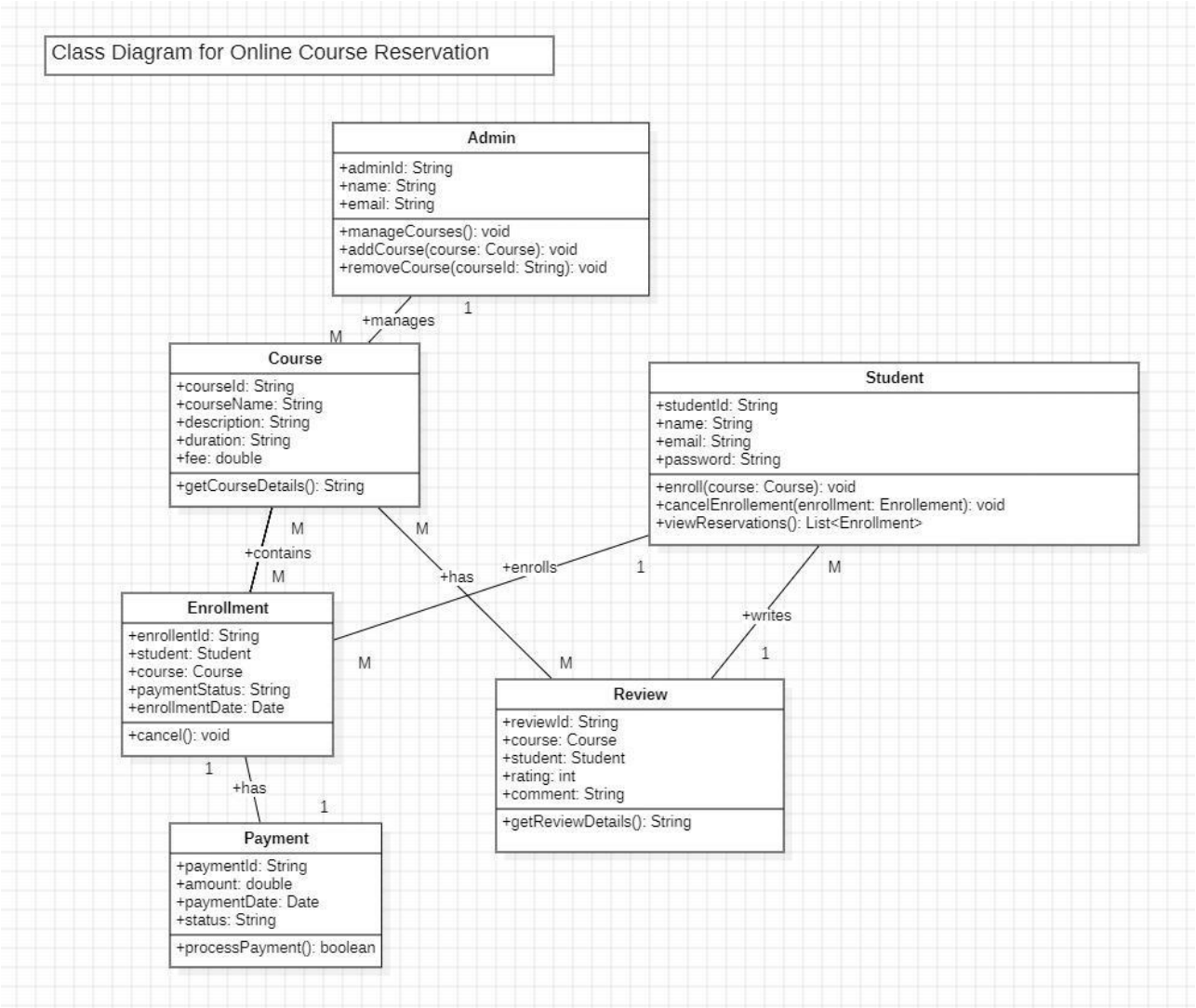
Use Case ID:	1018202402		
Use Case Name:	Online Course Reservation		
End Objective:	Enable students to search, enroll in, and manage course reservations efficiently, while allowing administrators to manage courses.		
Created by:	1. Sreeja 2. Pavitra 3. Pranavi 4. Sujith	**On (date):**	October 18,2024
User/Actor:	Student, Admin, Payment Gateway		
Trigger:	Student or Admin interaction with the system for course enrolment or management.		

Basic/Normal Flows	
User Actions	System Actions
Exception Flows	
User Action	**System Action**
The user attempts to log in by entering their credentials. The user tries to log in with incorrect credentials. The user attempts to register for a course.	The system prompts the user to register on the signup page before attempting to log in. The system displays a message: "Please check the username or password entered" and asks the user to try again. If the course is full, the system displays: "Course full" for the respective course. If payment is not processed, the system prompts the user to retry or choose a different payment method.

Use Case Diagram:



Class diagram:



-

- # mAppendix C: To Be Determined List

Here's a list of items that still need to be decided or clarified for the **Online Course Reservation System**:

1. **Payment Gateway**:
    - TBD: Which payment service will we use (like Stripe or PayPal)?
2. **User Interface Designs**:
    - TBD: Final designs and layouts for the user interface need to be created.
3. **Supported Languages**:
    - TBD: Which languages will the system support initially?
4. **Regulatory Compliance**:
    - TBD: What specific regulations do we need to follow (like GDPR)?
5. **User Testing Plan**:
    - TBD: How will we conduct user acceptance testing before launching?
6. **Hosting Details**:
    - TBD: Where will we host the application (cloud provider or local server)?
7. **Backup Plan**:
    - TBD: What will our backup and recovery procedures be in case of data loss?

This list helps keep track of important decisions that still need to be made as we work on the project.