

# Software Requirement Specifications

## Student Management System: FAST FLYERS

*Version: 001*

Project Code	–
Supervisor	Ms. Rubab Jaffer
Co Supervisor	TA
Project Team	Alizah Basit      22K-5155 Amna Mansoor      22K-5159 Hafsa Salman      22K-5161
Submission Date	23 March, 2024

## Instructions

- No section of template should be deleted. You can write 'Not applicable' if a section is not applicable to your project. But all sections must exist in the final document.
- All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in the final document.
- This 'Instruction' section should also be removed in the final document.
- MS-Word Reviewing feature must be used to get the document reviewed by supervisors or co-supervisors.

## Document History

Version	Name of Person	Date	Description of change
01	Amna Mansoor	10/03/2024	Document created
02	Hafsa Salman	10/03/2024	Document edited
03	Alizah Basit	16/03/2024	Document edited
04	Hafsa Salman	18/03/2024	Document edited
05	Amna Mansoor	20/03/2024	Document edited
06	Alizah Basit	21/03/2024	Document edited
07	Hafsa Salman	22/03/2024	Document edited
08	Alizah Basit	22/03/2024	Document edited
09	Hafsa Salman	22/03/2024	Document edited
10	Amna Mansoor	23/03/2024	Document edited
11	Hafsa Salman	23/03/2024	Document edited
12	Alizah Basit	23/03/2024	Document edited

## Distribution List

Name	Role
Ms. Rubab Jaffar	Supervisor
TA	Co- Supervisor

## Document Sign-Off

Version	Sign-off Authority	Sign-off Date

# Table of Contents

1. INTRODUCTION	8
1.1. Purpose of Document.....	8
1.2. Intended Audience.....	8
1.3. Abbreviations .....	8
1.4. Document Convention.....	8
2. OVERALL SYSTEM DESCRIPTION	9
2.1. Project Background.....	9
2.2. Project Scope.....	9
2.3. Not In Scope.....	9
2.4. Project Objectives.....	9
2.5. Stakeholders.....	9
2.6. Operating Environment.....	9
2.7. System Constraints.....	10
2.8. Assumptions & Dependencies.....	10
3. EXTERNAL INTERFACE REQUIREMENTS	11
3.1. Hardware Interfaces.....	11
3.2. Software Interfaces.....	11
3.3. Communications Interfaces.....	11
4. FUNCTIONAL REQUIREMENTS	11
4.1. FUNCTIONAL HIERARCHY	11
4.1.1. User Authentication and Authorization(Admin).....	11
4.1.2. Course Registration.....	12
4.1.3. Attendance.....	12
4.1.4. Marks.....	13
4.1.5. Grades and Transcript.....	13
4.1.6. Fee Generation.....	13
4.2. Use Cases	14
4.2.1. Login (Admin).....	16
4.2.2. Add Course (Admin).....	16
4.2.3. Add Student (Admin).....	17
4.2.4. Add Teacher (Admin).....	18
4.2.5. Edit Course (Admin).....	18
4.2.6. Assign Course toTeacher (Admin).....	19
4.2.7. View Course List (Admin).....	20
4.2.8. View Student List (Admin).....	21
4.2.9. View Teacher List (Admin).....	21
4.2.10. Remove Courses (Admin).....	22
4.2.11. View Fee Structures (Admin).....	23
4.2.12. Edit Fees (Admin).....	24
4.2.13. Fees Summary (Admin).....	24

4.2.14. Log out (Admin).....	25
4.2.15. Login (Teacher).....	26
4.2.16. View Assigned Course (Teacher).....	26
4.2.17. View Student (Teacher).....	27
4.2.18. Mark Attendance (Teacher).....	28
4.2.19. View Attendance (Teacher).....	29
4.2.20. View Grade (Teacher).....	29
4.2.21. Mark Student (Teacher).....	30
4.2.22. Get Personal Information (Teacher).....	31
4.2.23. Log Out (Teacher).....	31
4.2.24. Login (Student).....	32
4.2.25. View Attendance (Student).....	33
4.2.26. View Registered Courses (Student).....	33
4.2.27. Add Course (Student).....	34
4.2.28. Drop Course (Student).....	35
4.2.29. Transcript (Student).....	36
4.2.30. Pay Fees (Student).....	36
4.2.31. Fees Summary (Student).....	37
4.2.32. Log Out (Student).....	38
5. <i>NON-FUNCTIONAL REQUIREMENTS</i> .....	38
5.1. Performance Requirements.....	39
5.2. Safety Requirements.....	39
5.3. Security Requirements.....	39
5.4. User Documentation.....	39
6. <i>REFERENCES</i> .....	40
7. <i>APPENDICES</i> .....	41

# **1. Introduction**

## **1.1. Purpose of Document**

The main purpose of this document is to illustrate the software requirements for the Student Management System (SMS). The document gives the scope as well as detailed description of both functional and non-functional requirements of the system. This document aims to provide an overview of this product including the system requirements and features. This document will provide all the necessary details to all the stakeholders and major users of the system.

## **1.2. Intended Audience**

This document is designed for the Course Instructor for Software Design and Architecture (SDA): Miss Rubab Jaffar, and the developers of the System, stakeholders , administration and faculty of FAST NUCES, Karachi.

## **1.3. Abbreviations**

- API: Application Programming Interface
- DBMS: Database Management System
- FAST: Foundation for Advancement of Science and Technology
- GPA: Grade Point Average
- GUI: Graphical User Interface
- HTTP: Hypertext Transfer Protocol
- IDE: Integrated Development Environment
- I/O: Input/Output
- IOS: iPhone Operating System
- JDK: Java Development Kit
- JVM: Java Virtual Machine
- NUCES: National University of Computer and Emerging Sciences
- OS: Operating System
- RDBMS: Relational Database Management System
- SDA: Software Design and Architecture
- SMS: Student Management System
- SQL: Structured Query Language
- SRS: Software Requirements Specification
- WIFI: Wireless Fidelity

## **1.4 Document Convention**

- Convention for Main Heading:  
Font Face: Arial  
Font Style: Bold and Italic  
Font Size: 16
- Convention for Subheading:  
Font Face: Arial  
Font Style: Bold and Italic  
Font Size: 12



- Convention for Body:

Font Face: Arial

Font Style: None

Font Size: 11

## **2. Overall System Description**

### **2.1. Project Background**

The Student Management System is designed to meet the educational needs within the students, the faculty, and the administrators of the FAST (NUCES). The system will make the Student Management more efficient and accessible.

### **2.2. Project Scope**

The project will deliver a fully functional student management system encompassing modules for course registration, attendance tracking, grade management, fee generation, feedback collection, and transcript generation.

### **2.3. Not In Scope**

The student management system will not include integration with external systems or advanced analytics features beyond the scope defined in the Software Requirements Specification.

### **2.4. Project Objectives**

The project objective is to improve the accessibility of the system for the students as well as for the faculty. It will bring everyone on the same platform. It will improve communication between students and the faculty.

### **2.5. Stakeholders**

- Students
- Administrators
- Teachers
- Development Team
- Testers
- Design Team
- Software Design Team

### **2.6. Operating Environment**

- The server-side components of the software system must operate within the windows operating system, a better environment is achieved with windows 7, 8, 10 or 11. The software will also operate efficiently with the receipt generator and a credit card machine. The user-side components of the software system must operate within the latest stable release of android and IOS as well as common web-browser environments.
- The minimum set of androids and IOS that must support this app are:
  - Android 9 and above
  - IOS 10 and above
- The minimum set of browsers that must be compatible with system are:
  - Internet Explorer, Safari, and Chrome
- Whole system on the user side should be connected to WIFI.
- Language used: **JAVA**

## 2.7. System Constraints

- Software constraints
  - The system should be developed using JAVA programming language.
  - GUI frameworks such as JAVA Swing will be utilized for developing the user interface.
  - Object-oriented features of JAVA will be used to ensure a modular approach to development
- Hardware constraints
  - The software should be compatible with standard computing hardware configurations, including processors, memory, and storage requirements.
  - Minimum system requirements for running the software will be specified to ensure optimal performance on various hardware setups.
- Cultural constraints (includes language etc.)
  - The project will be implemented using the English language, system wide.
  - User interfaces, messages, and documentation should be culturally sensitive and avoid any language or imagery that may be offensive or insensitive to specific cultural groups.
- Legal constraints
  - The project must adhere to relevant copyright laws and licensing agreements for any third-party software components used.
  - The system should not facilitate any illegal activities or violate any local or international laws.
- Environmental constraints
  - The software should be designed to operate in standard office or educational environments.
  - Preferences of the target user group, such as teachers, administrators, and students, should be considered during the design phase to ensure usability.
- User constraints
  - The user interface should be intuitive and user-friendly to accommodate users with varying levels of technical expertise.
- Off the shelf components
  - Constraints of any third-party libraries, frameworks, or tools used in the project should be considered and documented.

## 2.8. Assumptions & Dependencies

It is assumed that:

- Teachers and administrators will be trained to operate the system.
- System will work efficiently with Windows OS and Database.

System have several dependencies:

- GUI is in English only.
- System access will only be granted if the user provides a valid username and password.

Incase of any potential change in assumed facts or system dependencies, SRS shall be flexible enough to adjust accordingly.

### **3. External Interface Requirements**

#### **3.1. Hardware Interfaces**

The system does not have any direct hardware interfaces. It operates solely as a software application and interacts with the underlying hardware through the operating system and Java Virtual Machine (JVM).

#### **3.2. Software Interfaces**

The system interacts with various external software components for its operation:

- Java Development Kit (JDK): The system is developed using Java programming language and requires JDK version 8 or later for compilation and execution.
- Operating System (OS): The system is compatible with any operating system that supports Java, including Windows, macOS, and Linux distributions.
- Database Management System (DBMS): Although not directly integrated, the system is designed to work with any relational database management system (RDBMS) such as MySQL, PostgreSQL, or SQLite for persistent storage of data.
- Integrated Development Environment (IDE): The system can be developed, maintained, and executed using popular IDEs such as IntelliJ IDEA, Eclipse, or NetBeans.
- Java Libraries: The system utilizes standard Java libraries for various functionalities such as file I/O, serialization, and user input/output.

#### **3.3. Communications Interfaces**

The system does not require direct communication interfaces for its core functionality. However, it may utilize standard communication protocols such as HTTP for web-based functionalities if integrated with web services or APIs.

### **4. Functional Requirements**

#### **4.1. Functional Hierarchy**

##### **4.1.1. User Authentication and Authorization**

- **Description and Priority**

This feature ensures secure access to the system with role-based permissions for different types of users i.e students, teachers, administration, etc. This feature has high priority.

- **Stimulus/Response Sequences**

Users are first asked to authenticate whether they are a student, teacher or admin. When the student, teacher or admin attempts to Log-in, giving correct credentials, the system verifies credentials and grants access based on the user's role. If a student/teacher/admin enters wrong credentials, an error message would be shown.

- **Functional Requirements**

REQ-1: The system should implement secure user authentication mechanisms.

REQ-2: Role-specific access should be granted in the system.

REQ-3: Users should be prompted to re-authenticate when engaging in sensitive operations.

#### 4.1.2. **Course Registration**

- **Description and Priority**

This feature solely focuses on the offered courses and their registration. Every new semester, teachers and students have to register for courses they want to teach and study respectively. This module is a high priority.

- **Stimulus/Response Sequences**

Admin is responsible for activating the course registration for the teachers and the students and providing the offered courses list in the following semester. The teachers have to go to the course registration page and select the courses they want to teach in the following semester. And for the students, they can register themselves for the courses they want to study. The students can also drop a course until and unless the registration is activated. If a student or a teacher is facing a problem registering courses, then the admin can register them in their desired courses.

- **Functional Requirements**

REQ-4: The system shall provide a function to list all available courses.

REQ-5: The system shall provide a function to admin to activate the course registration.

REQ-6: The system shall allow teachers to register themselves for the courses they want to teach.

REQ-7: The system shall allow students to add or drop a course.

REQ-8: The system shall allow admin to register students' and teachers' courses.

REQ-9: The system shall allow teachers to generate the list of students who have enrolled themselves in that particular course.

REQ-10: The system shall allow admin to generate the list of courses, enrolled students in the particular course and the teacher who is teaching the particular course.

#### 4.1.3. **Attendance**

- **Description and Priority**

This section is responsible for the attendance of the students. This is high-priority.

- **Stimulus/Response Sequences**

Upon the successful completion of the courses registration, students are bound to have 80% attendance to sit in their finals. To keep a track on their attendance, the student can view their attendance on a daily basis or whenever they log in to the system. Only teachers can update and edit the attendance.

- **Functional Requirements**

REQ-11: The system shall allow teachers to update students' attendance.

REQ-12: The system shall allow students to view their attendance.

**4.1.4. Marks**

- **Description and Priority**

This section is responsible for the marks of the students. This is high-priority.

- **Stimulus/Response Sequences**

Throughout the semester, students give quizzes, submit assignments and give exams. Teachers upload their marks on the SMS. After uploading, students can view their marks.

- **Functional Requirements**

REQ-13: The system shall allow teachers to update students' marks.

REQ-12: The system shall allow students to view their marks throughout the semester.

**4.1.5. Grades and Transcript**

- **Description and Priority**

This functionality is related to the grades and viewing transcript. Only students can view their own grades and transcript. No other student can look at other students' grades and transcript. A teacher can view the grades of the students for the course they are teaching. It is a high-priority.

- **Stimulus/Response Sequences**

After final exams, teachers upload the marks on the SMS. They can edit too but once the marks are finalized they are locked in. Once they are locked in, the system locks the marks and puts the grade on the transcript. Students can view their grades on the transcript.

- **Functional Requirements**

REQ-14: The system shall allow teachers to update students' marks.

REQ-15: The system shall put the grade on the transcript.

REQ-16: The system shall allow students to view their grades and transcript.

REQ-17: The system shall allow admin to look for students' transcripts.

**4.1.6. Fee Generation**

- **Description and Priority**

This module is related to the fees generation. Fees is calculated through the number of credit hours a student has opted for in the particular semester.

- **Stimulus/Response Sequences**

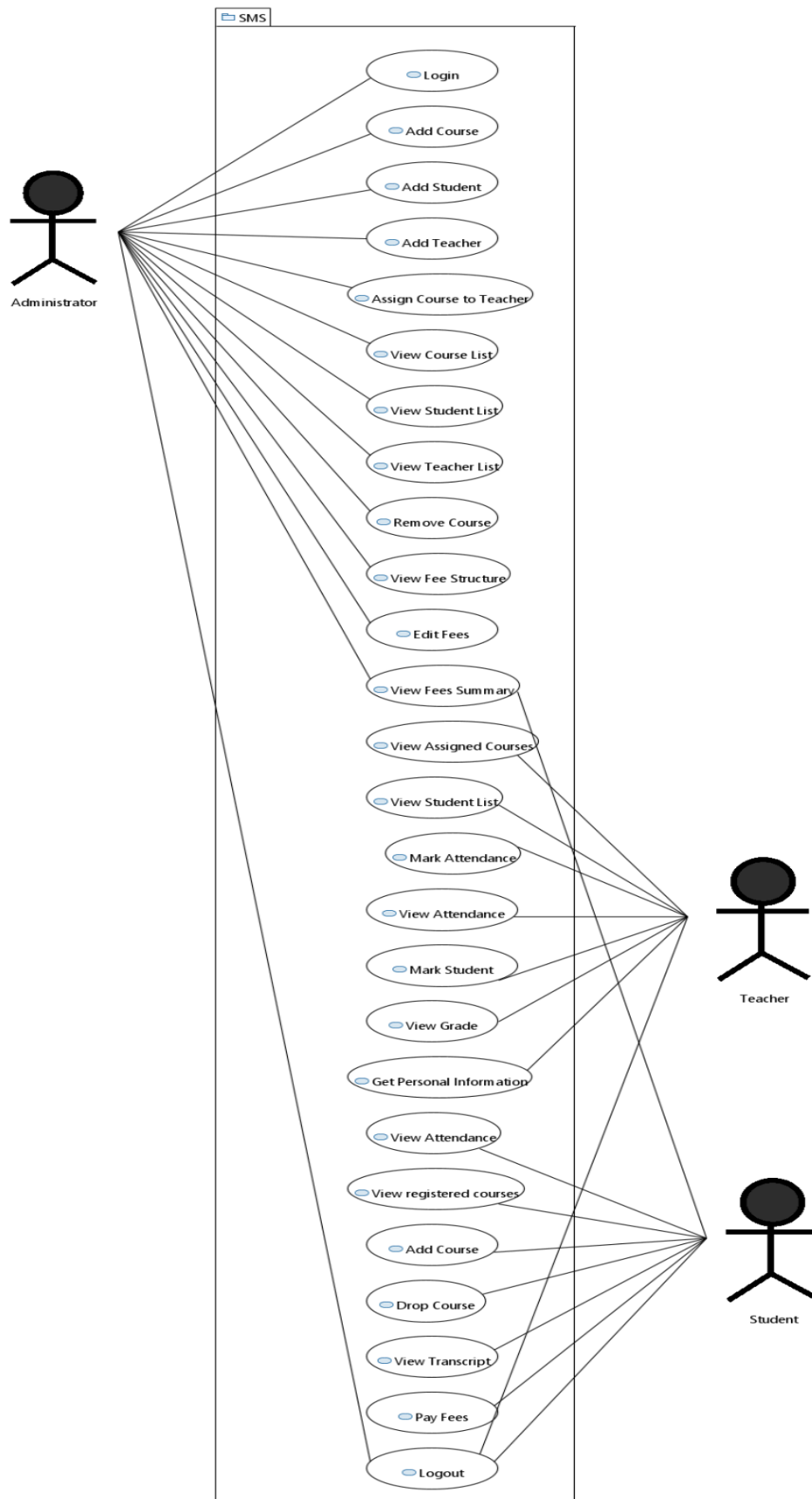
Once the courses are registered, the system takes the registered credit hours and multiplies it with the assigned fee for 1 credit hour which is set by the Admin.

- **Functional Requirements**

REQ-18: The system shall generate fee slips.

## **4.2. Use Cases**

**Use case diagram:**



**4.2.1. Login (Admin)**

SMS-001: Log in		
Use case Id:	SMS-001	
Actors:	Administrator	
Feature:	User Authentication and Authorization	
Pre-condition:	Person should be part of the administration part of FAST NUCES.	
Scenarios		
Step #	Action	Software Reaction
1.	User enters username and password	System verifies the credentials and the administrator is logged in.
Alternate Scenarios:		
1a: If the entered username or password is incorrect, the system prompts the user to re-enter the credentials		
Post Conditions		
Step #	Description	
1.	The user is successfully logged into the system.	
Use Case Cross referenced	None.	

**4.2.2. Add Course (Admin)**

<b>SMS-002: Add Course</b>	
<b>Use case Id:</b>	SMS-002
<b>Actors:</b>	Administrator
<b>Feature:</b>	Course Management
<b>Pre-condition:</b>	User must be logged into the system
<b>Scenarios</b>	



Step #	Action	Software Reaction
1.	Admin navigates to the "Add Course" section.	System retrieves and asks the user to add a course.
<b>Alternate Scenarios:</b>		
1a: System displays the messages that the course already exists.		
<b>Post Conditions</b>		
Step #	Description	
1.	The course has been successfully added by the admin.	
<b>Use Case Cross referenced</b>		None

#### 4.2.3. Add Student (Admin)

SMS-003: Add Student		
Use case Id:	SMS-003	
Actors:	Administrator	
Feature:	Student Management	
Pre-condition:	The admin must be logged into the system.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigates to the “Add Student” section.	System retrieves and asks the user to add a student.
Alternate Scenarios:		
1a: System displays the messages that the student already exists.		
Post Conditions		
Step #	Description	
1.	The student has been successfully added by the admin.	

<b>Use Case Cross referenced</b>	None
----------------------------------	------

#### 4.2.4. **Add Teacher (Admin)**

SMS-004: Add Teacher		
Use case Id:	SMS-004	
Actors:	Administrator	
Feature:	Teacher Management	
Pre-condition:	The admin must be logged into the system.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigates to the “Add Teacher” section.	System retrieves and asks the user to add a teacher.
Alternate Scenarios:		
1a: System displays the messages that the teacher already exist.		
Post Conditions		
Step #	Description	
1.	The teacher has been successfully added by the admin.	
Use Case Cross referenced	None	

#### 4.2.5. **Edit Course (Admin)**

<b>SMS-005: Edit Course</b>	
<b>Use case Id:</b>	SMS-005
<b>Actors:</b>	Administrator
<b>Feature:</b>	Course Management

<b>Pre-condition:</b>		The admin must be logged into the system and the course must exist in the system.
<b>Scenarios</b>		
<b>Step #</b>	<b>Action</b>	<b>Software Reaction</b>
1.	Admin navigates to the "Edit course" section	System retrieves and asks the admin to enter the course they want to edit.
2.	Admin enters the course code of the existing course	The system searches for the course and displays it on the screen.
<b>Alternate Scenarios:</b>		
1a: The course does not exist, and the system displays an appropriate message.		
<b>Post Conditions</b>		
<b>Step #</b>	<b>Description</b>	
1.	The course has been successfully edited.	
<b>Use Case Cross referenced</b>		None

#### 4.2.6. Assign Course to Teacher (Admin)

SMS-006: Assign Course to Teacher		
Use case Id:	SMS-006	
Actors:	Administrator	
Feature:	Course and Teacher Management	
Pre-condition:	Admin must be logged in to the system and the course and the teacher must exist.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigated to the “Assign course to Teacher” section	System retrieves and asks to enter the course code.
2.	Admin enters course code.	System, then asks to enter the teacher's name.

3.	Admin enters the teacher's name.	System finds the teacher and assigns them with the course.
<b>Alternate Scenarios:</b>		
1a: The particular course does not exist, and the system displays an appropriate message.		
2a: The particular teacher does not exist, and the system displays an appropriate message.		
<b>Post Conditions</b>		
<b>Step #</b>	<b>Description</b>	
1.	The teacher has been assigned the course successfully.	
<b>Use Case Cross referenced</b>		None

#### 4.2.7. View Course List (Admin)

SMS-007: View Course List		
Use case Id:	SMS-007	
Actors:	Administrator	
Feature:	Course Management	
Pre-condition:	The admin must be logged into the system. The courses must exist.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigates to the “View Course List” section.	System retrieves and displays the list of the courses.
Alternate Scenarios:.		
1a: There are no courses, and the system displays the list empty.		
Post Conditions		
Step #	Description	
1.	The system displays the list of the courses.	

<b>Use Case Cross referenced</b>	None
----------------------------------	------

#### 4.2.8. **View Student List (Admin)**

SMS-008: View Student List		
Use case Id:	SMS-008	
Actors:	Administrator	
Feature:	Student Management	
Pre-condition:	The admin must be logged into the system and the student must exist in the system.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigates to the “View Student List” section.	System retrieves and displays the list of the courses.
Alternate Scenarios:		
1a: There are no students, and the system displays the list empty.		
Post Conditions		
Step #	Description	
1.	The system displays the list of the students.	
Use Case Cross referenced	None	

#### 4.2.9. **View Teacher List (Admin)**

<b>Use Case ID Name: Log in</b>	
<b>Use case Id:</b>	SMS-009
<b>Actors:</b>	Administrator
<b>Feature:</b>	Teacher Management

<b>Pre-condition:</b>		The admin must be logged into the system. The teacher must exist.
<b>Scenarios</b>		
<b>Step #</b>	<b>Action</b>	<b>Software Reaction</b>
1.	Admin navigates to the “View Teachers List” section.	System retrieves and displays the list of the teachers.
<b>Alternate Scenarios:</b>		
1a: There are no teachers, and the system displays the list empty.		
<b>Post Conditions</b>		
<b>Step #</b>	<b>Description</b>	
1.	The system displays the list of the teachers.	
<b>Use Case Cross referenced</b>		None

#### 4.2.10. Remove Courses (Admin)

SMS-010: Remove Courses		
Use case Id:	SMS-010	
Actors:	Administrator	
Feature:	Course Management	
Pre-condition:	The admin must be logged into the system and the course must exist.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigates to the “Remove Course” section.	System retrieves and asks the admin to enter the course code.
2.	The admin enters the course code.	System removes the course.
Alternate Scenarios:		

<b>1a:</b> The course does not exist, and the system displays an appropriate message.	
<b>Post Conditions</b>	
<b>Step #</b>	<b>Description</b>
1.	The course has been successfully removed.
<b>Use Case Cross referenced</b>	None

#### 4.2.11. View Fee Structures (Admin)

SMS-011: View Fee Structures		
Use case Id:	SMS-011	
Actors:	Administrator	
Feature:	Fee Management	
Pre-condition:	The admin must be logged in to the system and the courses must exist.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigates to the “View Fee Structure” section.	System retrieves and displays the fees.
Alternate Scenarios:		
1a: The course does not exist since the fees depend on the credit hours, and the system displays an empty Fee Structure.		
Post Conditions		
Step #	Description	
1.	The fee structure list has been displayed by the system.	
Use Case Cross referenced	None	

**4.2.12. Edit Fees (Admin)**

SMS-012: Edit Fees		
Use case Id:	SMS-012	
Actors:	Administrator	
Feature:	Fees Management	
Pre-condition:	The admin must be logged in to the system and the courses must exist.	
Scenarios		
Step #	Action	Software Reaction
1.	Admin navigates to the “Edit Fees” section.	System retrieves and asks the user to enter the course code.
2.	Admin enters the course code.	System asks the user to update the fees.
3.	Admin enters the updated fees.	System updates the fees.
Alternate Scenarios:.		
1a: The particular course does not exist.		
Post Conditions		
Step #	Description	
1.	The fees have been updated successfully.	
Use Case Cross referenced	None	

**4.2.13. Fees Summary (Admin)**

<b>SMS-013: Fees Summary</b>	
<b>Use case Id:</b>	SMS-013
<b>Actors:</b>	Administrator
<b>Feature:</b>	Fees Management



<b>Pre-condition:</b>		The admin must be logged into the system. The course
<b>Scenarios</b>		
<b>Step #</b>	<b>Action</b>	<b>Software Reaction</b>
1.	Admin navigates to the “View Fees Summary” section	System retrieves and displays the fees summary.
<b>Alternate Scenarios:</b>		
1a: The fees for the courses do not exists.		
<b>Post Conditions</b>		
<b>Step #</b>	<b>Description</b>	
1.	The fees summary has been successfully displayed.	
<b>Use Case Cross referenced</b>		None

#### 4.2.14. Logout (Admin)

SMS-014 : Logout		
Use case Id:	SMS-014	
Actors:	Admin	
Feature:	User Authentication and Authorization	
Pre-condition:	The Admin must be logged into the system	
Scenarios		
Step#	Action	Software Reaction
1.	Admin selects the “Logout” option.	System logs the admin out and redirects them to the login page.
Alternate Scenarios:		
1a: If any operation is in-progress, the system doesn't logout the admin until the operation is completed.		
Post Conditions		

SMS-014 : Logout	
Step#	Description
1	The admin is logged out of the system.
Use Case Cross referenced	None

#### 4.2.15. Login (Teacher)

SMS-015 : Login		
Use case Id:	SMS-015	
Actors:	Teacher	
Feature:	User Authentication and Authorization	
Pre-condition:	Teacher has been employed by the administration	
Scenarios		
Step #	Action	Software Reaction
1.	User enters username and password	System verifies the credentials and the teacher is logged in.
Alternate Scenarios:		
1a: If the entered username or password is incorrect, the system prompts the user to re-enter the credentials.		
Post Conditions		
Step#	Description	
1	The user is successfully logged into the system.	
Use Case Cross referenced	None	

#### 4.2.16. View Assigned Course (Teacher)

SMS-016 : View Assigned Course		
Use case Id:	SMS-016	
Actors:	Teacher	
Feature:	Course Management	
Pre-condition:	The teacher must be logged into the system.	
Scenarios		
Step#	Action	Software Reaction
1.	Teacher navigates to the "View Assigned Course" section.	System retrieves and displays the list of courses assigned to the teacher.
Alternate Scenarios:.		
1a: System doesn't show any course because the teacher has not been assigned any course by the administration yet.		
Post Conditions		
Step #	Description	
1	The teacher can view the list of courses they are assigned to teach.	
Use Case Cross referenced	None	

#### 4.2.17. View Student (Teacher)

SMS-017: View Student List		
Use case Id:	SMS-017	
Actors:	Teacher	
Feature:	Student Management	
Pre-condition:	The teacher must be logged into the system.	
Scenarios		
Step#	Action	Software Reaction

1.	Teacher selects a course.	System retrieves and displays the list of students enrolled in the selected course.
<b>Alternate Scenarios:</b>		
1a: System doesn't show any students because no student has registered for a course yet.		
<b>Post Conditions</b>		
<b>Step#</b>	<b>Description</b>	
	The teacher can view the list of students enrolled in the selected course.	
<b>Use Case Cross referenced</b>		None

#### 4.2.18. Mark Attendance (Teacher)

SMS-018: Mark Attendance		
Use case Id:	SMS-018	
Actors:	Teacher, Student	
Feature:	Attendance Management	
Pre-condition:	The teacher must be logged into the system.	
Scenarios		
Step#	Action	Software Reaction
1.	Teacher selects a course and a date for which attendance needs to be marked.	System presents a list of students enrolled in the selected course.
2.	Teacher marks student for each student (present or absent)	System saves the attendance sheet
Alternate Scenarios:		
1a: System doesn't show any students because no student has registered for a course yet.		
Post Conditions		
Step#	Description	

1	The attendance for the selected date is marked for the chosen course.
Use Case Cross referenced	None

#### 4.2.19. View Attendance (Teacher)

SMS-019 : View Attendance		
Use case Id:	SMS-019	
Actors:	Teacher, Student	
Feature:	Attendance Management	
Pre-condition:	The teacher must be logged into the system.	
Scenarios		
Step#	Action	Software Reaction
1.	Teacher selects a course to view the attendance.	System presents the list of students with their attendance.
Alternate Scenarios:		
1a: System doesn't show any students because no student has registered for a course yet.		
Post Conditions		
Step#	Description	
1	The teacher can view the student's attendance records.	
Use Case Cross referenced	None	

#### 4.2.20. View Grade (Teacher)

SMS-020 : View Grade	
Use case Id:	SMS-020
Actors:	Teacher, Student
Feature:	Grade Management
Pre-condition:	The teacher must be logged into the system

Scenarios		
Step#	Action	Software Reaction
1.	Teacher navigates to the "View Grade" section.	System retrieves and displays the grades of students enrolled in the particular course taught by the teacher
<b>Alternate Scenarios:</b>		
1a: System doesn't show any grade because the teacher hasn't marked any grade yet.		
<b>Post Conditions</b>		
Step#	Description	
1	The teacher can view the grades of students enrolled in the courses they teach.	
<b>Use Case Cross referenced</b>		None

#### 4.2.21. Mark Student (Teacher)

SMS-021: Mark Student		
Use case Id:	SMS-021	
Actors:	Teacher, Student	
Feature:	Grade Management	
Pre-condition:	The teacher must be logged into the system	
Scenarios		
Step#	Action	Software Reaction
1.	Teacher selects a course and a student	System presents the grading interface
2.	Teacher inputs or modifies the student's grade for the selected course.	System presents the students with their marks.
Alternate Scenarios:		
1a: Students are not registered for the course yet.		
2a: Grading period has not started yet.		

Post Conditions	
Step#	Description
1	The student's grade for the selected course is updated.
Use Case Cross referenced	None

#### 4.2.22. Get Personal Information (Teacher)

SMS-022: Get Personal Information		
Use case Id:	SMS-022	
Actors:	Teacher	
Feature:	Information Management	
Pre-condition:	The teacher must be logged into the system	
Scenarios		
Step#	Action	Software Reaction
1.	User navigates to the "Personal Information" section.	System retrieves and displays the user's personal information (name, age, salary, etc.).
Alternate Scenarios:		
1a: The teacher information is not approved by the administration yet.		
Post Conditions		
Step#	Description	
1	The teacher can view their personal information.	
Use Case Cross referenced	None	

#### 4.2.23. Log Out (Teacher)

SMS-023 : Logout	
Use case Id:	SMS-023.
Actors:	Teacher

<b>Feature:</b> User Authentication and Authorization		
<b>Pre-condition:</b>		The teacher must be logged into the system
<b>Scenarios</b>		
<b>Step#</b>	<b>Action</b>	<b>Software Reaction</b>
1.	Teacher selects the "Logout" option.	System logs the teacher out and redirects them to the login page.
<b>Alternate Scenarios:</b>		
1a: If any operation is in-progress, the system doesn't logout the teacher until the operation is completed.		
<b>Post Conditions</b>		
<b>Step#</b>	<b>Description</b>	
1	The user is logged out of the system.	
<b>Use Case Cross referenced</b>		None

#### 4.2.24. Log In (Student)

<b>SMS-024: Log in</b>		
<b>Use case Id:</b>		SMS-024
<b>Actors:</b> Student		
<b>Feature:</b> User Authentication and Authorization		
<b>Pre-condition:</b>		Student has been registered by the admin
<b>Scenarios</b>		
<b>Step #</b>	<b>Action</b>	<b>Software Reaction</b>
1.	User enters username and password	System verifies the credentials and the teacher is logged in.
<b>Alternate Scenarios:</b>		
1a: If the entered username or password is incorrect, the system prompts the user to re-enter the credentials		
<b>Post Conditions</b>		



SMS-024: Log in	
Step #	Description
1	The user has successfully logged into the system.
Use Case Cross referenced	None

#### 4.2.25. View Attendance (Student)

SMS-025: View Attendance		
Use case Id:	SMS-025	
Actors:	Student	
Feature:	Attendance Management	
Pre-condition:	The student must be logged in to the system.	
Scenarios		
Step #	Action	Software Reaction
1.	The Student selects a course to view the attendance.	System presents a list of days and the corresponding attendance.
Alternate Scenarios:		
1a: System does not show any courses because the student has not registered a course yet.		
2a: System does not show any attendance because the teacher has not uploaded it yet.		
Post Conditions		
Step #	Description	
	The student can view the attendance for the selected course.	
Use Case Cross referenced	None	

#### 4.2.26. View Registered Courses (Student)

SMS-026: View Registered Courses		
Use case Id:	SMS-026	
Actors:	Student	
Feature:	Course Management	
Pre-condition:	The student must be logged in to the system.	
Scenarios		
Step #	Action	Software Reaction
1.	The student navigates to the "View Registered Courses" section.	System presents a list of courses registered by the student.
Alternate Scenarios:		
1a: System does not present a list of courses as they have not been registered by the student.		
Post Conditions		
Step #	Description	
1	Students can view the courses they have registered.	
Use Case Cross referenced	None	

#### 4.2.27. Add Course (Student)

SMS-027: Add Course		
Use case Id:	SMS-027	
Actors:	Student	
Feature:	Course Management	
Pre-condition:	The student must be logged in to the system.	
Scenarios		
Step #	Action	Software Reaction

SMS-027: Add Course		
1.	The student enters a course to register.	System registers student into the course
<b>Alternate Scenarios:</b>		
<p><b>1a:</b> The course does not exist/invalid course, and a message indicating that will be displayed.</p> <p><b>2a:</b> The student is already registered in the course, and a message indicating that will be displayed.</p>		
<b>Post Conditions</b>		
<b>Step #</b>	<b>Description</b>	
	The student is registered into the course.	
<b>Use Case Cross referenced</b>		None

#### 4.2.28. Drop Course (Student)

SMS-028: Drop Course		
Use case Id:	SMS-028	
Actors:	Student	
Feature:	Course Management	
Pre-condition:	The student must be logged in to the system.	
Scenarios		
Step #	Action	Software Reaction
1.	The student enters a course to drop.	System removes student from the course.
Alternate Scenarios:		
1a: The course does not exist/invalid course, and a message indicating that will be displayed.		
2a:The student is not registered in the course, and a message indicating that will be displayed.		
Post Conditions		

SMS-028: Drop Course	
Step #	Description
	The student is removed from the course.
Use Case Cross referenced	None

#### 4.2.29. Transcript (Student)

SMS-029: Transcript		
Use case Id:	SMS-029	
Actors:	Student	
Feature:	Grade and Transcript	
Pre-condition:	The student must be logged in to the system.	
Scenarios		
Step #	Action	Software Reaction
1.	User navigates to the "Transcript" section.	System retrieves and displays the user's Transcript (course name, marks, GPA, etc.).
Alternate Scenarios:		
1a: The student is yet to be graded, hence the transcript will be empty.		
Post Conditions		
Step #	Description	
1	The student can view the transcript.	
Use Case Cross referenced	None	

#### 4.2.30. Pay Fees (Student)

SMS-030: Pay Fees	
Use case Id:	SMS-030
Actors:	Student

SMS-030: Pay Fees		
Feature:		Fee Slips Generation
Pre-condition:		The student must be logged in to the system.
Scenarios		
Step #	Action	Software Reaction
1.	User enters the course.	System checks if the course exists.
2.	User enters amount to pay	System validates amount and fee is paid
Alternate Scenarios:		
1a: The course does not exist/invalid course, and a message indicating that will be displayed. 2a:The student is not registered in the course, and a message indicating that will be displayed.		
Post Conditions		
Step #	Description	
	The course fee has been paid.	
Use Case Cross referenced		None

#### 4.2.31. Fees Summary (Student)

SMS-031: Fees Summary		
Use case Id:	SMS-031	
Actors:	Student	
Feature:	Fee Slips Generation	
Pre-condition:	The student must be logged in to the system.	
Scenarios		
Step #	Action	Software Reaction
1.	User navigates to the "Fees Summary" section.	System retrieves and displays the user's Fees Summary (course name,total fees, paid fees, etc.).

SMS-031: Fees Summary		
<b>Alternate Scenarios:</b>		
<b>1a:</b> The Student is not registered in any courses.		
<b>Post Conditions</b>		
Step #	Description	
	The student can see the fee summary..	
<b>Use Case Cross referenced</b>		None

#### 4.2.32. Log Out (Student)

SMS-032 : Logout		
Use case Id:	SMS-032	
Actors:	Student	
Feature:	User Authentication and Authorization	
Pre-condition:	The student must be logged into the system	
Scenarios		
Step#	Action	Software Reaction
1.	User selects the “Logout” option.	System logs the user out and redirects them to the login page.
Alternate Scenarios:		
1a: If any operation is in-progress, the system doesn’t logout the teacher until the operation is completed.		
Post Conditions		
Step#	Description	
1	The user is logged out of the system.	
Use Case Cross referenced		None

## 5. Non-functional Requirements

### 5.1. Performance Requirements

- Speed:
  - The system should promptly respond to the user requests.
  - Data should be updated in real-time and without any noticeable details.
  - The processes such as course registration, attendance tracking, grade management etc. should be efficient.
- Precision:
  - The student records, grades, attendance etc. should be accurate.
  - Calculations such as GPA calculation should be precise.
  - Data should be consistent throughout the modules.
- Capacity:
  - The system should be able to accommodate the entire student body and faculty,
- User Authentication:

The learning management system will be used by various stakeholders, i.e. students, teachers, admins, etc. Thus, the system must have an exceptional response time at the time of the authentication. The system should authenticate the user in under 2 seconds. This is called “swift response” and its purpose is to improve the user experience. It also makes the system very efficient to use. The system should allow at least 2000 user authentication at the same time.

### 5.2. Safety Requirements

- Data Loss Prevention:
  - Safety Guards against data loss should be implemented especially in case of system crash or failures.
- Harm Migration:
  - Actions that can harm the system must be prevented.
  - Error handling mechanisms should be implemented.
- Safety Certifications and Compliance:
  - The system should adhere to relevant safety standards and regulations.
  - The system should comply with data protection laws.
- Documentation and training:
  - Making a document helps understand what the management system comprises. Making of the manual guides, brochures and the training workshops helps the user: admin, teacher, student and all the other stakeholders know how to use it and what to do in case an error occurs.

### 5.3. Security Requirements

- User Identity Authentication:
  - It is mandatory for all the users using the SMS to have a valid ID and password.
- Privacy:
  - Users will not be able to access other users data, unless authorized to do so.

### 5.4. User Documentation

- Admins and teachers will be provided with a user manual for the new system.
- Students, on the other hand, will be provided with a how-to tutorial on the website.

## **6. References**

Project Proposal



## **7. Appendices**

N/A