

Student Management System (SMS)

Software Requirements Specification (SRS)

Prepared by: Al-Saeed Wael, Fares Tarek, Seif 3laa Eldin,
Yousif Sherif, Adham Mahmoud, Kirolos Melad, Yousef
Ahmed

Submission Date: 30/11/2024

Table of Contents

- 1. Introduction**
 - 1.1 Purpose**
 - 1.2 Scope**
 - 1.3 References**
 - 1.4 Overview**

- 2. Overall Description**
 - 2.1 Product Perspective**
 - 2.2 Product Features**
 - 2.3 User Classes and Characteristics**
- 3. Use Case Diagram and Descriptions**
- 4. External Interface Requirements**
 - 4.1 User Interfaces**
 - 4.2 Hardware Interfaces**
 - 4.3 Software Interfaces**
- 5. Dataflow Diagrams**

1.0 Introduction

1.1 Purpose

The purpose of the Student Management System (SMS) software is to streamline student enrollment. It is intended for educational institutions to improve operational efficiency and provide a centralized platform for academic management.

1.2 Scope

The Student Management System (SMS) is a python-based platform that facilitates student's teacher management, course registration. The system is designed to be accessible on multiple platforms, including desktops, tablets, and smartphones. It will serve students, teachers, and administrative staff.

1.3 References

- [1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications."
- [2] Standardized SRS Templates - Coursera
- [3] Example SRS for Cafeteria System.

1.4 Overview

The system is designed to manage student records, facilitate course registration, and provide tools for effective communication among stakeholders. It also highlights system constraints, external interfaces, and assumptions.

2.0 Overall Description

2.1 Product Perspective

The online e-learning platform is a standalone system designed to provide accessible, user-friendly, and interactive educational content to learners.

The e-learning platform will feature a range of functionalities aimed at addressing the needs of multiple stakeholders:

1. **For Learners:**
 - View course data and assigned teacher.
2. **For Teachers:**
 - Real-time analytics to track learner data and info.
3. **For Administrators:**
 - Role-based access control for managing users and permissions.
 - Detailed reporting and performance monitoring of courses and users.
 - Tracking the performance of both learners and Teachers.
 - Adding/Removing specific students, teachers and courses.

This e-learning platform will compete with existing solutions such as Ebn El Haytham , FLMS and more. Obviously its not a new idea but there is a gap in these softwares we are going to fill with our additional features

2.2 Product Features

Course Enrollment

Students can search for available courses, enroll in classes.

Student Access:

Students can view assigned courses and assigned teachers.

Teacher Access:

Teachers can View assigned courses and students.

Admin Access:

Full control over user accounts, including creating, updating, and deleting Teachers / Students / Courses.

Manage courses, class schedules, and teacher assignments.

2.3 User Classes and Characteristics

1. Students:

- Primary users who access course materials and View Teacher's Name assigned to the course.

2. Teachers:

- Manage courses and view Students.

3. Administrators:

- Manage user roles, Add/Delete Teachers, Students, Courses.

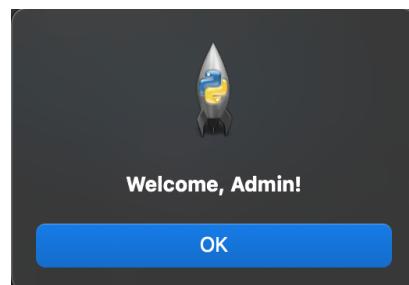
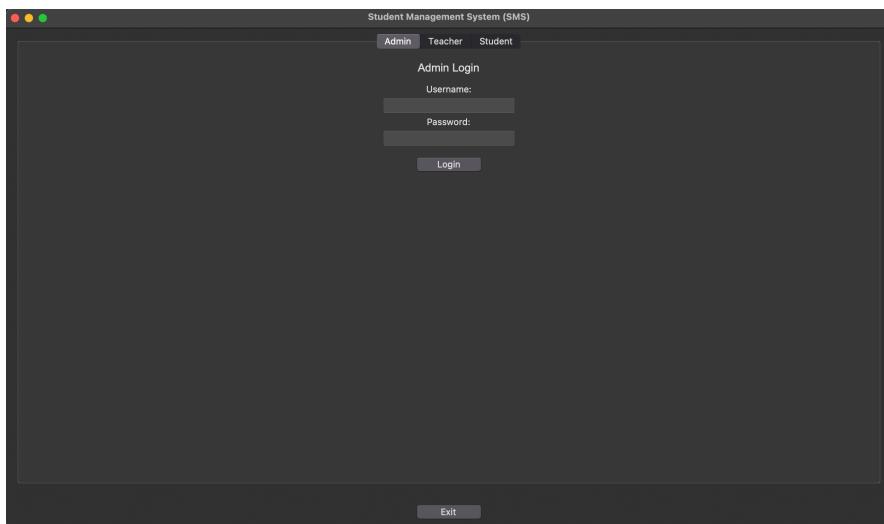
4. Courses:

- Students and Teachers assigned.

4.0 External Interface Requirements.

4.1 User Interface

1. **Admin Panel:** Manage Teachers (Add>Show/Remove), Courses (Add>Show/Remove), and Students (Add>Show/Remove).



2. **Teacher Panel:** View Selected course's assigned students.

The screenshots illustrate the Teacher Panel interface and its interaction with the Courses list.

Screenshot 1: Teacher Panel Login

This screenshot shows the "Teacher Panel" login screen. It features a "Username:" field (highlighted in blue), a "Password:" field, and a "Login" button. The "Teacher" tab is selected in the top navigation bar. The title bar reads "Student Management System (SMS)".

Screenshot 2: Courses List with Alert

This screenshot shows the "Courses" list under "Teacher Actions". A single row is visible with Course ID 1 and Course Name Math 1. A modal dialog box is overlaid on the screen, containing a yellow warning icon with an exclamation mark and the text "Please select a course first!". An "OK" button is at the bottom right of the dialog.

Screenshot 3: Courses List with Selection

This screenshot shows the same "Courses" list, but the row for Course ID 1 has a blue selection bar underneath it, indicating it is currently selected.

| Students in Math 1 | |
|--------------------|--------------|
| Course: Math 1 | |
| Student ID | Student Name |
| 1 | Ahmed |
| 2 | Alaa |
| 3 | Nader |

[Close](#)

4. Student Panel: View course details, assigned teachers.

Student Management System (SMS)

Admin Teacher **Student**

Student Login

Username:

Password:

[Login](#)

[Exit](#)

Student Dashboard

Student: Ahmed (ID: 1)

| Course ID | Course Name | Teacher Name |
|-----------|----------------------|------------------|
| 1 | Math 1 | Dr Ahmed Atef |
| 2 | Software Engineering | Dr Ahmed Samir |
| 4 | Computer Vision | Dr Marwa Alsayed |
| 5 | Neural Network | Dr Sahar Ahmed |

[Logout](#)

Design Approach

1. **Consistent Theme:** Each panel uses a uniform design layout for easy navigation.
2. **List Views:** Displays student records and courses.
3. **Modular Navigation:** Back-to-Dashboard buttons simplify movement across sections.
4. **Error Messages:** Pop-ups for invalid input, authentication errors, or empty fields.

4.2 Hardware Interfaces

Admin Panel

| | |
|-------------------------------|---|
| Keyboard | For entering login credentials, teacher/student/course information, and performing CRUD operations. |
| Mouse/Touchpad | For navigating the GUI, selecting options, and performing actions like adding or deleting entries. |
| Monitor/Display Screen | To display the admin dashboard, teacher/student lists, and system notifications. |
| CPU/GPU | Required for running the Python-based GUI and handling database operations efficiently. |
| Hard Drive/SSD | To store the system and Python dependencies locally for quick access. |
| Processor | Dual-core 2.0 GHz or higher. |
| RAM | At least 4 GB for a smooth operation. |
| Storage | Minimum 2 GB free space for Python installation and dependencies. |
| Display Resolution | 1280x720 or higher for optimal GUI experience. |

Teacher Panel

| | |
|-------------------------------|---|
| Keyboard | For entering login credentials, teacher/student/course information, and performing CRUD operations. |
| Mouse/Touchpad | For navigating the GUI, selecting options, and performing actions like adding or deleting entries. |
| Monitor/Display Screen | To display the admin dashboard, teacher/student lists, and system notifications. |
| CPU/GPU | Required for running the Python-based GUI and handling database operations efficiently. |
| Hard Drive/SSD | To store the system and Python dependencies locally for quick access. |
| Processor | Dual-core 2.0 GHz or higher. |
| RAM | At least 4 GB for a smooth operation. |
| Storage | Minimum 2 GB free space for Python installation and dependencies. |
| Display Resolution | 1280x720 or higher for optimal GUI experience. |

Student Panel

| | |
|-------------------------------|---|
| Keyboard | For entering login credentials, teacher/student/course information, and performing CRUD operations. |
| Mouse/Touchpad | For navigating the GUI, selecting options, and performing actions like adding or deleting entries. |
| Monitor/Display Screen | To display the admin dashboard, teacher/student lists, and system notifications. |
| CPU/GPU | Required for running the Python-based GUI and handling database operations efficiently. |
| Hard Drive/SSD | To store the system and Python dependencies locally for quick access. |
| Processor | Dual-core 2.0 GHz or higher. |
| RAM | At least 4 GB for a smooth operation. |
| Storage | Minimum 2 GB free space for Python installation and dependencies. |
| Display Resolution | 1280x720 or higher for optimal GUI experience. |

Hardware Interface Workflow

1. Admin Login:

- Keyboard input → GUI displayed on monitor → User navigates options with mouse.

2. Teacher Assignments:

- Keyboard input for grades/attendance → CPU processes and updates data → Display results on the screen.

3. Student Access:

- Mouse navigation for viewing details → CPU fetches and displays relevant data.

4.3 Software Interfaces

The software interface is designed to be intuitive and user-friendly, featuring the following sections:

1. **Dashboard:** Displays key statistics and shortcuts to primary functions.
2. **Students Module:** Allows the addition, deletion, and modification of student records.
3. **Courses Module:** Manage course information and their details.
4. **Grades Module:** Record and update student grades for each course.

Technologies Used

The Student Management System is built using the following technologies:

- **Database:** MySQL or SQLite for managing data.

| User Table | | | Course Table | | |
|-------------|------------|-----------|----------------|----------------------|------------|
| id | username | password | id | name | teacher_id |
| 1 | admin | admin | 1 | Math 1 | 1 → |
| | | | 2 | Software Engineering | 2 → |
| | | | 3 | Machine Learning | 3 → |
| | | | 4 | Computer Vision | 4 → |
| | | | 5 | Neural Network | 5 → |
| | | | | | |
| | | | | | |
| | | | | | |
| Grade Table | | | Sequence Table | | |
| id | student_id | course_id | grade | name | seq |
| 1 | 1 → | 1 → | NULL | admins | 16 |
| 2 | 2 → | 1 → | NULL | teachers | 5 |
| 3 | 3 → | 1 → | NULL | students | 13 |
| 4 | 1 → | 2 → | NULL | courses | 5 |
| 5 | 2 → | 2 → | NULL | grades | 33 |
| 6 | 4 → | 2 → | NULL | | |
| 7 | 12 → | 2 → | NULL | | |
| 8 | 8 → | 2 → | NULL | | |
| 9 | 3 → | 3 → | NULL | | |
| 10 | 5 → | 3 → | NULL | | |
| 11 | 6 → | 3 → | NULL | | |
| 12 | 7 → | 3 → | NULL | | |
| 13 | 8 → | 3 → | NULL | | |
| 14 | 9 → | 3 → | NULL | | |
| 15 | 10 → | 3 → | NULL | | |
| 16 | 11 → | 3 → | NULL | | |

- **Frontend:** To be implemented using python.

- **Backend:** Python for handling business logic and database interaction.

Use Cases

1. Adding a New Student:

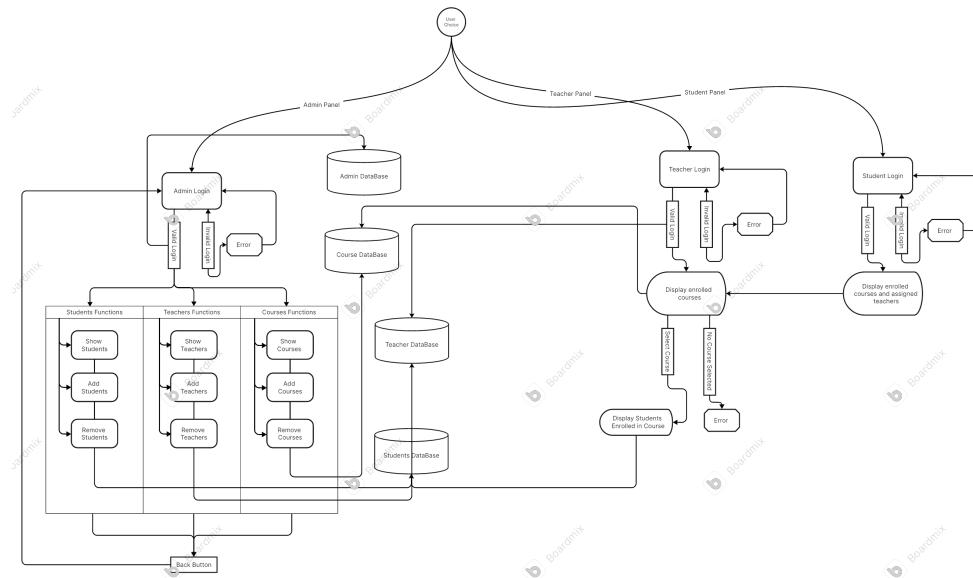
- The admin enters the student's details, and the system saves them in the database.

2. Enrolling a Student in a Course:

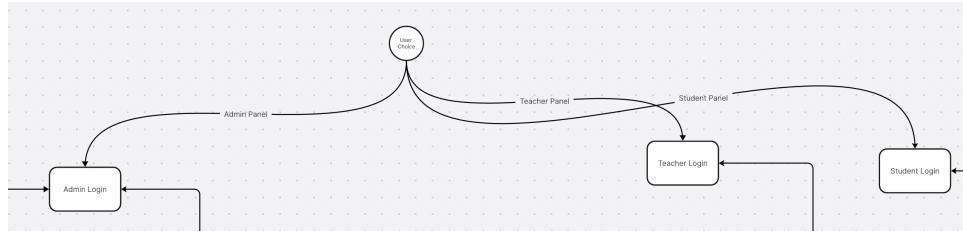
- The admin selects a student and assigns them to a course. The enrollment details are recorded.

5.0 Dataflow Diagrams

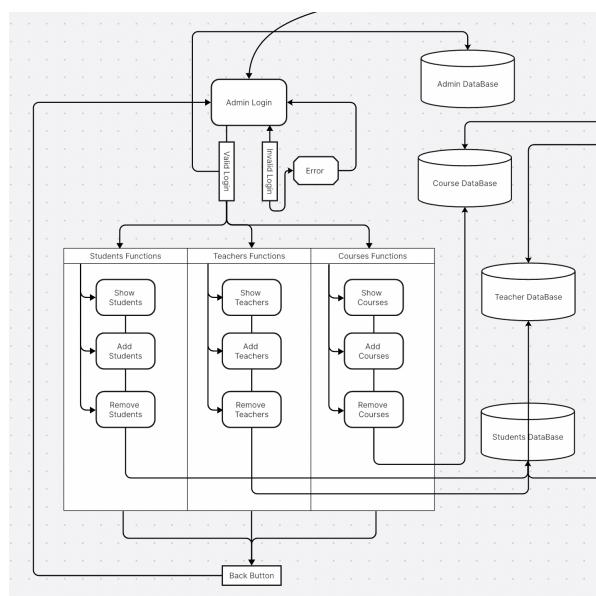
General Data flowchart:



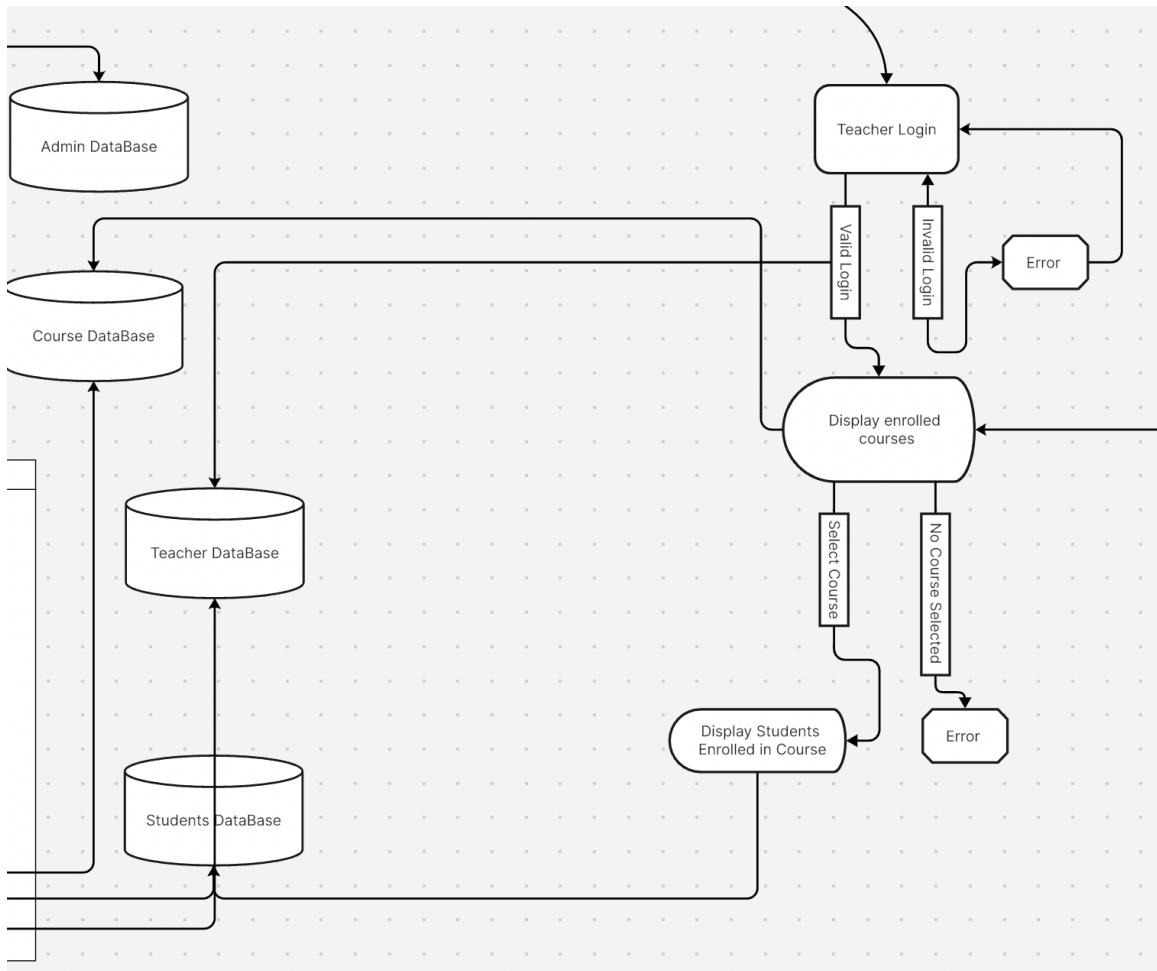
User Flowchart:



Admin Flowchart:



Teacher Flowchart:



Student Flowchart:

