

Assignment No:	(2) Group Programming Project.
Assignment Title:	Student Attendance System in Python.
Assignment Weighting (%):	30%
Number of students:	A maximum of four (3) and a minimum of two (2).
Assignment written report:	80% - This group assignment, designed for up to 3 students, will assess your understanding and practical application of Python programming in creating a pilot Student attendance System (SaS). The students will create a report to document their system and its design. This assignment promotes teamwork, software design, and the utilisation of key Python concepts.
Assignment presentation:	20% - A group project of up to fifteen (15) slides and ten (10) minutes for the presentation. PowerPoint (ppt & pptx) format only.

Table of Contents

Pilot Student Attendance Project Objective	4
Objectives	4
Assignment Details	4
Task 1	4
System Design and Planning (15%).....	4
Task 2	5
Core Functionality (25%).....	5
Task 3	5
User Interface (UI) Development (15%)	5
Task 4	5
Security and Privacy (10%).....	5
Task 5	6
Documentation and Testing (20%)	6
Deliverables	6
Assessment Criteria.....	6
Additional Notes.....	6
Project Success	6
Task 6	7
In-class Presentation (20%)	7
Introduction	7
Appendices	9
1. Suggested Recommended Libraries	9
2. Sample Data Schema and Data.....	9
Students	9
Courses	9
Classes.....	9
Attendance	10
Relationships	10
3. Sample Databased Relationship for the Student attendance System (SaS)10	
4. Sample flowchart for The Student attendance System (SaS)	11

5. Work Breakdown Structure (WBS).....	11
Sample Student Attendance System Project	12
Level 1 (Sample of Levels).....	12
Level 2 (Sample tasks).....	12
Level 3 (Example for Screen Development).....	12

Pilot Student Attendance Project Objective

The pilot project aims to develop a basic (pilot) Student Attendance System using Python language to develop the project. This system will serve as a prototype to track the attendance of students in a classroom setting. While this project's scope is limited, it will provide a solid foundation for future enhancements and more complex attendance management systems.

This group assignment, designed for up to 3 students, will assess your understanding and practical application of Python programming in creating a pilot. It promotes teamwork, software design, and the utilisation of key Python concepts.

Objectives

- Collaboratively develop a functional and user-friendly Student attendance System (SaS).
- Implement Python programming skills to create an interactive application.
- Explore data handling and user interface development with Python libraries.

Assignment Details

This project involves building a software application to manage student attendance in a classroom setting. Throughout development, ensure the system adheres to data security principles and student privacy regulations.

Task 1

System Design and Planning (15%)

Conceptual Design

- a. Define the system's functionalities, including student registration, attendance recording, and data reporting.
- b. Consider different methods for recording attendance (e.g., manual input or scanning using QR codes).
- c. Design the user interface layout for both instructor and student access (separate interfaces can be considered).

Data Model

- a. Plan the data structure for storing student information, attendance records, and (optionally) course details.
- b. Consider using Python libraries for data manipulation (e.g., pandas).

Task 2

Core Functionality (25%)

- a. Develop a mechanism for student registration, capturing essential information like name and ID.
- b. Implement a system for instructors to record attendance. This could involve:
 - Manually marking attendance from a student list.
 - Implementing a QR code scanning feature (requires additional research and libraries).
- c. Create a method to store and retrieve attendance data securely. Depending on the complexity, consider using a file-based system (e.g., CSV, JSON) or a database (e.g., SQLite, MySQL, etc.).

Task 3

User Interface (UI) Development (15%)

- a. Design a user-friendly interface for instructors. Include functionalities to:
 - View registered students.
 - Record attendance for a specific class session.
 - Generate reports on student attendance (e.g., Students in each class, percentage attendance per student, compare attendance per class, etc.).
- b. Develop a basic interface for students (optional, depending on the chosen attendance recording method). This might involve:
 - Students logging in with their ID to mark their attendance (QR code scanning).
- c. Python libraries like `tkinter` can be used for a simple graphical user interface (GUI) or consider command-line interface (CLI) libraries for text-based interaction.

Task 4

Security and Privacy (10%)

- a. Implement measures to protect student data privacy. This might include:
 - Password protection for access to the system.
 - Secure storage of attendance records.
- b. Restriction on data access based on user roles (instructor vs. student).

Task 5

Documentation and Testing (20%)

- a. Create a comprehensive README.md file that details:
 - Installation and usage instructions for the application.
 - Explanation of functionalities.
 - Data structure and file formats used.
- b. Develop unit tests for core functionalities of the system. This ensures the reliability of your code.
- c. Ensure the code adheres to PEP 8 style guidelines for readability and maintainability.

Deliverables

- a. Completed Python application with all implemented features.
- b. Comprehensive documentation (README.md)
- c. A suite of unit tests for the application.
- d. A project log documenting development progress, team member contributions, and development decisions.

Assessment Criteria

- a. Functionality and user-friendliness of the attendance system.
- b. Code quality, organisation, and adherence to best practices.
- c. Implementation of security and privacy measures.
- d. Comprehensiveness of documentation and testing strategies.
- e. Evidence of effective team collaboration and project management is reflected in the project log.

Additional Notes

- a. This assignment provides a framework, and specific functionalities can be tailored based on your team's capabilities and chosen libraries.
- b. Explore additional features like exporting attendance data to different formats, exporting reports for specific date ranges, or integrating with existing class management systems (optional).
- c. Remember to comment your code for better understanding and maintainability.

Project Success

Success in this project will demonstrate your ability to:

- a. Collaborate effectively in a team environment.
- b. Apply Python programming concepts for data handling and user interaction.
- c. Design and develop a functional and user-friendly application.

Task 6

In-class Presentation (20%)

Introduction

Clearly state the presentation's purpose by introducing The Student attendance System (SaS) developed by your team. The presentation should begin by introducing the development team and briefly outlining the purpose of the presentation.

Provide a high-level overview of The Student Attendance System, emphasising its importance in streamlining attendance tracking and improving efficiency within educational institutions.

Briefly outline the key components and benefits of the system.

The Problem Statement

Explain the approach taken to identify the issue related to the project and explain the challenges schools or institutions face regarding student attendance tracking.

Discuss the negative consequences of poor attendance, such as academic performance, student engagement, and school funding, to highlight the impact.

Create slides to explain the following:

Introduce your project solution, "The Student attendance System (SaS)"

1. System overview by providing a high-level explanation of the system's architecture and functionality.
 - 1.1. Explain the key features by discussing the system's core features.
 - 1.2. Real-time tracking by monitoring attendance by updating student attendance records, which are updated in real-time.
 - 1.3. Notifications generated by reports or screens (dashboards) will alert administrative staff regarding student absences or poor timekeeping.
 - 1.4. Analytics on stored attendance data used to monitor can also be used to analyse and identify trends and patterns in attendance.
 - 1.5. How can the system be integrated with existing school management software?
 - 1.6. Explain the system's benefits and advantages, including improved attendance rates, enhanced communication, and data-driven decision-making.
2. Demonstration or Case Study
 - 2.1. Presentation using Visual aids like screenshots, diagrams, or a live demonstration to illustrate the system's features.
3. Conclusion
 - 3.1. Recap key points by summarising The Student attendance System's main features and benefits.
 - 3.2. Encourage the audience to consider the benefits of adopting or implementing the system in their schools or institutions.

4. Additional Tips

- 4.1. Use clear and concise language.
- 4.2. Practice your presentation beforehand.
- 4.3. Engage with the audience by asking questions or encouraging discussion.
- 4.4. Be prepared to answer questions about the system.

By following this outline, you can effectively communicate the value and potential impact of your team's Student Attendance System.

Appendices

1. Suggested Recommended Libraries

The following is a sample of selective libraries that can be used to help you develop your project.

- MySQL Connector/Python:** This is used to interact with MySQL databases.
- SQLAlchemy:** A high-performance ORM that abstracts database interactions.
- Tkinter:** A standard Python GUI toolkit.
- PyQt5:** A popular cross-platform GUI toolkit.
- Kivy:** A multi-touch GUI toolkit for mobile and desktop applications.
- Flask or Django:** This is used to build web-based interfaces if needed.

Combining these libraries and following the provided design can create a robust and feature-rich student attendance system.

2. Sample Data Schema and Data

The following tables are samples for the developers to understand and have a base for developing their assignment. You will need to include additional tables and columns for the delivery of your project.

Students

<i>student_id</i>	<i>student_name</i>	<i>student_email</i>	<i>student_password</i>
1	John Doe	j.doe@myed.edu.com	password123
2	Jane Smith	j.smith@myed.edu.com	password456
3	Michael Johnson	m.johnson@myed.edu.com	password789

Courses

<i>course_id</i>	<i>course_name</i>	<i>course_code</i>
1	Introduction to Programming	COMP101
2	Data Structures and Algorithms	COMP202
3	Database Systems	COMP303

Classes

<i>class_id</i>	<i>course_id</i>	<i>class_date</i>	<i>class_time</i>
1	1	2024-09-06	10:00:00
2	1	2024-09-13	10:00:00
3	2	2024-09-20	13:00:00

Attendance

<i>attendance_id</i>	<i>student_id</i>	<i>class_id</i>	<i>attendance_status</i>	<i>excuse_reason</i>
1	1	1	present	
2	2	1	absent	
3	3	1	present	
4	1	2	present	
5	2	2	absent	Illness
6	3	2	present	

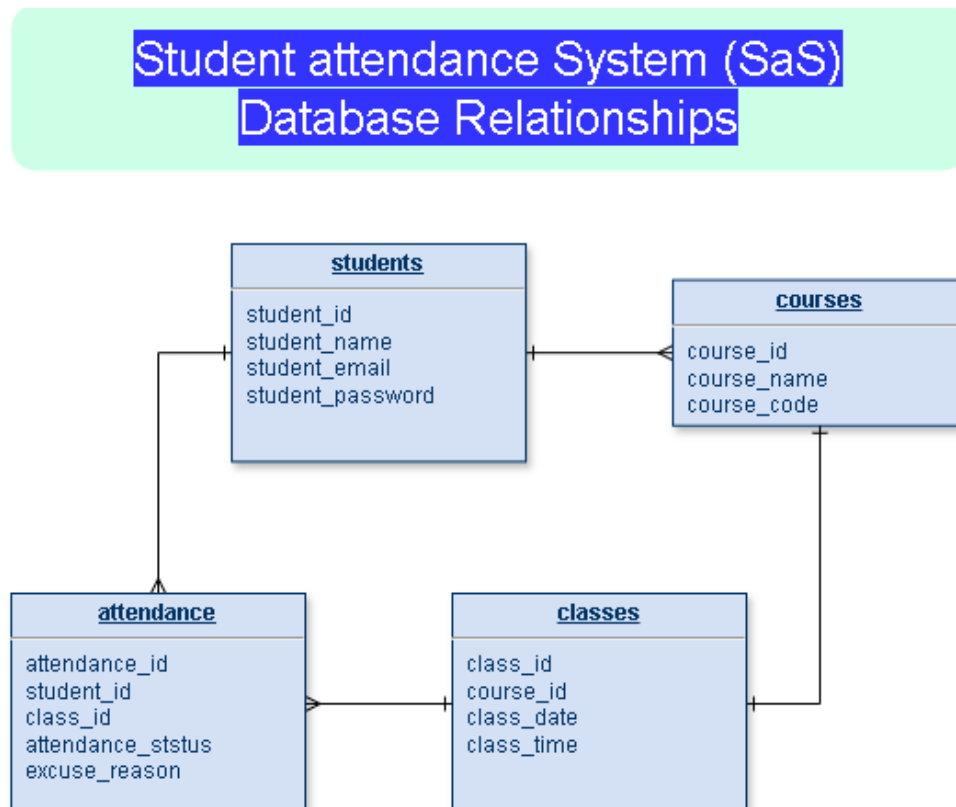
Relationships

I have included a sample of relationships. There will be more, which will become apparent as you investigate and research your project.

- A student can attend multiple classes.
- A class can have multiple students attending.
- A course can have multiple classes.

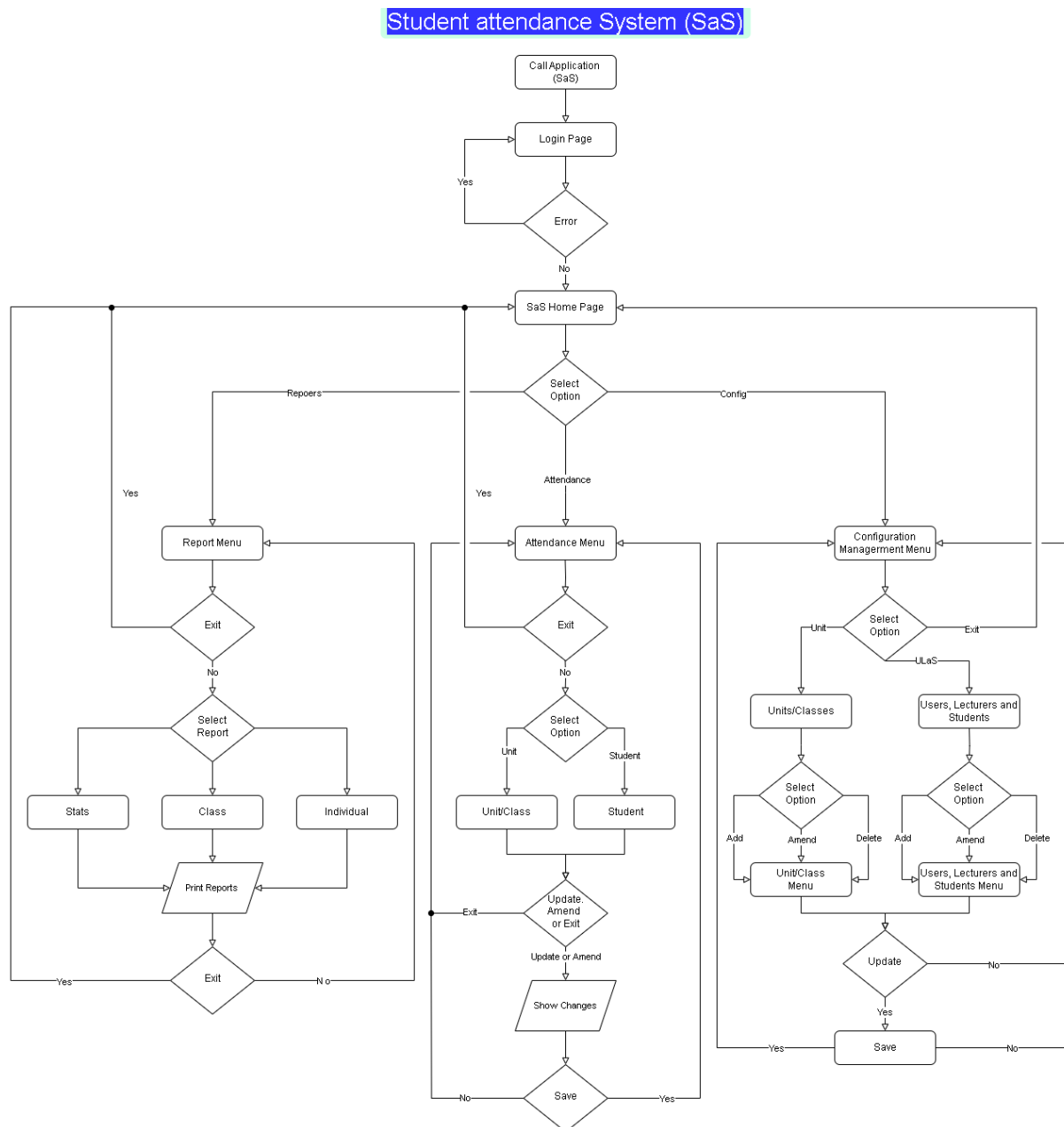
3. Sample Databased Relationship for the Student attendance System (SaS)

The following diagram is a sample design for the Database relationships.



4. Sample flowchart for The Student attendance System (SaS)

The following diagram is a sample design for the logic and processes needed for this type of project.



5. Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) outlines the hierarchical breakdown of tasks required to complete the Student Attendance System project. It divides the project into manageable phases, including project initiation, system analysis and design, screen development, database development, attendance tracking and reporting, and

testing and deployment. Each phase is further divided into more detailed tasks, ensuring a clear and organised approach to project execution. This WBS is a valuable tool for project planning, resource allocation, and tracking progress throughout development. Please use the sample WBS to create your own WBS.

Sample Student Attendance System Project

Level 1 (Sample of Levels)

1. Project Initiation.
2. System Analysis and Design.
3. Screen Design and Development.
4. Database Development.
5. Attendance Tracking and Reporting.
6. Testing and Deployment.

Level 2 (Sample tasks)

- 1.1. Project Planning.
- 1.2. Resource Allocation.
- 1.3. Risk Assessment.
- 2.1. Requirements Gathering.
- 2.2. System Architecture.
- 2.3. Database Design.
- 3.1. User Interface Design.
- 3.2. Screen Development.
- 3.3. Data Validation.
- 4.1. Database Creation.
- 4.2. Data Access Layer Development.
- 4.3. Data Migration (if applicable).
- 5.1. Attendance Tracking Implementation.
- 5.2. Report Generation.
- 5.3. Data Visualisation.
- 6.1. Unit Testing.
- 6.2. Integration Testing.
- 6.3. System Testing.
- 6.4. User Acceptance Testing.
- 6.5. Deployment Planning.
- 6.6. Deployment Execution.

Level 3 (Example for Screen Development)

- 3.2.1. Login Screen.
- 3.2.2. Student List Screen.
- 3.2.3. Attendance Entry Screen
- 3.2.4. Report Viewing Screen