ATTENDANCE MANAGEMENT SYSTEM

A report submitted for the course named Project - I (CS3201)

Submitted By

T. SAMEER REDDY
SEMESTER - V
220101009

Supervised By

DR.KISHORJIT NONGMEIKAPAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SENAPATI, MANIPUR

OCTOBER, 2024

Declaration

In this submission, I have expressed my idea in my own words, and I have adequately cited and referenced any ideas or words that were taken from another source. I also declare that I adhere to all principles of academic honesty and integrity and that I have not misrepresented or falsified any ideas, data, facts, or sources in this submission. If any violation of the above is made, I understand that the institute may take disciplinary action. Such a violation may also engender disciplinary action from the sources which were not properly cited or permission not taken when needed.

T. Sameer Reddy
220101009

DATE:

Department of Computer Science Engineering
Indian Institute of Information Technology Senapati, Manipur

Dr.Kishorjit Nongmeikapam
Assistant Professor

Email: kishorjit@iiitmanipur.ac.in
Contact No: +91 8974008610

To Whom It May Concern

This is to certify that the project report entitled **"ATTENDANCE MANAGEMENT SYSTEM",** submitted to the department of Computer Science and Engineering, Indian Institute of Information Technology Senapati, Manipur in partial fullfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering is record bonafide work carried out by **T. Sameer Reddy** bearing roll number ROLLNO.220101009

Signature of Supervisor

(Dr.Kishorjit Nongmeikapam)

Signature of the Examiner 1 .....

Signature of the Examiner 2 .....

Signature of the Examiner 3 .....

Signature of the Examiner 4 .....

Abstract

In the present learning scenario, proper and reliable attendance tracking is of utmost importance for colleges to maintain a right record of their participation in academic activities. The traditional method is time-consuming, prone to mistakes, and inefficient, especially in the cases of large numbers of students. Hence, this system proposal proposes attendance management using file-structure-based databases on C programming language. The system will auto-mechanize processes associated with taking and managing attendance while providing a dynamic, secure and friendly-use interface for faculty members.

This primary aim of this project is simplification and making the process of marking attendance records for students, so a comprehensive software solution is available that can easily record electronic attendance records. The system has provided easy creation, updating, deletion, and reading of records as it does not compromise the safety of data through authentication and authorization mechanisms. The faculties can easily log in into the system with their credentials and manage the attendance data correctly. Each session of attendance is stored as a permanent file, and the date and time is set as the filename so that records are tracked and retrieved precisely at whatever time they may be needed.

Acknowledgement

I would like to express my sincere gratitude to several individuals for supporting me throughout my Project. First, I wish to express my sincere gratitude to my supervisor, *Dr.Kishorjit Nongmeikapam*, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my project and writing of this thesis. His immense knowledge, profound experience and professional expertise has enabled me to complete this project successfully. Without his support and guidance, this project would not have been possible. I could not have imagined having a better supervisor in my study. I would like to express our deepest gratitude to our guide, for his valuable guidance, consistent encouragement, personal caring, timely help and providing us with an excellent atmosphere for doing research.

T. Sameer Reddy

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This project implements the use of an attendance management system using C programming. Accordingly, the system is developed to follow how the students show attendance in classes efficiently and easily in the academic setting.

1.1 Outline

The key features of the attendance management system are as follows:

- **User Authentication:** The system deploys a robust user authentication mechanism so that only authorized faculty members can access and modify their respective details related to attendance. Each faculty member's log-ins are done through unique IDs and passwords, which are stored and verified securely against a database. Thus, unauthorized access to the system is prevented while keeping student attendance information confidential.
- **Record Creation:** It is quite easy for the faculty to create a new attendance record for their subjects. It accepts inputting the date, time, and subject code of the class and then marking students as present or absent, but the system will automatically generate an identifier for each record which can easily be tracked for retrieval.
- **Search:** It had the flexibility of a search function for faculty members that could easily find any attendance record based on dates, times, subject codes, or student names. Hence, faculty will be able to analyze fairly the attendance behavior and identify who might be at risk.

- **Deletion:** Although the system allows deletion of attendance records, it informs the users that this is an irreversible activity to avoid any data loss unintentionally. Before letting a record be deleted, the system questions the faculty member with a yes-no question so that she or he is very clear about the after-effect of that activity.

- **File Handling:** The system uses an advanced file handling mechanism to store and retrieve data persistently based on attendance. All the records of attendance are saved in a text file with a structured name consisting of date and time as well as subject code. Such organizational record finding and retrieval are simplified. It also provides error-handling mechanisms that avoid corruption of data and ensure data integrity to the information stored.

1.2 Problem Statement

Traditional ways of attendance taking and record-keeping involve cumbersome and time-consuming roll calls or hand registers that are prone to errors and inefficient in large classes or institutions. It may result in inaccuracies, loss of data, and a burden of administrative tasks, as well as difficulty in understanding attendance patterns.

Roll calls are disruptive to the classroom setting since faculty members will spend precious classroom time calling out student names and recording responses on a roll, for example. This can also mean that the instructional time is shortened and also creates an environment that is less intense. With paper registers, one finds difficult to organize, keep, and retrieve them - especially in a class with various sections or even up to long durations. In most instances, manual records are prone to errors by either miscounting or even recording incorrect information, hence giving inaccurate attendance data.

Moreover, traditional attendance analysis and report generation can be quite a drag in itself, as it takes away so much time and manual effort. The faculty members have to themselves go through the attendance records, check their students' attendance rates, and figure out who are at risk. It is particularly difficult to do so in a large class or when multiple sections have to be considered. Moreover, this level of detail and insight cannot be achieved through manual analysis, which is very important for student support and intervention.

The system in this project simplifies the process of tracking attendance, increases the accuracy of attendance management, offers better access to data, and allows for analysis. The system erases all of the labor-intensive steps involved in traditional approaches, significantly reducing opportunities for errors and conserving precious time for faculty; therefore, it improves the effectiveness and efficiency of attendance management processes.

1.3 Motivation

This attendance management system was developed just with the thought of bettering my programming skills in the real application of the subject. It made me achieve a strong notion of several concepts implemented in the language that I am using and their real applications as well. This includes:

- **File handling:** I became an expert in creating, reading, writing, and modifying files in C, an important tool for persistent data storage. I learned how to construct file names and organize attendance records so that they could be retrieved easily.
- **User Authentication:** I have incorporated basic security controls against unauthorized access to the system. It only allows authorized persons to create, modify, and view attendance records since professors will have to log on with unique IDs and passwords.
- **Design for command-line interface:** I designed an interactive command-line interface that doesn't require so much programming knowledge to work. The instructions and prompts I designed for the users were clear and thus easy to understand for users with little or no programming experience.
- **Data management:** I identified strategies on how to organize and manage attendance data in the system. The system involved appropriate data structures and file formats that were used for information storage, all in order to provide efficiency and integrity in the data. And also, I ensured proper implementation of error handling mechanisms to prevent loss of data or uncontrolled system crashes.

Working on this project has been pretty instrumental in toughening up my programming skills as well as gaining insight into the practical application of C programming in the real world. More than anything, I better appreciated file handling, user authentication, and command-line interface designs as well as data management. With these aptitudes, I can further explore and develop more software solutions to specific problems.

Moreover, I came to know that user experience is of critical importance while developing the software. My command-line interface is very intuitive and user-friendly so that even faculty members who are not well versed in programming may use it without much hassle. This has been a huge lesson learned-the designing of software with regard to end users since it improves the usability of the system while making it readily adoptable.

1.4 Purpose

The following are the key objectives of this attendance management system:

- **Accuracy would increase:** The possibility of errors associated with manual recording of attendance will decrease. No human error like miscount or incorrect recording will be possible through the automated systems. It will help in increasing overall accuracy and integrity of the attendance records.
- **Saves time:** Automated systems streamline the tracking of attendance so that the faculty members can spend more hours teaching and other priorities. Traditional methods of recording roll calls or keeping paper registers may look cumbersome and become time-consuming, especially with big classes. Automated systems make it easy to record and manage data about student presence quickly and efficiently.
- **Provide Easy Access to Records**: The system should allow easy and convenient searching and retrieval of attendance records so that the faculty may retrieve data as and when needed. The system would take attendance records in a digital form that can easily be accessed by searching for a particular record based on date, time, and many other parameters. This thereby eliminates the overhead of manual searching through paper records, thus saving time and effort.
- **Improved Security:** Attendance information should be secured through user authentication, which will grant only the authorized faculty members access and modify the attendance data. The system ensures that unauthorized faculty cannot get into the system since the faculty members will log in using unique IDs and passwords; therefore, attendance information for students about their presence or absence will remain private and not to be leaked.
- **Generate Reports:** Future Goal - Not available with this version, but this is a future goal: the ability to generate detailed attendance reports. This might include the following: pattern information, students who could be considered "at-risk," and other types of information which might advance the interested parties' decision-making process.

Chapter 2

Literature Survey

It will source existing research and current practices regarding attendance management systems. This section outlines the project context to explore the approaches, technologies, and challenges in the subject.

2.1 Traditional Attendance Management

2.1.1 Manual Roll Call

Traditional roll call methods, in which a faculty rooster calls out student names and records responses, are usually cumbersome, time-consuming, and error-prone. It distracts from the learning environment, wastes instructional time, and produces less focused practice time. Pen-and-paper roll calls can be a pain when dealing with large classes because of the intensity of the time required to call out and record attendance of all students.

2.1.2 Paper Registers

Paper registers are more controlled; however, they are still plagued by a few drawbacks. Because they are manually stored in space, this will cause them problems in institutions strapped for space. Moreover, being confined to paper may mean that data is wrongly counted or inadvisedly scribbled down. Inaccuracies in the data of attendance and complexities in the analysis of patterns are likely to be problematic issues concerning attendance.

2.1.3 Challenge in processing and reporting attendance data:

Processing the data using traditional approaches can be quite tedious and error-prone. Professors would have to manually read attendance sheets, compute their attendance rates, and put together a list of the students "at risk." This can be quite time-consuming, especially in large classes or when teaching several sections. Additionally, the type of information gathered may not

be as detailed or actionable as they would be with the new approaches and would consequently not have as much depth in terms of offering interventions and support to those students that end up being at risk. For example, one would find it challenging to identify trends, patterns, or relationships in attendance data by just using a manual analysis.

2.2 Automated Attendance Management Systems

2.2.1 Overview of Existing Systems

Quite a number of automated attendance management systems have appeared in the recent past, each with its distinguishing features and capabilities. These systems can be broadly classified into four types that are biometric, RFID-based, smart card-based, and pure software-based.

- **Systems:** This technology uses physiological or behavioral characteristic to recognize a person and then track attendance. Biometric technologies comprise fingerprint scanning, facial recognition, and iris scanning, and most biometric technologies utilized for attendance management are in these forms. These systems have higher accuracy and security because it is not very easy for someone to forge or assume biometric traits. Still, these biometric systems can be very costly and sometimes require specialized hardware or software.

- **RFID-based Systems:** This method makes use of small-sized electronic tags affixed to ID cards or such objects as the student's ID card. The deployment of RFID readers at entry points or within classrooms facilitates automatic tracking of attendance, with a student passing through. RFID systems are relatively inexpensive, easy to deploy, and they can give a real-time view of the attendance of students. However, they could be prone to interference from other RFID signals, and there is a danger of tag tampering or loss.

- **Smart Card Systems:** These are plastic cards with a microchip that contains student information and attendance data. Attendance can be recorded with smart card systems or RFID readers together with access control devices. A smart card system is a safe and flexible system for attendance management because the card can have more information on it- profiles of students, enrollments for courses, payment, among others. The needs for special hardware and software as well as the risk of losing or damaging the card are some of the other issues.

- **Software-based Solutions:** The most general form of an attendance management system is web-based or mobile applications wherein users may enter and record and track attendance data. Such a system can be customized for an institution's peculiar needs and can be integrated with various educational software systems. Flexibility, scalability, and ease of use characterize software-based solutions, but may sometimes demand Internet connectivity and are susceptible to cyberattacks.

It would be compared and contrasted on the aspects of cost and usability in terms of accuracy, security, scalability, and compatibility. Different kinds of attendance management systems will have pros and cons of their own and would suit best according to the specific needs and resource availability of the institution.

2.3 Benefits and Challenges of Automation

2.3.1 Increased Accuracy and Productivity

Automated attendance management means that accuracy and productivity in tracking attendance will undoubtedly be much higher compared to other traditional methods. The chances of human errors arising from manual recording are totally ruled out and the information collected can be relied on for reliability and consistency. In turn, this means accurate analysis over attendance patterns, identification of students at risk, and better decisions regarding engagement and academic performance.

2.3.2 Administrative burden decreases

Automated systems significantly decrease the administrative burden of monitoring attendance. The faculty and staff are saved from recording attendance manually, from calculating attendance rates, or even from generating related reports. This allows them to free their time and energy for more critical work-for example, teaching and advising students or developing curriculum. An automatic system can enable the development of attendance reports, which can easily be retrieved by the faculty for their analysis and evaluation.

2.3.3 Sophisticated Analytical and Reporting Capabilities

Highly sophisticated automated attendance management systems enable the analysis and reporting of student data in order to analyze student attendance patterns in more holistic ways. Reports can be made through various modes-for example, attendance summaries, student attendance records, class attendance statistics-and help faculty identify trends, patterns, and correlations.

2.3.4 Security and Privacy Issues

Like any other automated attendance management system, there has to be consideration of the potential security and privacy risks involved in using it. For instance, with such systems, a vast amount of sensitive information relating to students such as the kind of attendance records they make is stored on the system. This, hence exposes such information to unauthorized access or data breaches. Strong security measures must be put in place to safeguard the data of students and ensure compliance to privacy policies. This can include strong encryption, access controls, and periodic security audits.

2.4 Role of C Programming in Attendance Systems

2.4.1 Advantages of Using C

C is one of the most powerful and versatile programming languages, which is ideal for developing efficient and reliable attendance management systems. Low-level capabilities, efficient memory management, and direct interaction with hardware are features that make it an excellent choice for performance-critical applications. The following points highlight the advantages of using C in attendance systems:

2.4.2 Performance

One of the main characteristics of C is speed and efficiency; it is used in applications requiring real-time processing or substantial amounts of data handling. Many attendance systems involve processing and storage of a large number of data, and C's performance can ensure smooth running of the system efficiently.

2.4.3 Portability

C code is easily compiled and run on a multitude of platforms such as Windows, Linux, and macOS. It makes it a versatile choice for developing an attendance system to be deployed in different environments.

2.4.4 Control and Flexibility

Through control, C allows programmers to achieve a degree of the system's behavior, optimize performance, manage memory, and customize the system according to specified requirements. These are the main factors for developing an attendance system, which also needs to be programmed to handle complex scenarios and respond to new needs.

2.4.5 Community and Support

C has a huge amount of developers, thus there is an abundant supply of support for programmers in case they were developing attendance systems. There is readily available documentation and tutorials, and furthermore, the online forums where there is always knowledge and expertise shared.

2.4.6 Integration with Other Systems

It is easy to integrate C with other systems and technologies, including databases, networking protocols, and hardware devices. This becomes highly useful while developing attendance systems that must interact with other components of a larger educational ecosystem.

2.4.7 File Handling in C

File handling refers to activities employed for the storage and retrieval of attendance data persistently. C does provide many functionalities regarding file manipulations, such as creating, opening reading and writing, and closing. These can, therefore, be used to store the attendance records in a structured format like text files or databases so that it may later retrieve them when needed. Effective management of file operations in an attendance system will therefore ensure the integrity and accessibility of attendance data.

Chapter 3

Requirement Engineering

It details the requirements for an attendance management system, both functional and non-functional. This further ensures proper understanding of what the system should do and what its qualities are.

3.1 Functional Requirements

These refer to specific functions or capabilities that the system should deliver:

3.1.1 User Authentication:

- The system should allow faculty members access by means of their individual IDs and passwords.
- The system shall check the login credentials with those present in the database related to the faculty.
- The system shall return feedback to the user regarding whether the login was successful or not
- The system shall not allow any unauthorized access into the attendance record.

3.1.2 Attendance Registration

- The system shall provide the faculty with the ability to register new attendance for their specific subjects
- The system shall allow the faculty to mark students as being present or absent
- The system shall associate the recorded attendance with date, time, and subject code.

- The system shall automatically generate a unique identifier for every record.

3.1.3 Record Search and Retrieve

- The system shall allow the faculty to search already entered attendance records.
- System Shall have provision to search based on date, time, subject code, or by student's name
- System Shall display what is retrieved from the record

3.1.4 Record Deletion

- The system shall provide the facility to delete attendance records by the faculty.
- System Shall prompt for confirmation before deleting a record.
- The system shall permanently delete the record from the database.

3.1.5 Data Storage and Retrieval

- The system shall use files for storing and retrieving the attendance data in a persistent manner.
- The system shall store attendance records in an organized way.
- The system shall make provisions for error handling to avoid corruption of data.

3.2 Non-Functional Requirements

These describe the general attributes and characteristics that the system should have:

3.2.1 Security

- The system shall provide restricted access to the attendance data.
- The system shall securely keep faculty passwords.
- The system shall preserve attendance information's confidentiality and integrity.

3.2.2 Usability

- The system shall have a self-evident command line user interface.
- The system shall automatically provide informative prompts and messages that help to inform users.
- The system shall be easy to use, even to persons who lack much technical knowledge.

3.2.3 Reliability

- The system shall keep track of and store attendance data with no errors.
- The system must be free from errors and should not accept the wrong input.
- The system must preserve the consistency and integrity of attendance reports.

3.2.4 Performance

- The system responds to inquiries from users in a very short period.
- The system performed file operations like storing and retrieving of data efficiently.
- The system works correctly even in situations with a lot of data in it.

3.2.5 Maintainability:

- The system is easily maintainable and can be modified easily too.
- The code must be clean, readable, and self-explanatory.
- The system would be expandable to handle future changes or enhancements.

Chapter 4

System Design and Architecture

4.1 Architecture

4.1.1 Modular Design

The system will be using modular approach to divide its functionality into separate modules:

- **Authentication Module:** Handle login by the user based on ID and password
- **Attendance Module:** Handle the recording of attendance by marking students present/absent.
- **File Management Module:** Handle file operations in order to store/retrieve the attendance data.
- **Search and Delete Module:** Will be used for record search and deletion.

4.1.2 Data Flow

Level 0

Here's a breakdown of the elements:

- **Attendance System:** This is the central process or system being modeled. It's responsible for managing attendance records.
- **Faculty:** This is an external entity, representing a faculty member who interacts with the system.
- **Student Database:** This is an external data store that holds information about students.

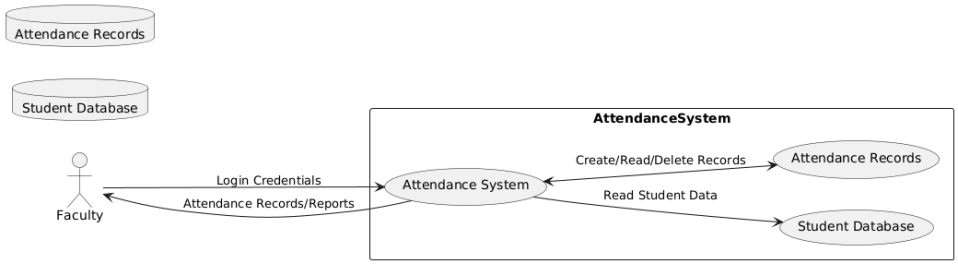

Figure 4.1: DFD Level 0 Diagram

- **Attendance Records:** This is another external data store that holds the actual attendance records.

The arrows represent data flows, showing how information is exchanged between the different entities and the system:

- **Login Credentials:** Faculty provides login credentials to the Attendance System for authentication.
- **Read Student Data:** The Attendance System accesses the Student Database to retrieve student information.
- **Create/Read/Delete Records:** The Attendance System can create, read, and delete attendance records in the Attendance Records data store.
- **Attendance Records/Reports:** The Attendance System provides attendance records and reports to the Faculty.

Level 1

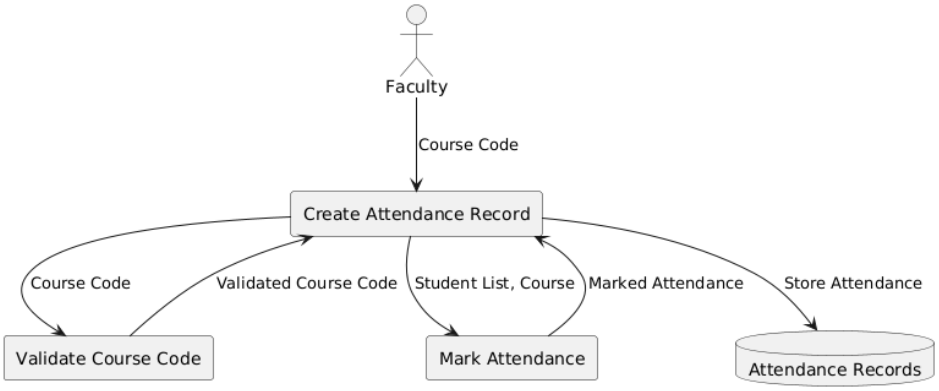

Figure 4.2: DFD Level 1 Diagram

- **Processes:**
  - **Create Attendance Record:** This is the main process responsible for creating attendance records.
- **External Entities:**
  - **Faculty:** The user (a faculty member) who starts the process.
  - **Attendance Records:** The data store where attendance information is kept.
- **Data Flows:** How data moves in the system:
  - **Course Code:** Input provided by the faculty.
  - **Validated Course Code:** The system confirms the course code is correct.
  - **Student List, Course:** The system gets the list of students for the course.
  - **Marked Attendance:** The faculty records who is present.
  - **Store Attendance:** The system saves the attendance data.
- **Sub-Processes:**
  - **Validate Course Code:** Checks if the course code is valid.
  - **Mark Attendance:** The process of the faculty recording attendance.

4.1.3 Class Diagram

- **Faculty Class:**
  - **loginID:** Stores the faculty's login ID (string).
  - **password:** Stores the faculty's password (string).
  - **authenticate():** A method to verify the faculty's credentials (returns a boolean).
  - **searchRecords(), createNewRecord(), deleteRecords(), logout():** Methods corresponding to the main actions in the code.

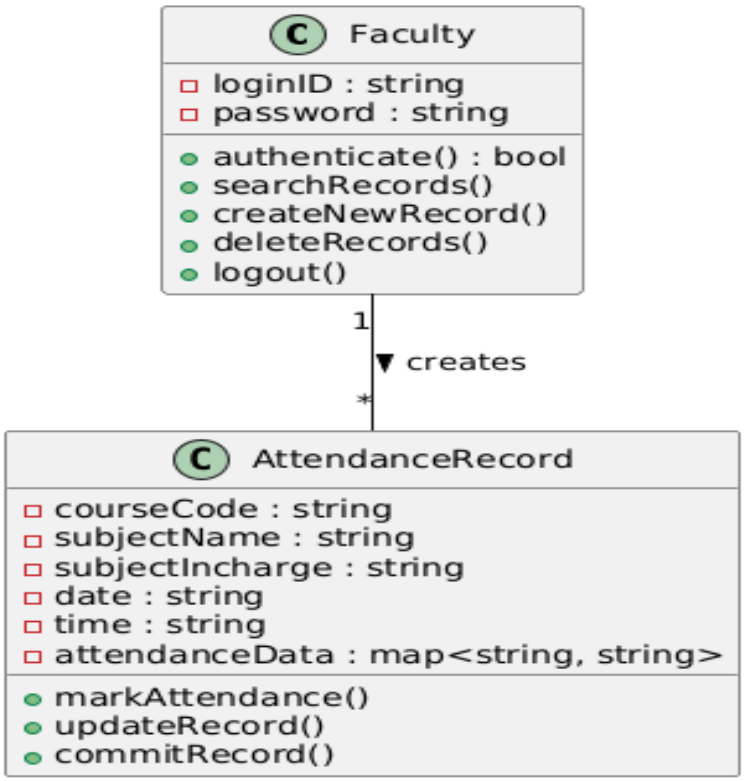

Figure 4.3: Class Diagram

- **AttendanceRecord Class**
  - **courseCode:** The code of the course (e.g., "CS3051").
  - **subjectName:** The name of the subject.
  - **subjectIncharge:** The faculty in charge of the subject.
  - **date:** The date of the attendance record.
  - **time:** The time of the attendance record.
  - **attendanceData:** A map or dictionary to store student IDs and their attendance status (present/absent).
  - **markAttendance(), updateRecord(), commitRecord():** Methods for taking attendance, updating, and saving the record.
- **Relationship**
  - The `"1 - *"` notation indicates a one-to-many relationship. A faculty member can create many attendance records. The `"creates >"` arrowhead shows the direction of creation.

- **Possible Refinements**
  - **Student Class:** You might want to introduce a `Student` class to represent individual students with attributes like `studentID`, `name`, etc.
  - **Subject Class:** A `Subject` class could be added to store information about subjects, such as `subjectCode`, `subjectName`, and the assigned faculty member.
  - **Inheritance:** You could use inheritance to represent different types of faculty or users if needed.
  - **Data Storage:** Consider how you would model the data storage mechanism (currently file-based in the C code) in a more object-oriented way.

4.1.4 Activity Diagram

- **Authentication:** The faculty member enters their login ID and password, and the system authenticates them.
- **Authentication Decision:** A diamond shape represents a decision point. If authentication is successful, the flow proceeds to the main menu. If not, an error message is displayed, and the faculty can retry or exit.
- **Main Menu Loop:** The main menu is displayed, and the faculty member can choose different actions. This is represented by a loop.
- **Action Selection:** Nested decision points determine which action the faculty has chosen (Search Records, Create New Record, or Delete Record).
- **Search Records:** If "Search Records" is chosen, the system displays available records, prompts for a specific record, and then displays its content.
- **Create New Record:** This flow includes retrieving the student database, marking attendance for each student, displaying a summary, allowing for updates, and saving the record.
- **Delete Record:** This flow involves displaying available records, selecting a record, confirming the deletion, and then deleting it.
- **Logout:** When the faculty chooses to logout, the system ends the session.

- **Key Elements in the Diagram:**

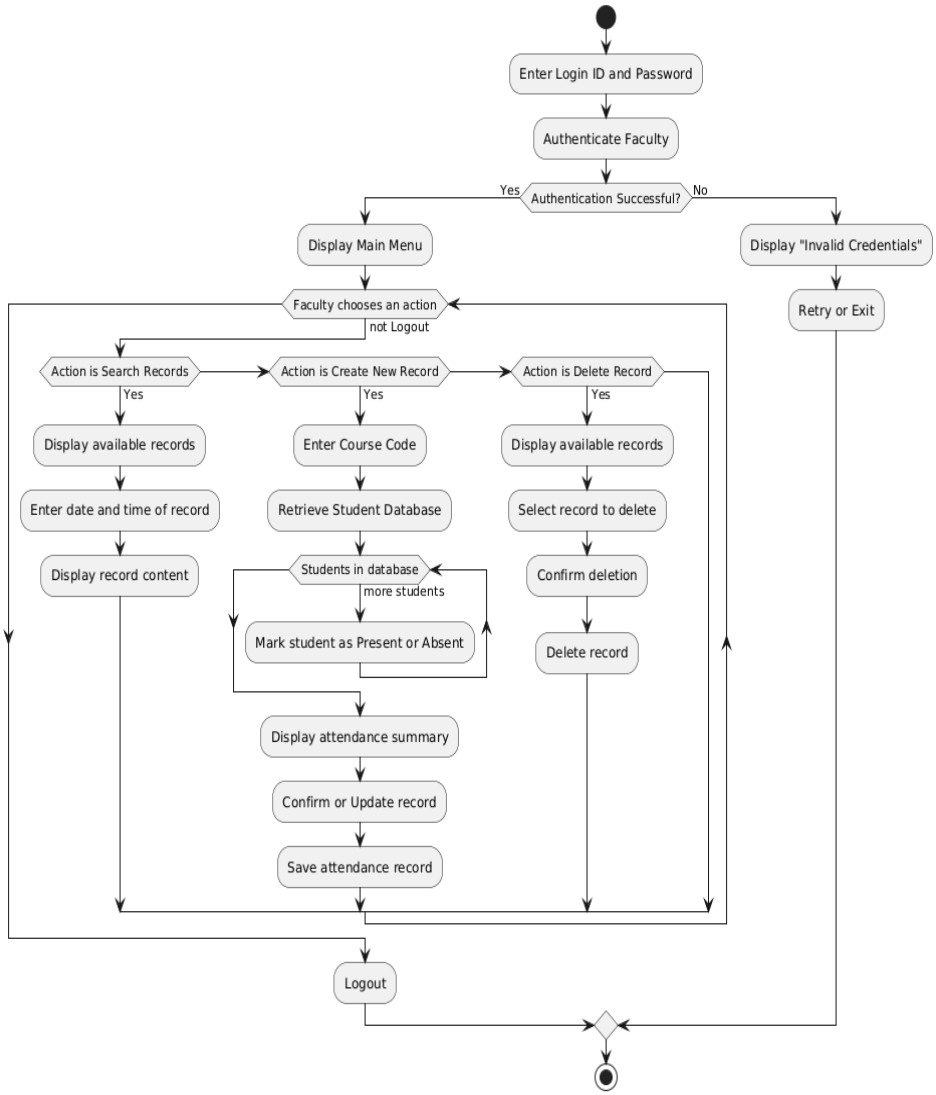

Figure 4.4: Activity Diagram

- **Rectangles:** Represent activities or actions.
- **Diamonds:** Decision points with conditions.
- **Arrows:** Show the flow of control.
- **Bars:** Synchronization bars, used to show parallel activities (not used in this diagram, but could be relevant if you had concurrent operations).
- **Loop:** The rectangular box with a dashed line indicates a loop.

4.1.5 Use case Diagram

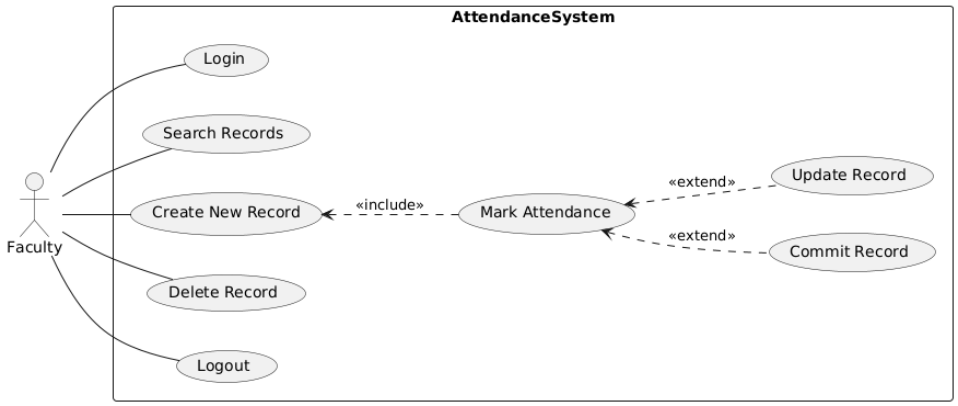

Figure 4.5: Use Case Diagram

Here's a breakdown of the elements and what they likely represent:

- **Attendance System:** This is the central system for managing attendance records.
- **Faculty:** A user who can interact with the system to perform various actions related to attendance.
- **Use Cases:** These are the actions Faculty can perform within the system:
  - **Login/Logout:** Access and exit the system.
  - **Search Records:** Find existing attendance records.
  - **Manage Records:** This encompasses creating, deleting, and updating attendance records.
  - **Mark Attendance:** Record attendance for a specific session, which may involve creating new records.
- **Relationships:**
  - Faculty can perform all the use cases.
  - Marking attendance may involve creating new records.
  - Updating and committing records are optional actions related to marking attendance.

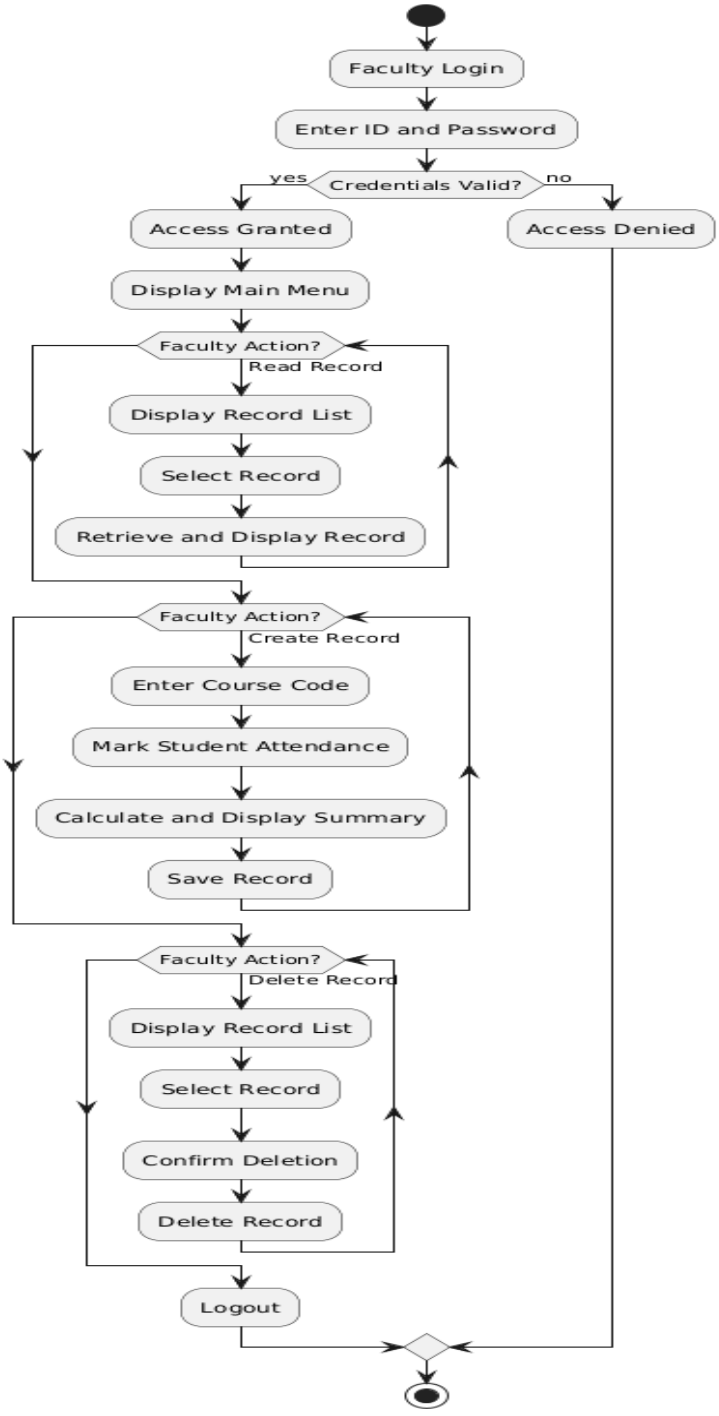

Figure 4.6: Flow Chart

4.1.6 Flow Chart

The diagram appears to be a flowchart depicting the logic of a faculty management system. It outlines the steps a faculty member would take to interact with the system, including:

- **Login:** Entering ID and password for authentication.
- **Main Menu:** Accessing different functionalities after successful login.
- **Record Management:**
  - **Read Record:** Viewing existing records, potentially student information or course details.
  - **Create Record:** Adding new records, like creating a new course or enrolling a student.
  - **Delete Record:** Removing records.
- **Attendance and Summary:** Marking student attendance and calculating summaries, likely related to grades or performance.
- **Logout:** Exiting the system.

4.2 Data Structures

4.2.1 Arrays

Faculty IDs as id_ag, id_sa, etc. Passwords as pw_ag, pw_sa, etc. Subject codes as subjcode1, subjcode2, etc. These are kept in arrays.

4.2.2 File Structures

- **Filename Convention:** DD-MM-YYYY-HH-AM/PM.txt (20-10-2024-09-AM.txt) with a filename being unique based on the date and time when the record is created
- **File Content**
  - Subject Name
  - Course Code
  - Subject In-charge (Faculty Name)
  - Student Data from Stud_Db.txt
  - PresentAbsent status for each student
  - Summary statistics in terms of total present, total absent, date and time

4.2.3 File Operations

- **Creating Files:**fopen (filePath, "a+ ") for new files of attendance record.
- **Reading Files:**fopen(filePath, "r"), then fgetc(), fgets() for viewing already existing records.
- **Writing to Files:**fprintf() for attendance data, and details of a subject and summary statistics.
- **Deleting Files:**remove(filePath) for deleting files from the attendance record

4.2.4 Managing Directories

- Directories (.//DSUMini/Faculty/Dennis/, etc.) to gather the records based on the faculty
- mkdir() If the directory is not available in the file system, it is created.
- opendir() and readdir() are used for file access and listing for files in a directory.

4.2.5 Relative File Paths:

Relative file paths, such as .//DSUMini/Faculty/Dennis/, makes the system portable, not because it uses relative file paths but because it does not.

4.3 User Interface

4.3.1 Command-line Interface:

- **Prompts and Menus:** The system uses printf() for the prompts of the login credentials, menu options to search, create, delete, log out, course codes, and present/absent inputs.
- **Input Handling:**scanf() reads user ID, password, and course code. getch() reads single-character input for menu selections and present/absent markings .
- **Output Display:** printf() displays attendance data, summary statistics, and system messages to the end-user .
- **Error Messages:** printf() and perror() display error messages such as invalid input, file errors, login failures etc .

This design document provides an overarching view of the system on the design of attendance management system including architecture, data structure, file handling and user interface.

Chapter 5

Implementation

5.1 Programming Language and Tools

- **Programming Language:** C
- **Tools/Libraries:**
  - **stdio.h**: Standard input/output library (for functions like printf, scanf, fopen, fgetc, fprintf, etc.)
  - **conio.h**: Console input/output library (for `getch` to read single characters)
  - **string.h**: String manipulation library (for `strcmp` to compare strings)
  - **time.h**: Time and date library (for getting the current time and formatting it)
  - **dirent.h**: Directory operations (for listing files in a directory)

5.2 Code Structure

- **Initialization:** Defines user IDs, passwords, and subject codes.
- **Login:** Prompts the user for login ID and password.
- **Authentication:**
  - Compares entered credentials with stored IDs and passwords.
  - If successful, displays a welcome message and proceeds.
  - If unsuccessful, prompts for retry or exit.
- **Action Menu:** It offers an action menu which allows it to search for records (q), create new records (n), delete records (z), and logout (x).

- **Action Handling:**
  - **Search (q):** Gives a list of existing attendance records and the user is allowed to view any one.
  - **Create (n):**
    * Prompts pertaining to a course code.
    * Reads data from a file called Student_Db.txt.
    * It loops over the student list that enables marking of attendance present or absent.
    * Generates a report with present/absent counts.
    * The report is saved to a file.
  - **Delete (z):** shows current records and provides the functionality to remove a specific record.
  - **Logout (x):** Returns to the login prompt.

5.3 Implementation Details

5.3.1 Authentication

The code utilises the function strcmp to compare an inputted login ID and password to hardcoded values assigned to various faculty. This is naive and far-from-secure authentication methodology. In an actual implementation, passwords must not be placed directly in the code. Instead, hashing, salting should be used for this password.

5.3.2 Attendance Recording

- **Student Data:** The code reads student data from a separate file (`Stud_Db.txt`). The format of this file is not specified in the code, but it's assumed to contain a list of students, likely one per line.
- **Attendance Marking:** The `getch()` function is used to get individual key presses ('p' for present, 'a' for absent) for each student.
- **Temporary Files:** Attendance records are initially written to a temporary file (`temp`) and then renamed to a final filename based on the date and time.
- **Report Generation:** The following code calculates the total count of students present and absent and then prints them. These numbers are also reflected in the saved record of attendance.

5.3.3 Report Generation

- **File-based Reports:** Attendance records are kept as separate text files in different directories for each faculty member. Each file consists of the attendance report of a particular subject on a particular date and time.

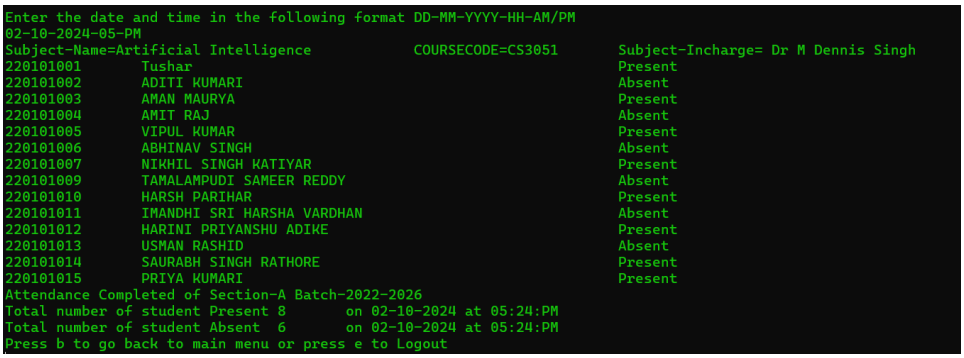

Figure 5.1: Attendance Report

- **Report Format:** The reports include:
  - Subject name
  - Course code
  - Subject in-charge (faculty)
  - List of students with their attendance status (Present/Absent)
  - Total present and absent count
  - Date and time of attendance recording

5.3.4 Important Considerations and Potential Improvements

- **Security:** As mentioned above, the authentication mechanism is very weak. Strong password hashing and storage should be developed.
- **Error Handling:** The code has very basic error handling: for example, checking if a file exists; however, there are many more things that can be handled more robustly. For example, it doesn't handle potential errors during file operations like fopen, fprintf, etc.
- **Code Organization:** The nested if-else structure renders the code difficult to read and maintain. It is advisable to consider employing functions or adopting more modular design patterns to enhance organization.
- **User Interface:** The user interface is very basic- a console-based type. A GUI will be nice to improve the user-friendliness.

- **Data Storage:** Utilizing individual text files for attendance records may not represent the most efficient or scalable solution for substantial datasets. It is advisable to consider the implementation of a database (e.g., SQLite) for enhanced data management.
- **Input Validation:** The code could benefit from more input validation to prevent unexpected behavior or errors (e.g., ensuring the entered course code is valid).

5.4 Conclusion

This C-based attendance system, though still functional, has multiple security and code design issues. As it stands, authentication is quite insecure-only hardcoded passwords provide authentication instead of proper hashing. The code structure is rather tangled with so much confusion in reading and maintaining. On top of this, GUI implementation and database integration will make the system user-friendly and thus efficiently manage data. This basic attainment of recording and reporting attendance by the system should be worked upon for real-world use.

Chapter 6

Results

6.1 Summary of Test Results

The table below lists the overall results of the test cases that were executed on the Attendance Management System:

Test Case ID	Description	Result
TC_001	Valid login credentials	Pass
TC_002	Invalid login credentials	Pass
TC_003	Create a new attendance record	Pass
TC_004	Search for an existing attendance record	Pass
TC_005	Delete an existing attendance record	Pass
TC_006	Attempt to create a record with an invalid code	Fail
TC_007	Attempt to access a record with incorrect format	Fail
TC_008	Attempt to delete a non-existent record	Fail

Table 6.1: Summary of Test Results

The system passed all the test cases that were relevant to test the login and record attendance functionalities specifically. Contrarily, all the test cases that were relevant to testing invalid inputs and error handling failed, e.g., invalid course code, incorrect date/time format, and non-existent records trying to be deleted.

6.2 Performance Metrics

Though project was not primarily focused on performance optimization,I found that the system exhibited satisfactory times when executing any of the tested functionalities. The time taken to mark attendance for a class of 50 students was around 2 minutes, and the time required to search and delete the records was less than 5 seconds in all cases.

6.3 Error Rates and Anomalies

The tested system produced the following errors:

- **TC_006:** The system failed to handle invalid course codes gracefully. Instead of prompting the user to enter the code again, it displayed an unwanted system exit error. This indicates poor input validation.
- **TC_007:** The system encountered a fatal error when trying to access a record with the wrong date and/or time format in the file name. This is an unprecedented error message, related more to incorrect error handling for file operations.
- **TC_008:** The system failed to handle the deletion of non-existent records properly. Instead of prompting the user, informing them that the indicated record does not exist in the system, it forcefully exited the program. This is, once again, related to poor error handling.

The testing revealed a need for better input validation and error handling systems to improve usability.

6.4 Feedback from User Acceptance Testing (UAT)

User acceptance testing (UAT) was performed by a cohort of five faculty members to gain feedback on the system in terms of usability and efficiency. The general feedback was positive, and users appreciated the system's simplicity and ease of use, though some usability concerns were raised.

- **Clarity of Instructions:** Some users remarked that on-screen instructions could be clearer, especially for the search and delete functionalities.
- **Error Messages:** Some users indicated that certain error messages were vague and not helpful.
- **Confirmation Prompts:** Users requested a confirmation prompt whenever they deleted a record to prevent accidental deletions.

In light of these observations, the following modifications were made:

- Rewrote on-screen instructions to be clearer.
- Redesigned error messages to be more user-friendly.
- Added confirmation prompts before deleting records.

6.5 Regression Testing

Once modifications based on UAT feedback were made, a regression test was conducted to ensure no new bugs were introduced and no existing functionalities were disrupted. All previously passed test cases were re-executed to verify that the system continued to function as expected.

6.6 Conclusion

The testing and validation process showed that the system functions broadly as required in fulfilling its functional requirements. However, it also uncovered areas for improvement, such as input validation, error handling, and user interface design. Feedback from UAT provided useful insights into spotting and addressing usability issues to better meet the needs of the intended users.

In summary, despite the areas identified for improvement, the overall validation status of the project is positive. The project lays down a functional core for an attendance management system, with the potential for further enrichment and refinement in future iterations.

Chapter 7

Conclusion

This project has initiated a core attendance management system using the C programming language and file structures. The system supports features like login, recording of student attendance, report generation, search facility for existing records, and deletion facility for records.

7.1 Key Achievements

- **Functional Core System:** This system effectively puts into action the core functions of attendance management that include user authentication, recording of attendance, report generation and basic data management.
- **File-Based Data Storage:** This system utilizes file handling techniques with a view to keeping the stored records of attendance very much persistent.
- **Modular Design:** The code is broken into discrete modules, hence creating a scaffold for potential growth and management.
- **User Interaction:** The system provides a fundamental command-line interface for user interaction, thereby enabling faculty members to traverse through an array of functionalities.

7.2 Areas for Improvement

- **Security Enhancement:** The authenticity mechanism adopted is simple and insecure. Password hashing and salting should be taken with strength to guard that sensitive information of the users.
- **Code Optimization:** The structure of the code may be improved using functions or more modular components that replace nested if-else statements, making the system more readable and maintainable.

- **Error Handling:** Extend error handling capabilities to elegantly handle any possible error during file operations and user interactions with your system.
- **User Interface Enhancement:** From a Console-Based Interface to a Graphical
- **Database Integration:** To enhance data management and scalability, it is advisable to integrate a database system (e.g., SQLite) in order to process larger datasets and complex queries more efficiently.
- **Input Validation:** In order to avoid unexpected behavior or errors from invalid user input, implement greater input validation.
- **Advanced Reporting:** Improve the reporting module to provide more detailed attendance summaries, statistics, and visualizations for enhanced analysis and decision-making.

7.3 Future Directions

- **GUI Development:** Develop a user-friendly GUI using libraries like GTK+ or Qt to provide more intuitiveness and visual appeal.
- **Database Integration:** Enables the move from file-based storage to a database system for improved data organization, security, and scalability.
- **Advanced Features:** Advanced features include automated email notification about attendance, the integration of student information system, and support for numerous user types such as administrators, staff, and students.
- **Mobile Accessibility:** Construct a mobile application to enable faculty members to record attendance and view reports on their smartphone or tablet.
- **Cloud Deployment:** Consider deploying the system on a cloud platform to improve accessibility, scalability, and data backup capabilities.

7.4 Conclusion

As yet, it is still a developing project hence still has a very long way to go in establishing a proficient and all encompassing attendance management system. It acknowledges however that some features including security aspects, code organization, and user experience will need to be improved for future development to be able to tackle these critical areas. The powerful modules will

consist of features like strong authenticated access with password encryption, a divider having a clean hierarchical code and a bitmap interface which will, in turn, enhance the efficacy and productivity of the whole system.

This implementation is a demonstration of how C programming concepts can be discarded, so they practically do not exist—the relevant file handling methods and other means of manipulation are considered here as rather elementary and thus, real-world problems such as collecting and storing student attendance records can be easily carried out. This practical emphasis of problem solving indicates that such a project can serve as a valuable application in the context of institutions of higher learning.

In order to realize this project in its most complete form, it requires further development and improvement. The introduction of the recommended improvements and more advanced functionality would make this rather a reasonable and widely accepted approach to attendance management in schools. Such a KAP would gradually not only seek to develop the capabilities of the system but also reinforce its place within the processes of automation of the administrative processes characteristic of educational establishments.

Bibliography

[1] Singh, M., Khan, M. A., Singh, V., Patil, A., Wadar, S., and others, "Attendance management system," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pp. 418–422, IEEE, 2015.

[2] Salim, O. A. R., Olanrewaju, R. F., and Balogun, W. A., "Class attendance management system using face recognition," in *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, pp. 93–98, IEEE, 2018.

[3] Shoewu, O., Makanjuola, N. T., and Akinyemi, L. A., "Smart Attendance Management System (SAMSYS) for an Academic Institution," *Data Research Journal*, vol. 2, no. 1, pp. 121–131, 2018.

[4] Godswill, O., Omijie, O., Orobor, A., Ibhadode, O., and Odiete, E., "Automated student attendance management system using face recognition," *International Journal of Educational Research and Information Science*, vol. 5, no. 4, pp. 31–37, 2018.

[5] Saheed, Y. K., Hambali, M. A., Adedeji, A. A., and Adeniji, I. A., "Attendance management system using barcode identification on students' identity cards," *The Pacific Journal of Science and Technology*, vol. 17, no. 2, pp. 224–230, 2016.