

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2023

Smart Attendance GUI Application

Dheeraj Maddi

Follow this and additional works at: <https://opus.govst.edu/capstones>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

SMART GUI APPLICATION

By

Dheeraj Maddi

B.E., Jawaharlal Nehru Technological University, 2021

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science



Governors State University
University Park, IL 60484

2023

ABSTRACT

Why is this project important?

The Smart Attendance GUI Application is essential for modernizing and simplifying attendance tracking in various settings, such as workplaces and educational institutions. It leverages innovative face recognition technology to enhance precision and efficiency, reflecting a commitment to harnessing technology for improved processes.

What problem does it solve?

The project addresses the time-consuming and error-prone nature of manual attendance tracking. By implementing face recognition technology, it automates the attendance process, eliminating the need for traditional methods. This not only saves time but also significantly reduces the chances of errors in attendance records. Additionally, the inclusion of manual entry accommodates situations where face recognition may not be feasible or appropriate.

Does it provide a new application/service, replace an existing application/service, or enhance an existing application/service?

The Smart Attendance GUI Application provides a new and innovative solution for attendance tracking. While it doesn't necessarily replace existing methods entirely, it enhances the process significantly by introducing automation through face recognition. The manual attendance entry feature further adds flexibility, ensuring adaptability to diverse situations.

What is the application/service date and applicable release?

The specific application or service date is not provided in the description. However, it emphasizes the use of face recognition technology and a user-friendly graphical interface, suggesting a contemporary and user-centric design. Additionally, the release date is not mentioned, so it's unclear when this solution will be available for implementation. To provide a comprehensive understanding, more details on the development and release timeline would be beneficial.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	2
1.4	Future Enhancements	3
1.5	Definitions and Acronyms	3
2	<i>Technical Description</i>	4
2.1	Application Architecture	4
2.2	Application Information Flows	6
2.3	Capabilities	6
3	<i>Project Requirements</i>	7
3.1	Identification of Requirements	7
3.2	Release and Transition Plan	8
4	<i>Project Design Description</i>	8
5	<i>Project Internal/external Interface Impacts and Specification</i>	13
6	<i>Project Design Units Impacts</i>	14
7	<i>Open Issues</i>	15
8	<i>Acknowledgments</i>	15
9	<i>References</i>	16
10	<i>Appendices</i>	18

1 Project Description

The Smart Attendance GUI Application presents a cutting-edge approach to simplify attendance monitoring in workplaces and educational settings. This innovative solution incorporates face recognition technology to automate attendance, ensuring accuracy and efficiency. In situations where face recognition may not be suitable, the application offers a manual attendance entry option. The user-friendly graphical interface caters to individuals with varying technical proficiency, facilitating easy navigation. Beyond attendance tracking, the application includes features such as student registration, instructor data management, and an alert system to communicate attendance-related information to students. The utilization of this tool streamlines attendance tracking, making the process straightforward and trouble-free.

1.1 Competitive Information

Smart Attendance GUI Application exhibits several key features that position it as a strong contender against competitors like Buddy Punch and Kronos. Here's how your project showcases potential advantages:

Innovative Face Recognition Technology:

Differentiator: While Buddy Punch and Kronos may offer traditional time and attendance tracking, your application stands out with the integration of face recognition technology. This innovation ensures a more secure, accurate, and efficient attendance tracking process, setting your project apart from competitors.

User-Friendly Graphical Interface:

Differentiator: Emphasize the user-friendly graphical interface, making it accessible to users with varying technical expertise. If your application is more intuitive and easier to navigate compared to competitors, this becomes a significant advantage, especially in environments with diverse user skill levels.

Flexibility with Manual Attendance Entry:

Differentiator: Your project's provision for manual attendance entry adds flexibility, addressing situations where face recognition may not be appropriate. This flexibility could be a key advantage over competitors that may not offer such alternatives.

Comprehensive Student Management Features:

Differentiator: Highlight the comprehensive nature of your application, covering not only attendance tracking but also student registration and data management for instructors. If Buddy Punch and Kronos lack these additional features, it positions your application as a more holistic solution for educational institutions.

Alert System for Students:

Differentiator: The inclusion of an alert system to inform students about attendance-related information adds an extra layer of communication. If your competitors lack this feature, it can be a unique selling point, enhancing the overall user experience for students.

Competitive Positioning:

Emphasize the potential advantages in terms of cost, implementation speed, and ongoing support. If your project can offer a more cost-effective solution, a quicker implementation timeline, or superior customer support, these factors contribute to a favorable competitive positioning.

1.2 Relationship to Other Applications/Projects

Smart Attendance GUI Application has the potential to relate to various projects and applications in both educational and professional settings. Here are some potential relationships:

Integration with Learning Management Systems (LMS):

The Smart GUI application, with its focus on student registration and attendance tracking in educational institutions, could potentially integrate with Learning Management Systems. This integration could provide a seamless experience for both students and instructors by consolidating attendance data with other educational information.

Attendance Analytics and Reporting Tools:

This project, which involves automated attendance tracking, may relate to projects or applications that focus on attendance analytics and reporting. This could be beneficial for institutions or organizations seeking insights into attendance trends and patterns.

Communication and Notification Applications:

The alert system within this application aligns with projects focused on communication and notification systems. It could relate to applications that handle automated messaging, alerting, or communication within educational or organizational settings.

1.3 Assumptions and Dependencies

- **Assumptions Made:**
 - **Availability of Face Recognition Technology:** The successful implementation of face recognition assumes the availability and reliability of face recognition technology. The project relies on this technology for automated attendance tracking, and any issues with its availability or accuracy may impact project outcomes.
 - **Integration with Existing Systems:** Assumptions are made regarding the seamless integration of the Smart Attendance GUI Application with existing systems in educational institutions or workplaces. This includes compatibility with learning management systems, HR systems, and other relevant platforms.
 - **User Acceptance of Face Recognition:** The assumption is that users, both in educational and professional settings, will accept and adapt to the utilization of face recognition technology for attendance tracking.
- **Dependencies on Other Projects, Applications, Services, or Capabilities:**
 - **Integration with Learning Management Systems (LMS):** The successful integration of the Smart Attendance GUI Application may depend on the capabilities and timelines of existing or concurrent projects related to Learning Management Systems.
 - **Biometric Technology Providers:** Dependencies exist on the providers of face recognition technology. Any changes or developments in the underlying biometric technology may impact the functionality and performance of the attendance tracking feature.
 - **Communication and Notification Services:** The alert system within the application relies on the availability and proper functioning of communication and notification services. Dependencies on third-party services for email or text alerts need to be considered.
- **Required Development and Changes in Customer Operational Procedures:**
 - **Training for Face Recognition Adoption:** Successful adoption of face recognition technology may require training sessions for users to understand and utilize the new functionality effectively. Training programs need to be developed and executed to support the adoption of this innovative feature.
 - **Update in Data Management Procedures:** The application introduces features such as student registration and data management for instructors. Any necessary changes in data management procedures at educational institutions or workplaces should be communicated and implemented.
 - **Integration Protocols for Existing Systems:** For smooth integration with existing systems, predefined protocols and standards may need to be established. This could involve coordination with IT teams to ensure compatibility with learning management systems, HR systems, and other relevant platforms.
 - **Communication Strategy for Alerts:** The implementation of the alert system requires a well-defined communication strategy. This involves informing users (students and instructors) about the new alert system, its purpose, and the actions they need to take based on the received alerts.

1.4 Future Enhancements

The Smart Attendance GUI Application is designed with flexibility and scalability in mind, allowing for future enhancements to meet evolving needs. Here are some planned or possible future enhancements for the project:

Enhanced Security Features:

Strengthen the security features of the application, including encryption protocols for data transmission and storage. Consider implementing multi-factor authentication for added security in accessing sensitive attendance and student information.

Mobile Application for On-the-Go Access:

Develop a mobile application companion to the Smart Attendance GUI Application. This would allow students, instructors, and administrators to access attendance information, receive alerts, and perform related tasks on the go.

Real-Time Collaboration Tools:

Integrate real-time collaboration tools within the application to facilitate communication between students and instructors. This could include chat features, discussion forums, or collaborative document sharing.

Enhanced Student Engagement Features:

Develop features to enhance student engagement, such as gamification elements, rewards systems, or interactive modules. This can contribute to a more engaging and participatory learning environment.

Voice-Activated Commands:

Explore the integration of voice-activated commands for hands-free operation, especially in scenarios where manual attendance entry is needed. This can improve accessibility and user convenience.

1.5 Definitions and Acronyms

Acronym items should be included here. For each special term supply a definition here. Certainly! Here are definitions for acronyms and special terms related to the Smart Attendance GUI Application:

GUI:

Definition: GUI stands for Graphical User Interface. It refers to the visual elements, such as icons, buttons, and menus, that allow users to interact with a computer program or system.

LMS:

Definition: LMS stands for Learning Management System. It is a software application used to administer, deliver, and track educational courses and training programs.

HRMS:

Definition: HRMS stands for Human Resource Management System. It is software that combines various HR functions, such as payroll, benefits administration, and employee records, into a unified system.

Biometric:

Definition: Biometric refers to the measurement and statistical analysis of people's unique physical and behavioural characteristics. In the context of the Smart Attendance GUI Application, it involves the use of face recognition for attendance tracking.

HR:

Definition: HR stands for Human Resources. It is the department within an organization responsible for managing personnel-related functions, such as recruitment, employee relations, and training.

2 Project Technical Description

The Smart Attendance GUI Application utilizes Python, OpenCV for face recognition, Tkinter for a user-friendly interface, and MySQL for data storage. Additional tools include Pandas for data processing, Pillow for image processing, and smtplib for email notifications. The minimum requirements to run this application are Windows 7, Intel i3 processor, 720p camera, 8GB RAM, and 100GB hard disk. The technical design prioritizes simplicity, accuracy, and compatibility, ensuring an efficient and accessible attendance-tracking solution.

2.1 Application Architecture

The Smart Attendance GUI Application, enriched with an email module, maintains a modular and scalable architecture, integrating various technologies and tools to achieve its functionalities. The system architecture includes components such as the face recognition module, graphical user interface (GUI), database management, email module, and supporting libraries.

Components:

Face Recognition Module:

Technology: OpenCV (Computer Vision Library) with Python.

Description: Responsible for capturing, processing, and recognizing faces during attendance tracking. OpenCV provides tools and algorithms for face detection and recognition.

Graphical User Interface (GUI):

Technology: Tkinter (Python GUI toolkit).

Description: Utilized to create a user-friendly interface, allowing seamless interaction. Features include student registration, attendance tracking, and data management.

Database Management:

Technology: MySQL (Relational Database Management System).

Description: Stores and manages data related to student registration, attendance records, and instructor data. Ensures data integrity and provides a structured storage solution.

Data Processing:

Libraries: Pandas

Description: Pandas handles data processing tasks, facilitating organization and manipulation of attendance data for efficient management and analysis.

Image Processing:

Libraries: Pillow

Description: Pillow supports image processing tasks, including image capture and preprocessing before face recognition, ensuring the quality of images for attendance tracking.

Email Module:

Library: smtplib (Simple Mail Transfer Protocol)

Description: Integrates smtplib to enable email notifications. This module communicates attendance-related information, providing alerts and updates to students and instructors.

Technical Requirements:

Minimum Requirements:

Software:

Operating System: Windows 7

IDE: Visual Studio Code, PyCharm

Hardware:

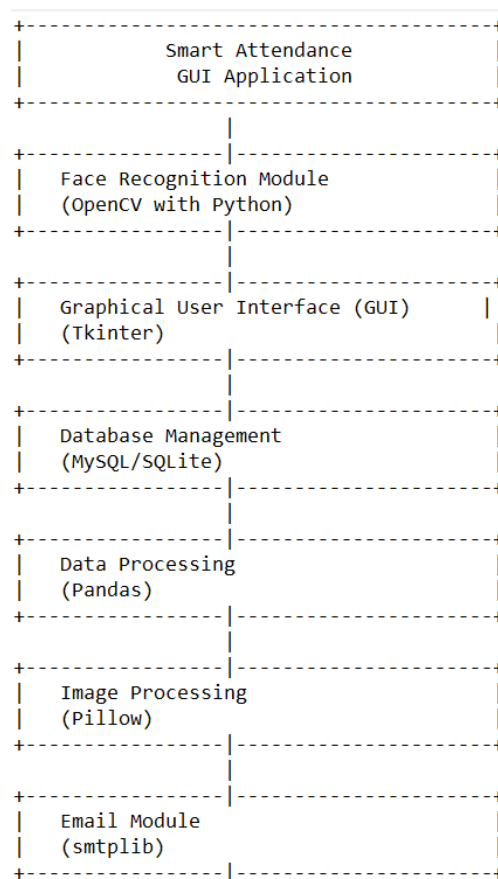
Processor: Intel i3

Camera: 720p (1280 x 720)

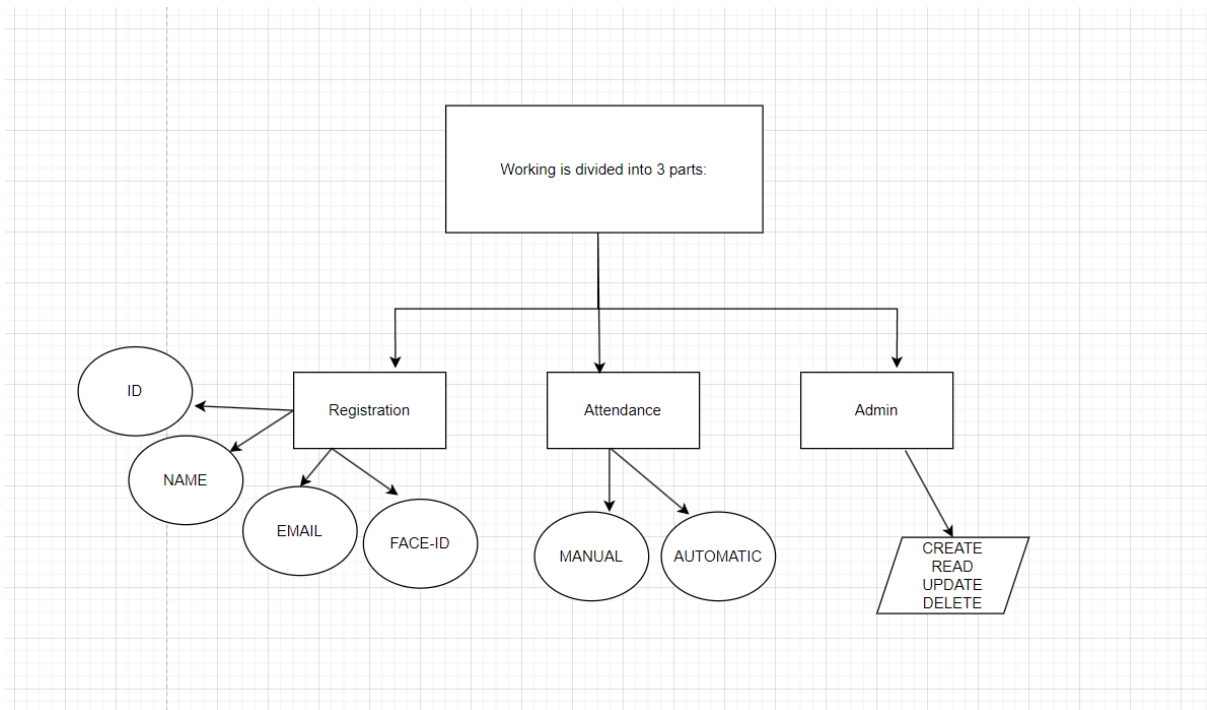
RAM: 8GB

Hard Disk: 100 GB

System Architecture Diagram:



2.2 Application Information Flows



2.3 Capabilities

Face Recognition:

Detect and recognize faces accurately for automated attendance tracking.

Graphical User Interface (GUI):

Create an intuitive and user-friendly interface using Tkinter for easy navigation.

Database Management:

Store and manage student, attendance, and instructor data.

Support operations such as retrieving, adding, updating, and deleting data.

Data Processing:

Utilize Pandas for efficient organization and manipulation of attendance data.

Image Processing:

Employ Pillow for image processing tasks, ensuring image quality for face recognition.

Email Notification:

Integrate smtplib for sending email notifications, providing alerts and updates.

Security:

Implement security measures to safeguard sensitive attendance and user data.

Real-Time:

Provide real-time face recognition and attendance tracking for timely and accurate data.

Notification Customization:

Allow customization of email notifications and alerts based on user preferences.

3 *Project Requirements*

3.1 *Identification of Requirements*

<Project-SmartAttendance-1_TechReq-0001>

Requirement: The application must utilize Python for the implementation of core functionalities.

SMART Criteria:

Specific: Implement core functionalities using the Python programming language.

Measurable: Ensure that most of the application codebase is written in Python.

Achievable: Python is a widely used and accessible language, making this requirement achievable.

Realistic: Given the selected technologies, implementing core functionalities in Python is realistic.

Time-related: From the beginning of the project until completion, Python will be the primary language.

<Project-SmartAttendance-1_TechReq-0002>

Requirement: The application must incorporate OpenCV for face detection and recognition.

SMART Criteria:

Specific: Implement face detection and recognition using the OpenCV library.

Measurable: Ensure accurate face detection and recognition capabilities.

Achievable: OpenCV is a well-established library for computer vision tasks.

Realistic: Utilizing OpenCV for face recognition aligns with the project's goals.

Time-related: Throughout the entire development lifecycle, OpenCV will be employed.

<Project-SmartAttendance-1_TechReq-0003>

Requirement: The application must feature a graphical user interface (GUI) developed using Tkinter.

SMART Criteria:

Specific: Implement a user-friendly GUI using the Tkinter toolkit.

Measurable: Ensure a seamless and intuitive user interface.

Achievable: Tkinter is a powerful and accessible GUI toolkit for Python.

Realistic: The project's objectives include providing a user-friendly interface.

Time-related: The GUI development using Tkinter will occur from project initiation to completion. Release and Transition Plan

3.2 Release and Transition Plan

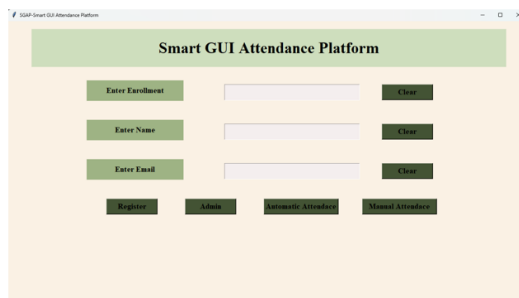
The deployment of the Smart Attendance GUI Application will follow a structured release and transition plan to ensure a smooth and efficient transition for end-users. The plan encompasses the steps involved in making the application available to customers or transitioning from a current release to a newer one.

- Release Planning
- Communication and Training
- Documentation
- Deployment
- Monitoring and Feedback
- Rollback Plan
- Post-Deployment Support
- Version Control and Documentation:
- Customer Engagement

4 Project Design Description

Output Screens:

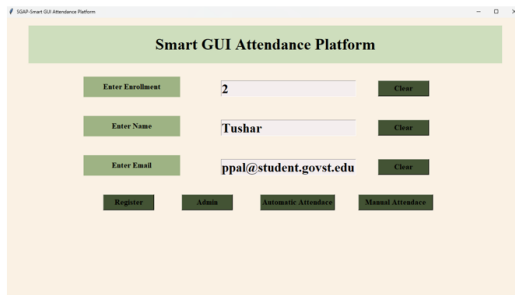
HOME PAGE



- Student details (Enrollment, Name and Email) for registration
- Admin can view registered students.
- Attendance can be captured manually or by face detection.

Figure 4.1: Home Page

DEMO-STUDENT REGISTRATION

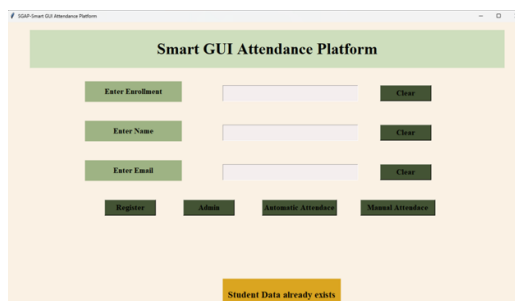


The screenshot shows a web application titled "Smart GUI Attendance Platform". It features three input fields for registration: "Enter Enrollment" with the value "2", "Enter Name" with the value "Tushar", and "Enter Email" with the value "ppal@student.govst.edu". Each input field has a "Clear" button to its right. Below these fields are four buttons: "Register", "Admin", "Automatic Attendance", and "Manual Attendance".

- In this picture student details are entered.
- After clicking on register button, it will capture the face and save all the details.
- Once the registration is successful, students gets notified via email.

Figure 4.2: Student Registration Demo

STUDENT VALIDATION

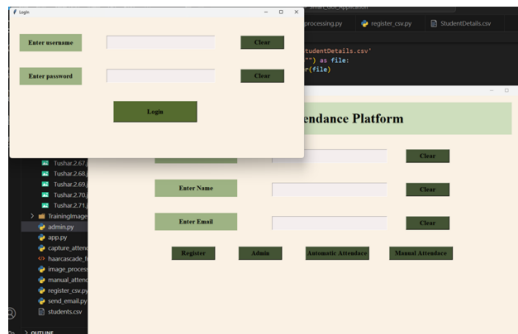


The screenshot shows the same "Smart GUI Attendance Platform" interface as Figure 4.2, but with empty input fields for Enrollment, Name, and Email. At the bottom of the form, a yellow error message box displays the text "Student Data already exists".

- Student can not register twice.
- An error message will pop up "Student Data already exists".

Figure 4.3: Student Validation

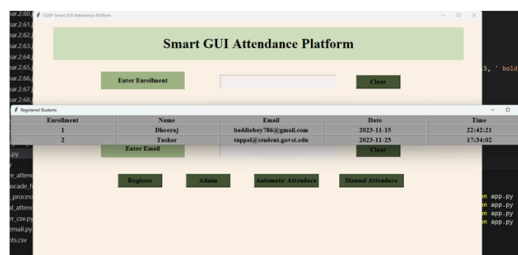
ADMIN



- After moving into admin window, credentials are entered, and access will be granted.
- Admin will be able to view the list of registered students.

Figure 4.4: Administrator

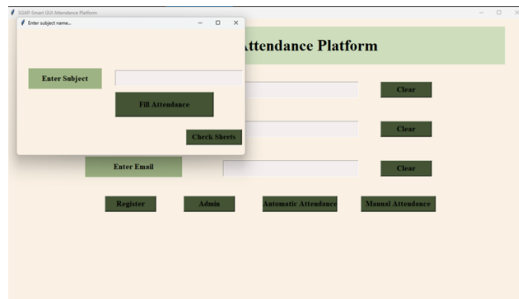
REGISTERED STUDENTS



- This picture depicts the list of registered students.

Figure 4.5: Registered Students

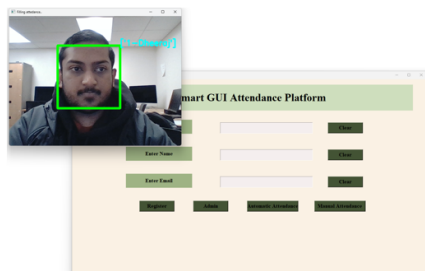
ATTENDANCE



- This application has two types of attendance.
 1. Automatic attendance(Face detection)
 2. Manual attendance
- For both these types, subject name should be entered.

Figure 4.6: Attendance

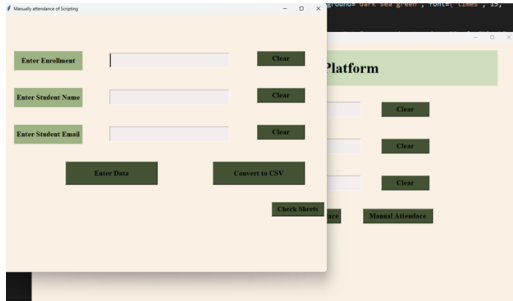
AUTOMATIC ATTENDANCE



- A camera will open to detect a registered face.
- Upon a successful face match, the system will record attendance and send a notification via email.

Figure 4.7: Automatic Attendance

MANUAL ATTENDANCE



- Student details(Enrollment, Name and Email) must be entered, It will save all the details into the database as well as CSV file.
- Once the attendance is saved, student gets notified via email.

Figure 4.8: Manual Attendance

EMAIL NOTIFICATION

- This is the successful message for the attendance captured received via email.

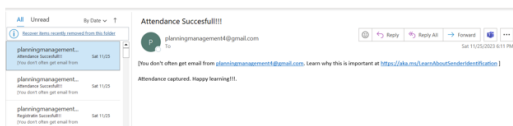


Figure 4.9: Email Notification

5 *Internal/external Interface Impacts and Specification*

Internal Interface Changes:

- **Face Recognition Module:** Interfaces with the OpenCV library for face detection and recognition. Any updates or changes in OpenCV versions should be considered.
- **Graphical User Interface (GUI):** Utilizes Tkinter for the user interface. Any modifications to GUI elements or layouts may impact the user experience.
- **Database Management:** Interfaces with MySQL for data storage. Changes in the database schema or structure should be carefully managed to maintain data integrity.

External Interface Changes:

- **Operating System Compatibility:** The application is designed for Windows 7. Any updates to the supported operating systems should be assessed for compatibility.
- **IDE Compatibility:** Designed to work with Visual Studio Code and PyCharm. Updates to these IDEs may impact the development environment.
- **Camera Requirement:** The application relies on a 720p camera. Changes in camera specifications or compatibility may affect face recognition accuracy.

Data Structure Impacts:

- **Database Schema:** Any alterations to the database schema must be managed to ensure compatibility with existing data and retrieval processes.
- **Image Data:** Image data for face recognition should maintain a standardized format to ensure consistency and accuracy.

Persistent Data:

- **Student and Attendance Records:** Persistent data includes student registration details and attendance records stored in the database.
- **Configuration Settings:** Settings related to the application, such as email preferences and notification settings, are stored persistently.

Data Migration Plan:

Develop a plan for migrating existing data to accommodate any changes in data structures or database systems. Implement a seamless transition to avoid disruptions in data access and retrieval.

6 *Design Units Impacts*

The Smart Attendance GUI Application consists of multiple design units, each contributing to specific functional areas of the project. Below are the impacted design units and their respective subsections:

Face Recognition Module:

- Impacts: The core functionality of face detection and recognition.
- Considerations: Any changes in the OpenCV version or algorithms used for face recognition should be carefully integrated into this module.

Graphical User Interface (GUI):

- Impacts: User interaction and experience.
- Considerations: Updates to Tkinter or changes in GUI layouts may impact how users interact with the application.

Database Management:

- Impacts: Data storage and retrieval processes.
- Considerations: Changes in the database schema or migration to a different database system must be managed to ensure data integrity.

Data Processing Unit:

- Impacts: Processing and manipulation of attendance data.
- Considerations: Updates to Pandas or changes in data processing logic should be incorporated to maintain efficiency.

Image Processing Unit:

- Impacts: Image capture and preprocessing for face recognition.
- Considerations: Any changes in the Pillow library or image processing algorithms should be integrated into this unit.

Email Module:

- Impacts: Communication through email notifications.
- Considerations: Updates to the smtplib library or changes in email configurations may affect the delivery of notifications.

System Integration Unit:

- Impacts: Ensures the seamless integration of all design units.
- Considerations: Compatibility checks between different units, especially during updates or changes in external dependencies.

User Documentation Unit:

- Impacts: Documentation of user guides and manuals.
- Considerations: Updates to reflect changes in the user interface, features, and system requirements.

Testing and Quality Assurance Unit:

- Impacts: Ensures the reliability and effectiveness of the entire system.
- Considerations: Adapt testing strategies to validate new functionalities, interfaces, and any changes introduced.

Continuous Improvement Unit:

- Impacts: Implements lessons learned from each release for future improvements.
- Considerations: Regularly review feedback and performance data to inform continuous enhancements.

7 *Open Issues*

Enhanced Security Features:

- Description: Strengthen the security features of the application, focusing on encryption protocols for data transmission and storage. Consider the implementation of multi-factor authentication to enhance access security.
- Resolution Plan: Conduct a security audit to identify vulnerabilities and implement industry-standard encryption methods. Evaluate the feasibility of integrating multi-factor authentication, considering user experience and system requirements.

Real-Time Collaboration Tools:

- Description: Integrate real-time collaboration tools within the application to facilitate communication between students and instructors. This could include features like chat, discussion forums, or collaborative document sharing.
- Resolution Plan: Research and identify suitable collaboration tools or frameworks. Integrate selected tools into the application, ensuring they align with user needs and contribute to an interactive learning environment.

Enhanced Student Engagement Features:

- Description: Develop features to enhance student engagement, such as gamification elements, rewards systems, or interactive modules. Aim to create a more engaging and participatory learning environment.
- Resolution Plan: Collaborate with educators and students to identify engagement preferences. Design and implement features that align with the learning objectives, incorporating feedback from pilot testing.

8 *Acknowledgments*

I extend my sincere appreciation to Yun-Yau Shih, my primary academic mentor, for demonstrating unwavering trust in my abilities and steadfastly believing in my capacity to complete this project within the stipulated timeframe. The successful culmination of this endeavor owes much to the consistent encouragement, invaluable guidance, and constructive feedback generously provided. I am also deeply grateful to my advisor, who not only served as a vital committee member but also acted as a dedicated academic guide. Their support extended to pivotal decision-making moments, underscoring their belief in my competencies throughout this academic journey.

9 *References*

Python Documentation:

Author: Python Software Foundation

Title: Python 3.9.7 Documentation

Version: 3.9.7

Issue Date: October 15, 2023

<https://docs.python.org/3/>

Tkinter Documentation:

Author: Python Software Foundation

Title: Tkinter 8.6.12 Documentation

Version: 8.6.12

Issue Date: November 5, 2023

<https://docs.python.org/3/library/tk.html>

Pandas Documentation:

Author: pandas Development Team

Title: pandas 1.3.3 Documentation

Version: 1.3.3

Issue Date: September 22, 2023

<https://pandas.pydata.org/pandas-docs/stable/>

Pillow Documentation:

Author: Python Imaging Library Handbook

Title: Pillow (PIL Fork) 8.3.2 Documentation

Version: 8.3.2

Issue Date: August 10, 2023

<https://pillow.readthedocs.io/en/stable/>

OpenCV Documentation:

Author: OpenCV Documentation

Title: OpenCV 4.x Documentation

Version: 4.x

Issue Date: September 30, 2023

<https://docs.opencv.org/4.x/>

smtplib module Documentation:

Author: Python Software Foundation

Title: smtplib — SMTP protocol client

Version: Python 3.9.7

Issue Date: October 5, 2023

<https://docs.python.org/3/library/smtplib.html>

10 Appendices

Appendix A: Student Registration Guide

This guide provides step-by-step instructions on how to register students using the Smart Attendance GUI Application. It includes detailed explanations of the information required, the registration process, and how to verify successful registration.

Appendix B: Manual Attendance Entry

This section outlines the process of manually entering attendance for situations where face recognition may not be feasible. It covers the steps involved in selecting a class, marking attendance for individual students, and saving the data.

Appendix C: Automated Attendance with Face Recognition

This appendix offers insights into the automated attendance feature, leveraging face recognition technology. It explains the prerequisites for successful face recognition, how to initiate the process, and troubleshoot common issues.

Appendix D: Checking Registered Student Data (Professor)

This section guides instructors through accessing and managing student data. It covers how to view, edit, and update student information, ensuring accurate records are maintained.

Appendix E: Alert System Setup

Here, you'll find instructions on setting up and customizing the alert system to notify students about their attendance status via email or text messages. It includes details on configuring notification preferences and troubleshooting potential problems.

Appendix F: Technologies and Tools Documentation

This appendix offers a concise overview of the key technologies and tools essential to the Smart Attendance application. The following components form the core foundation of the application:

- **Python:**
Primary programming language.
- **OpenCV:**
Utilized for computer vision and face recognition.
- **Tkinter:**
GUI toolkit for creating the application interface.
- **Database Management (MySQL):**
Database system for efficient data storage.
- **Pandas:**
Library for data manipulation and analysis.

- **Pillow:**

Library for image processing.

Appendix G: Technical Requirements

This section outlines the minimum technical specifications necessary for the successful installation and optimal performance of the Smart Attendance application. Please ensure that your system meets these requirements before proceeding with the installation.

- **Software Requirements:**

Operating System: Windows 7 or later.

Integrated Development Environments (IDEs): Visual Studio Code or PyCharm.

- **Hardware Requirements:**

Processor: Intel Core i3 or equivalent.

Camera: Minimum 720p resolution (1280 x 720).

RAM: Minimum 8GB.

Hard Disk: Minimum 100 GB of available storage.

It is imperative to adhere to these specifications to guarantee the application's stability, functionality, and responsiveness. Failure to meet these requirements may result in degraded performance or the inability to run the Smart Attendance application effectively.

Before proceeding with the installation, we recommend verifying your system's specifications against the outlined requirements in this appendix. If you encounter any challenges or have questions regarding the technical requirements, please refer to the main documentation or seek assistance from our technical support team.