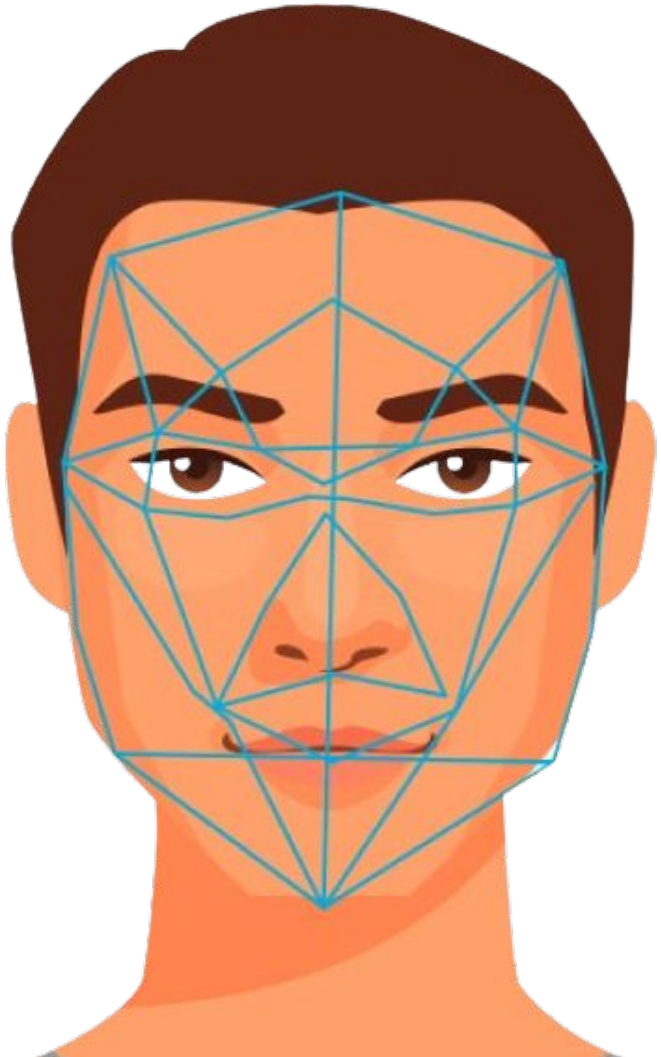
Presence

Auto-Attendance System with report generation



Presented by Team SARS

Rupesh Budhathoki, 191721
Apsara Bishwokarma, 191704
Sampada Kharel, 191723
Sudarshan Khsettri, 191729

TOC

Overview

"Presence" is an automated attendance system designed for college classrooms. It uses facial recognition technology to track student entry and exit, eliminating the need for manual attendance taking. It provides valuable data insights to help educators analyze attendance patterns and make informed decisions for better student engagement and success.

Motivation

1. Time-consuming and inefficient manual attendance tracking methods in college classrooms .

2. Errors and inaccuracies in manual attendance processes can lead to unreliable records.

3. Non-effective monitoring and decision-making.

Objectives

1. To automate attendance system for college classrooms
2. To provide insights into attendance patterns and behavior
3. To support data-driven decision-making for improved student success

Literature Review

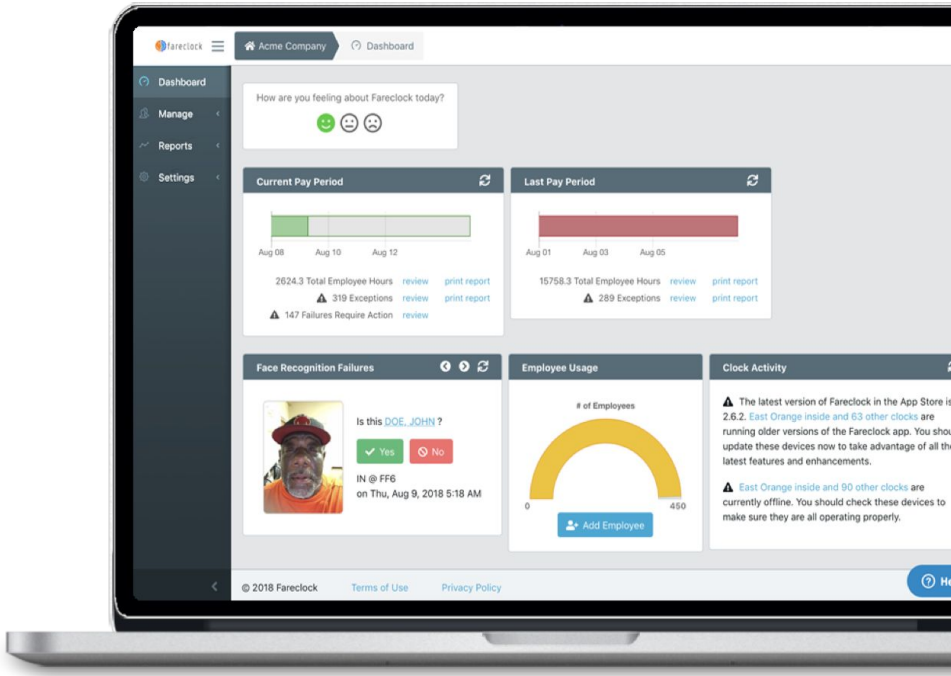
Fareclock

01

Fareclock[1] is a web-based time clock software designed to streamline employee time tracking, attendance management, and payroll processing.

Features

- track work hours
- face recognition attendance
- generate accurate payroll reports.



Literature Review

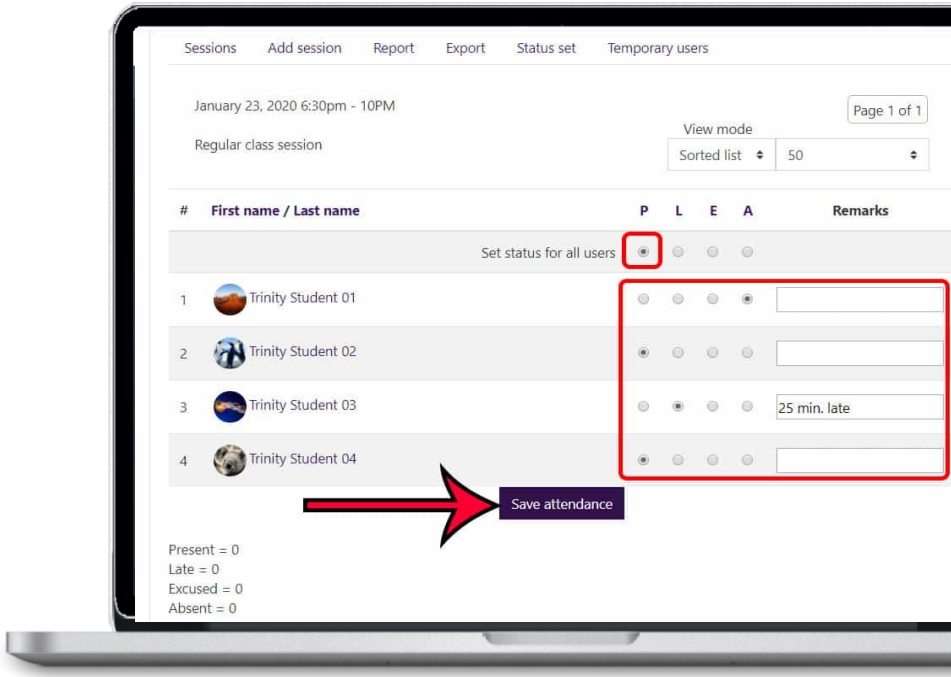
LMS (moodle)

02

Moodle offers attendance activity[2] to track attendance of students digitally.

Features

- Manual attendance marking by instructors
- Customizable attendance statuses (Present, Absent, Late, Excused)
- Download attendance reports in Excel or text format
- Option for students to record their own attendance



Literature Review

Then, Why Us?

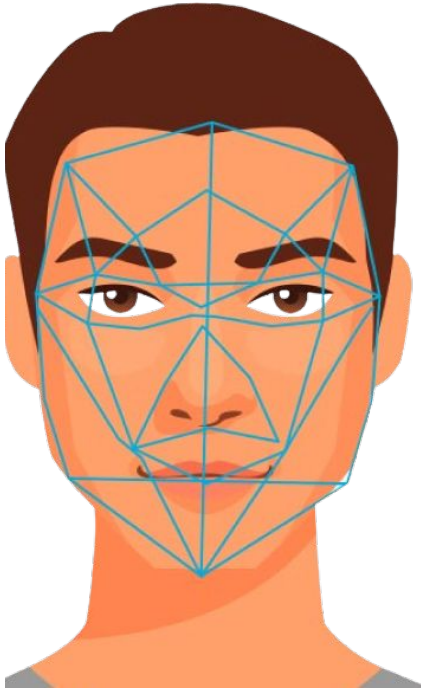
01 | Automated attendance tracking using facial recognition technology

02 | Captures both arrival and departure times

03 | Provides insights for optimizing class scheduling

04 | Real-time attendance updates

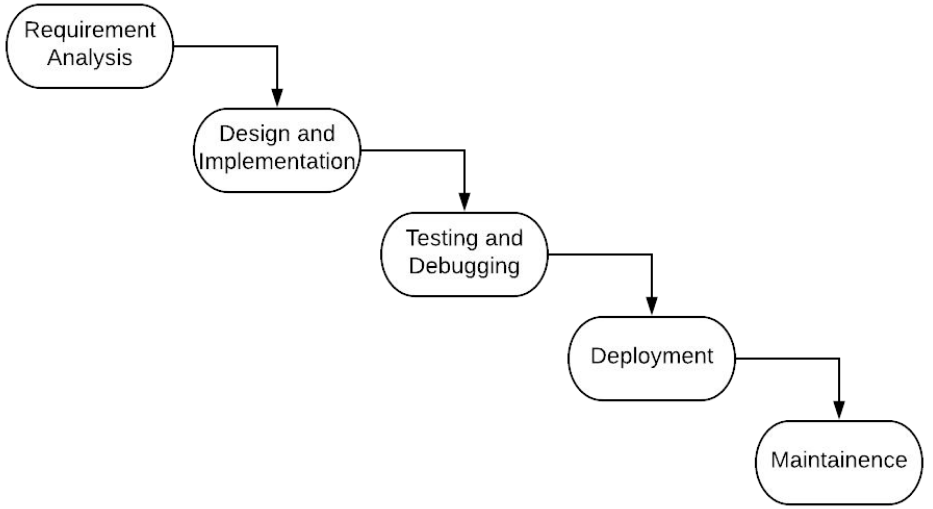
05 | View reports



Methodology

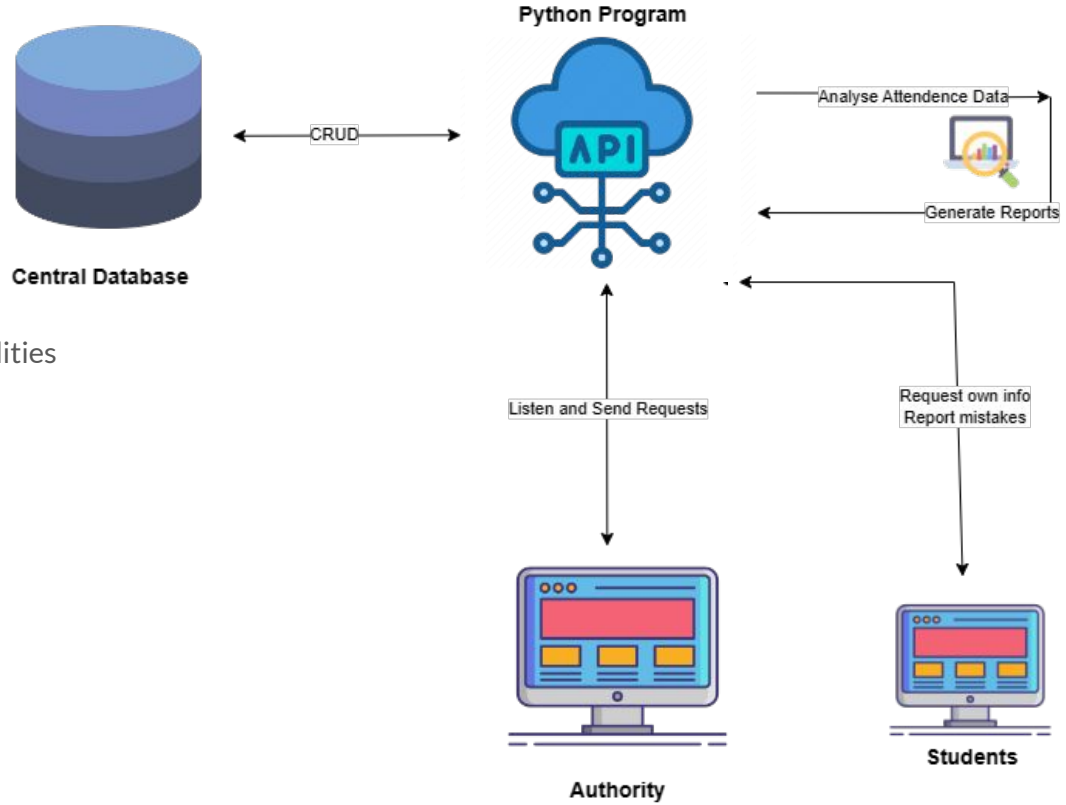
Waterfall model

Our proposed SDLC is WATERFALL MODEL[3] because of limited time and clear goals.



Client-Server Architecture

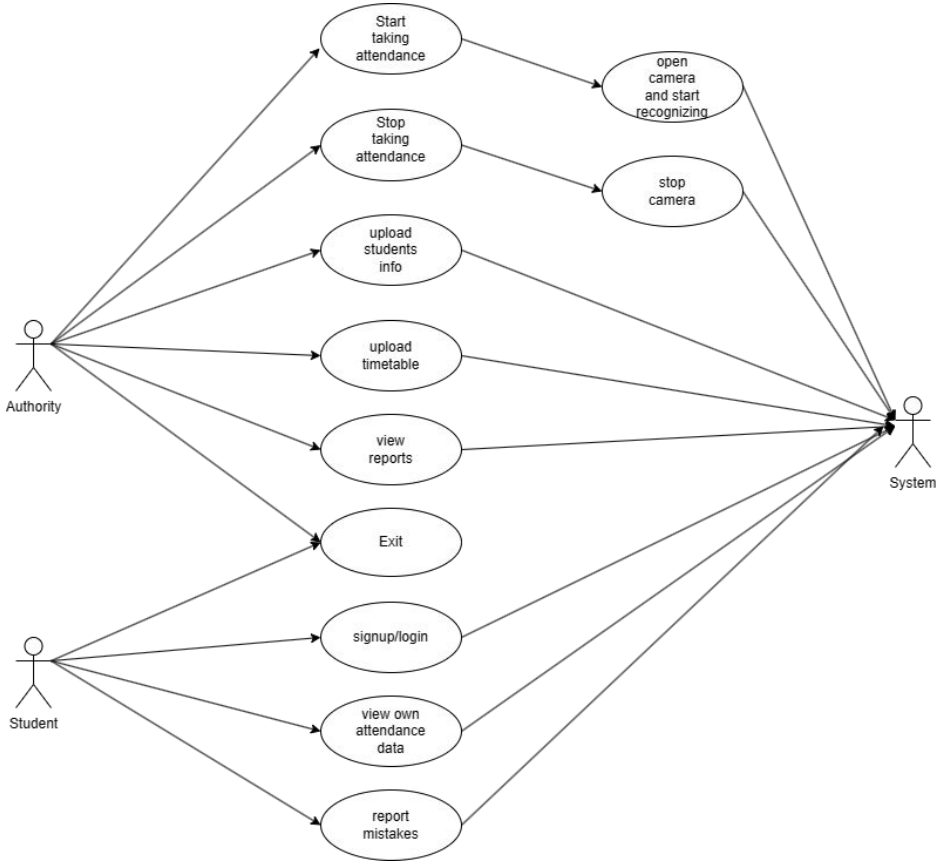
We will divide the functionality and responsibilities of a software application into two main components



Methodology

Use-Case Diagram

How the Authority and Student interacts with the system



Technology

Next.js

Frontend facing the authority and students

Python

Our core which recognizes faces and mark their attendance while also generating reports.

PostgreSQL

Industry favored SQL database to store students and attendance data

WebSocket

Realtime Frontend and backend communication

Typescript, TailwindCSS

Frontend tooling responsible for robust code and responsive design.

OpenCV

Face recognition

Methodology

OpenCV and Deep Learning

OpenCV face recognition pipeline. The key step is a CNN feature extractor that generates 128-d facial embeddings.

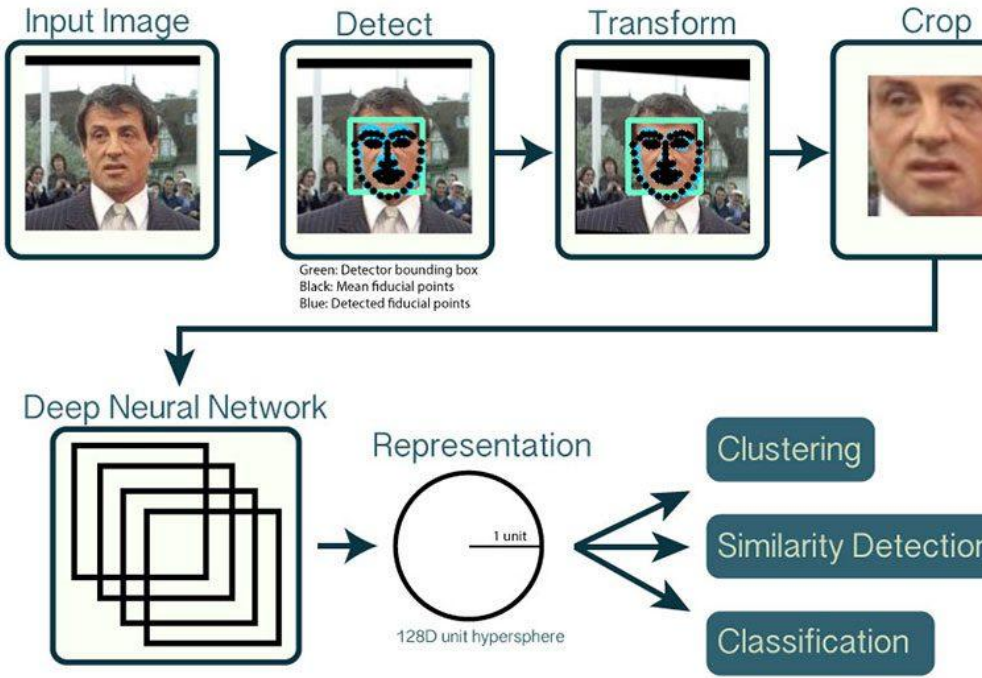
01 | Input an image or video frame

02 | Apply face detection to detect the location of a face

03 | Extract unique facial data (Crop)

04 | Recognize using trained model

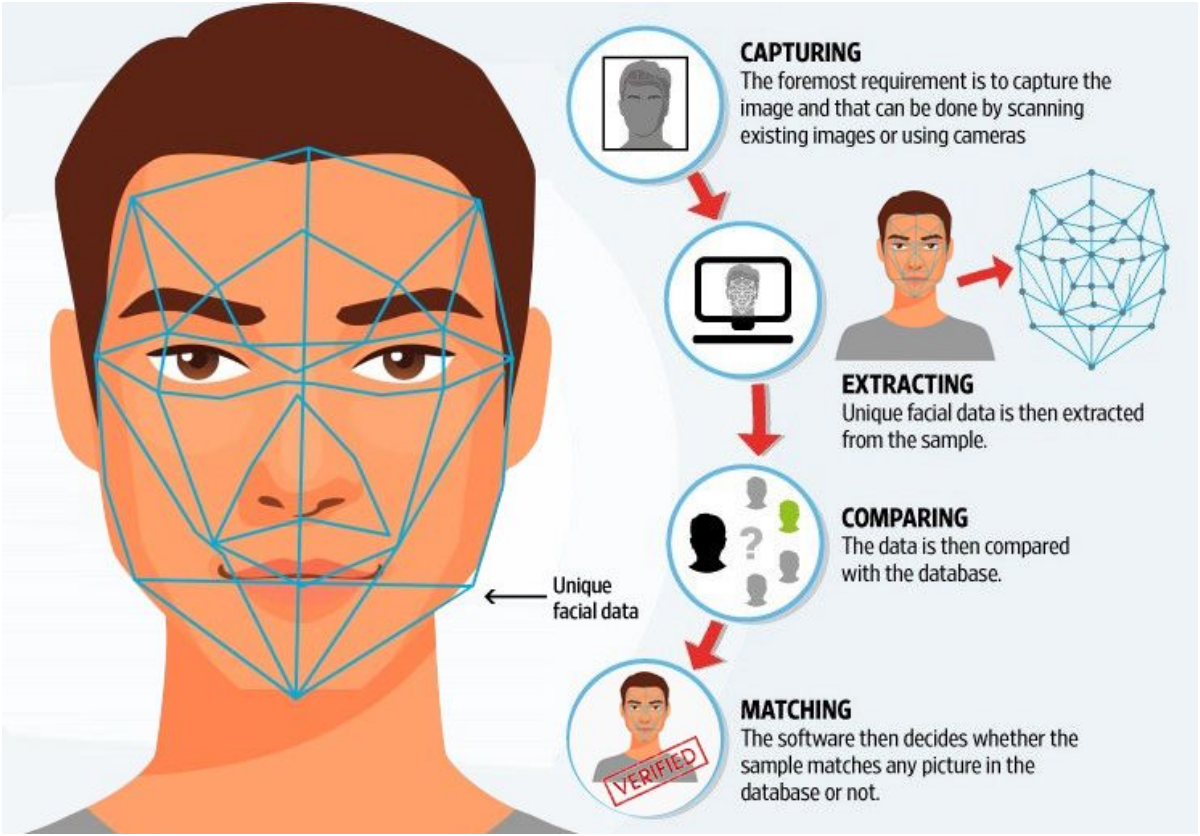
05 | The model is pre-trained with ~3million images



Methodology

Face Recognition Process

Simplified representation of face recognition process



Datasets and Working

- Load and preprocess the dataset using pre-trained model
- Train the facial recognition model
- Detect and align faces in input images
- Compute the Euclidean distance between embeddings to determine the identity of a face

Deliverables

These deliverables encompass various components and functionalities of the automated attendance system, ensuring a comprehensive solution for tracking and analyzing student attendance.

Realtime frontend interface

Client

Deployed Frontend interface where the authority can control the system and students can view their attendance information and report mistakes

Core mechanism

Server

Deployed Python program responsible for face-recognition, api communications and report generation. Including the database design and implementation

Time schedule

Requirement Analysis

Gathering and documenting project objectives and requirements

Testing and Debugging

Identifying and resolving any issues, errors, or defects found during testing

Maintenance

Monitoring the system's performance, resolving user-reported issues, and applying patches

Design and Implementation

Creating a high-level architectural design of the system and Developing the software according to the design specifications

Deployment

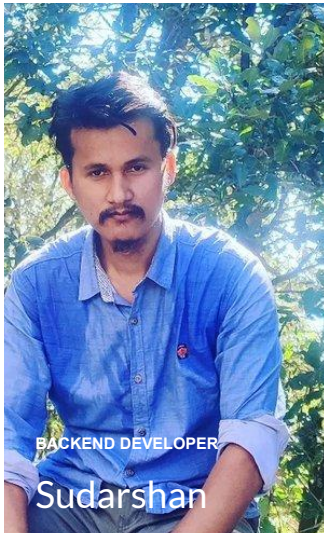
Releasing the software to the production environment or end-users.

Work division

1	Rupesh Budhathoki	• Project Management • Overall Coordination • Coding Support
2	Apsara Bishwokarma	• Frontend Development • Responsive Design
3	Sampada Kharel	• AI development • Reports generation
4	Sudarshan Kshettri	• Backend Development • Database Design and Implementation

Team

Our team consists of four dedicated individuals with diverse skill sets and roles



References

[1] Fareclock, https://www.fareclock.com/.
[2] Moodle, Attendance, https://moodle.org/plugins/mod_attendance.
[3] B. Lutkevich, “Definition: Waterfall model,” TechTarget, 2022. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model#:~:text=The%20waterfall%20model%20is%20a,the%20edge%20of%20a%20cliff.
[4] Rosebrock, A. (2018, June 18). Face recognition with OpenCV, Python, and deep learning. PyImageSearch. https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/

Thank you for your time!!!