भारतीय सूचना प्रौद्योगिकी संस्थान धारवाड़
INDIAN INSTITUTE OF
INFORMATION TECHNOLOGY DHARWAD
[Institute of National Importance by Act of Parliament]

Indian Institute of Information Technology, Dharwad
Software Requirement and Technical Feasibility Document

CS310: Software Engineering Project

Prepared by: Team **Hukum**
Group Members:

Amritanshu Aditya	*23bcs013*
Barghav Abhilash B R	*23bcs028*
Dhruv A. Koli	*23bcs044*
K V Modak	*23bcs067*

Faculty Advisor:
Dr. Vivekraj V K*

note†

January 19, 2025

*Dept. of Computer Science, IIIT Dharwad
† ***This is not the official documentation of our product.***

Contents

1 Introduction **3**
1.1 The Problem Statement . . . 3
1.1.1 The Traditional Attendance Procedure . . . 3
1.1.2 Hardware Dependency . . . 3
1.1.3 Proxy Attendance . . . 3
1.1.4 Data Management Challenges . . . 3
1.1.5 Scalability Issues . . . 3
1.1.6 User Experience . . . 3
1.2 Our Approach . . . 3
1.3 What the customer wants? . . . 3
1.3.1 Cost-effective Attendance System . . . 3
1.3.2 Reduced Time for Marking Attendance . . . 4
1.3.3 Minimized Proxies . . . 4
1.3.4 Ease of Use and Better Maintainability . . . 4
1.4 Objectives . . . 4
1.4.1 Cost Efficiency . . . 4
1.4.2 Minimized Proxies . . . 4
1.4.3 Automation . . . 4
1.4.4 Ease of Use and Maintenance . . . 5
1.4.5 Scalability . . . 5

2 Software Requirements **5**
2.1 Basic Requirements . . . 5
2.1.1 Scope . . . 5
2.1.2 Budget . . . 5
2.1.3 Time . . . 6
2.2 HLD . . . 6

3 Technical Feasibility **7**
3.1 Feasibility Analysis . . . 7
3.1.1 Data Collection and Training . . . 7
3.1.2 Platform for customer . . . 7
3.1.3 Processing of Images . . . 7
3.1.4 Feedback . . . 7
3.1.5 Monitoring . . . 7
3.2 Cost & Benefit Analysis . . . 8
3.2.1 Cost Analysis . . . 8
3.2.2 Benefit Analysis . . . 8
3.3 Technology Stack . . . 8

1 Introduction

1.1 The Problem Statement

1.1.1 The Traditional Attendance Procedure

- Current methods are time-consuming and prone to human error.
- Manual entry increases the administrative workload for faculty.

1.1.2 Hardware Dependency

- Existing automated systems often require expensive hardware, such as biometric devices or card readers.
- High hardware costs make these systems less accessible to many institutions.

1.1.3 Proxy Attendance

- Traditional systems, both manual and some automated, allow for proxy attendance, compromising the accuracy of attendance records.

1.1.4 Data Management Challenges

- Manually maintaining attendance records can lead to inconsistency and data loss.
- Retrieval and analysis of attendance data are cumbersome in traditional systems.

1.1.5 Scalability Issues

- Many current systems are not scalable to accommodate larger class sizes or multiple sessions without additional resources.

1.1.6 User Experience

- Complexity in using some existing systems can deter users, reducing system adoption and efficiency.
- Maintenance of these systems can also be challenging and resource-intensive.

1.2 Our Approach

We solve this issue by leveraging image recognition for identifying the students for attendance. We also involved automation of attendance marking system to reduce manual work for all parties involved in this.

1.3 What the customer wants?

1.3.1 Cost-effective Attendance System

- A system designed to minimize expenses while efficiently marking attendance.

1.3.2 Reduced Time for Marking Attendance

- The purpose of this project is to streamline and expedite the attendance process, saving valuable time.

1.3.3 Minimized Proxies

- Implement measures to significantly reduce the incidence of proxy attendance.

1.3.4 Ease of Use and Better Maintainability

- Ensure that the system is user-friendly and easy to maintain, promoting long-term usability.

1.4 Objectives

Our approach to creating an efficient attendance marking system is clear and practical. The main objectives are as follows.

1.4.1 Cost Efficiency

- Most automated attendance systems require significant hardware investments.
- Our model provides a cost-efficient solution that is entirely software-based.
- Using image recognition and automation, we eliminate hardware costs by leveraging open-source libraries to achieve the desired results.

1.4.2 Minimized Proxies

- Both manual and automated systems often have a margin of error, allowing proxies.
- Our solution addresses this by requiring students to physically be present in the classroom.
- Faculty members take images as evidence, which are used for image recognition.
- This ensures that proxies are nearly impossible, as the process depends entirely on the faculty's timing to capture and upload images.

1.4.3 Automation

- Image recognition is the foundation of our system.
- The system automatically generates attendance records for sessions conducted by faculty.
- All attendance data is logged into a web portal accessible to administrators, students, and faculty.

1.4.4 Ease of Use and Maintenance

- Our system is designed to be user-friendly and maintainable.
- The web portal provides easy access to all required data for clients, students, and faculty, ensuring convenience and compliance.

1.4.5 Scalability

- The system is scalable to accommodate the entire class of 70 students.
- Tests have shown that our image recognition model can accurately identify faces with over 66% accuracy for a single column of students in the classroom. Check references.

2 Software Requirements

2.1 Basic Requirements

2.1.1 Scope

- The software should be such that it allows the instructor of the course to access and manage the attendance of all students in their class.
- The software should allow students to see which classes they have been marked as present and which classes they have missed for all subjects in their particular semester.
- After marking attendance for a class, the software should have a feedback system for the students that would let them know that they have been marked present or absent in the attendance database.
- The software should have great precision in recognizing students and should have minimal or no mismatches.
- In the rare case of mis-detection, the software should allow the course instructor to edit and update attendance for the missed person, provided that the aggrieved student approaches them within a reasonable time.

2.1.2 Budget

- The project requires minimal investment. The training database will consist of photos of the students, which the software can map to photos taken in class for attendance marking.
- Photos that will be uploaded daily will be stored on an on-line server that handles File Transfer Protocol (FTP). The database will request an API from this server to access those images and mark attendance for the students.

2.1.3 Time

- Data Training and Data Collection: 3 to 4 weeks
- Development: 8 to 9 weeks
- Testing (Phase-wise testing and final testing): 2 to 3 weeks

2.2 HLD

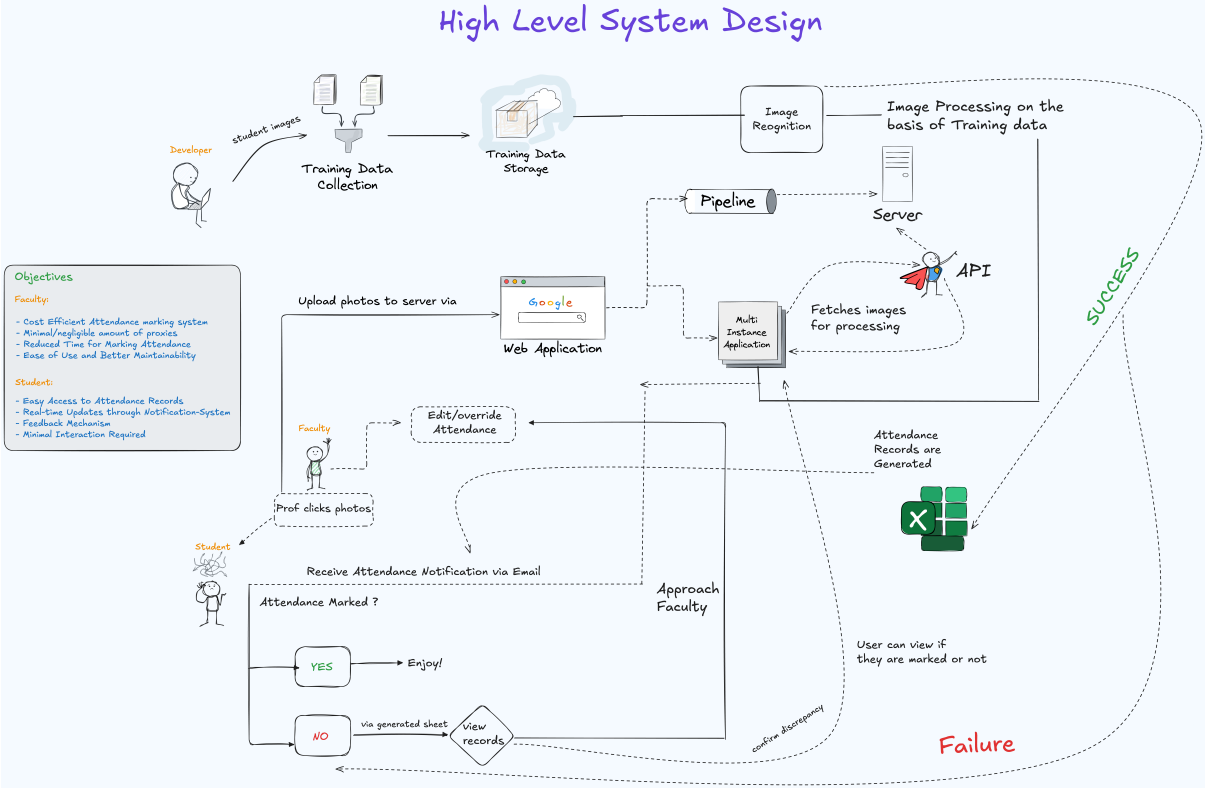

Figure 1: HLD

3 Technical Feasibility

3.1 Feasibility Analysis

3.1.1 Data Collection and Training

- Collection of adequate face data for all students.
- Training the face recognition model to recognize on the basis of the training data.
- Testing the system using varying light and occlusions.
- Can be done using pre-trained models such as **FaceNet** and **DeepFace** as well.

3.1.2 Platform for customer

- A web platform where user provides photos as input.
- Photos can be uploaded or taken live. However, the one with easier functionality for the user and the one that is less error-prone may be implemented.
- The photos are taken column-wise, ensuring that everyone's face is properly visible.
- One class will generate an input of 4 photos, 1 of each column.

3.1.3 Processing of Images

- Includes
  - Face Detection
  - Face Recognition
  - Mapping of records
- The images are processed via the backend which is trained upon the training image data.

3.1.4 Feedback

- Students receive an email notification regarding their attendance status for the class.
- In any rare case of face mismatch, manual overriding of records is possible.

3.1.5 Monitoring

- The attendance database are constantly monitored by our developers who oath to work honestly in the surveillance of the records.
- Helps reduce any potential discrepancies and aids in the further scalability of the product.

3.2 Cost & Benefit Analysis

3.2.1 Cost Analysis

- Storage requirements for new photos.
- Backup photos for a period of two weeks in case of discrepancies.
- A working camera phone.

3.2.2 Benefit Analysis

- Time and Cost efficient.
- Almost negligible proxies.
- More hassle free for students and faculty.
- Students get the access to their attendance records 24/7 anywhere from the world.

3.3 Technology Stack

Component	**Primary Options**	**Alternative Options**
Frontend	React.js with TS / JS	Vue.js with TS & Vuetify
UI Components	Shadcn/ui, Tailwind CSS	UnoCSS
Backend API	FastAPI, Python 3.9+	Django REST, Flask
Face Recognition	OpenCV, face_recognition	DeepFace, MediaPipe(Google)
Database	PostgreSQL, SQLite (for local testing)	Redis (Caching/Session), MongoDB
Storage	Google Drive API, Local Storage	Amazon S3 (free tier)
Email Service	SMTP (Python smtplib)	SendGrid (free tier)
Testing	pytest, integration tests	———
Dev Tooling	DevLogs, Documentation, GIT VCS	Code Linting
Deployment	Heroku (free tier), Vercel	Docker, Netlify
Monitoring	Basic Python logging	Sentry (for error tracking)

References

[1] Bradski, G., *The OpenCV Library*, Dr. Dobb's Journal of Software Tools, 2000. https://opencv.org/.

[2] Adam Geitgey, *Face Recognition with Python*, 2018, https://github.com/ageitgey/face_recognition.

[3] SmartPencs, *Attendance Management System Using Face Recognition* Live Face Detection without face recognition https://rb.gy/kelprs.

[4] Excalidraw, *High Level Design for Project* Whiteboard tool that lets you easily sketch diagrams https://excalidraw.com & https://libraries.excalidraw.com/.

[5] File Transfer Protocol, Useful in handling file (image) receive and send operations between server and face recognition code. https://en.wikipedia.org/wiki/File_Transfer_Protocol

[6] Google Drive API, *Google Drive API for file upload and downloads*

[7] OpenCV claims a successful match of faces with a score more than 66%.