B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OOMD Mini Project Report

Automated Attendance System in Rural Areas

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Navanidhi D J (1BM23CS204)
Neha Sajjanar (1BM23CS209)
Nidhi D (1BM23CS211)
Nishmitha K G (1BM23CS216)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

2025-26

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Navanidhi D J(1BM23CS204), Neha Sajjanar(1BM23CS209), Nidhi D B(1BM23CS211), Nishmitha K G(1BM23CS216) students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled " **Automated Attendance System in Rural Areas**" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester August 2025- December 2025. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

Navanidhi D J (1BM21CS204)
Neha Sajjanar (1BM23CS209)
Nidhi D B (1BM23CS211)
Nishmitha K G(1BM23CS216)

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD Mini Project titled "**Automated Attendance System in Rural Areas"** has been carried out by **Navanidhi D J (1BM23CS204), Neha Sajjanar (1BM23CS209), Nidhi D B (1BM23CS211), Nishmitha K G (1BM23CS216)** during the academic year 2025-2026.

Signature of the Faculty in Charge

Sunayana S

Assistant Professor

Table of Contents

Chapter 1: Problem Statement

Attendance management in rural schools has long been a persistent challenge that affects both teaching quality and administrative efficiency. In most rural educational institutions, teachers still rely on traditional pen-and-paper methods to track student presence, which consumes a significant portion of their limited class time. This manual approach not only takes away from actual teaching but also introduces numerous opportunities for mistakes—from illegible handwriting to calculation errors when compiling monthly reports. Moreover, physical registers are vulnerable to damage from harsh weather conditions, accidental loss, or even deliberate tampering, putting months of important attendance data at risk.

What makes this problem particularly acute in rural settings is the lack of basic infrastructure that urban schools often take for granted. Internet connectivity in these areas tends to be unreliable at best and completely unavailable at worst, ruling out most modern cloud-based solutions that depend on constant online access. Schools operating on tight budgets cannot afford expensive commercial attendance systems, and even if they could, the teachers and staff often lack the technical background needed to operate sophisticated software. Add to these the frequent power outages common in rural regions, and you have an environment where technology adoption becomes extremely challenging. These factors combine to create a situation where rural schools remain stuck with outdated methods despite the availability of better alternatives elsewhere.

The ripple effects of these attendance tracking difficulties extend far beyond simple record-keeping. Without an efficient system in place, schools find it nearly impossible to spot patterns in student absenteeism or identify at-risk students who might be quietly dropping out. Generating attendance reports for district officials or government audits becomes a tedious, error-prone task that takes days of manual compilation. Parents often remain in the dark about their children's actual attendance, discovering problems only when report cards arrive months later. This lack of transparency and timely information prevents everyone involved—teachers, administrators, parents, and education authorities—from taking prompt action to address attendance issues before they escalate into serious academic problems.

What's clearly needed is a practical attendance solution designed with rural realities in mind. Crucially, it must work seamlessly even when internet connectivity drops, storing records locally and then syncing them to a central database once the connection returns. The interface needs to be straightforward enough that teachers can use it confidently after minimal training, without requiring constant technical support. By running on affordable hardware like Raspberry Pi computers paired with basic webcams, the system can fit within rural school budgets. When implemented properly, such a solution would free up teachers to concentrate on what they do best—teaching—while giving school administrators and parents the real-time insights they need to support student success effectively.

Chapter 2: Software Requirement Specification

1. Introduction

1.1 Purpose

This document lays out the technical and functional blueprint for building an attendance system specifically crafted for rural educational settings. It walks developers, evaluators, and project stakeholders through what the system should do, how it should perform, where its boundaries lie, and what design choices matter most. By reading this, everyone involved gets a shared understanding of the project's goals and execution plan.

1.2 Scope

Taking attendance in rural schools currently involves teachers spending precious class time calling out names and marking registers by hand. This project replaces that tedious process with an automated solution that recognizes student faces through a camera and instantly logs their presence. What makes this particularly useful for rural areas is its ability to function without internet—a common reality in these regions. The system saves all attendance records on a local device, and whenever internet connectivity returns, it quietly uploads everything to the cloud. The end result is better accuracy, less burden on teachers, and more dependable student data tracking.

1.3 System Overview

At its core, the system combines a camera, an affordable computing device like a Raspberry Pi or basic computer, and a database stored right on the device. Using OpenCV technology, it spots faces, matches them against registered student profiles, and logs each attendance entry along with the exact time. Teachers get a straightforward dashboard where they can pull up attendance summaries, review individual student records, and create reports covering any time period they need. Meanwhile, administrators can access the synchronized cloud data to keep tabs on attendance patterns across entire schools or districts.

2. Overall Description

2.1 Product Perspective

This attendance system works independently without needing constant connection to external systems. It's built to thrive in challenging environments where resources are limited and internet comes and goes. The system brings together facial recognition capabilities, on-device data storage, cloud backup when possible, and an interface that anyone can navigate without extensive training.

2.2 Users of the System

Teachers: Check daily attendance, maintain student information, and create attendance reports whenever needed.

Administrators: Track attendance trends across different classes or even multiple schools to spot patterns or issues.

Students: Simply show up—the system handles everything else by recognizing their faces without any action required from them.

2.3 Operating Environment

Hardware: Any standard webcam, a Raspberry Pi or regular PC, and some form of local storage device.

Software: Built using Python programming, OpenCV for image processing, SQLite for local data storage, and MySQL or Firebase for cloud backup.

Network: Designed to work perfectly offline; cloud features kick in automatically whenever internet appears.

3. Functional Requirements

3.1 Image Capture

The system constantly monitors the camera feed during attendance time. When a student positions themselves in front of the camera, the system automatically captures their face image without any button pressing or manual intervention required.

3.2 Face Detection and Recognition

Using OpenCV algorithms, the system first detects whether a face is present in the camera view. It then compares the detected face against all previously registered student photos in its database. Each successfully matched student gets their attendance marked using their unique student ID.

3.3 Attendance Recording

Once the system successfully identifies a student, several things happen automatically: it marks them as present for that session, records the exact time they were recognized, and saves this complete entry into the local database. This entire process takes just a couple of seconds.

3.4 Offline Data Storage

All attendance information gets saved in SQLite, a lightweight database that lives right on the device. Even when there's zero internet connectivity, the system continues functioning normally, ensuring that no attendance data is ever lost because of connectivity issues.

3.5 Cloud Synchronization

The system periodically checks for internet connectivity. When it finds a connection, it automatically begins uploading all unsynced attendance records to the cloud server. It's smart enough to avoid creating duplicate entries by checking which records have already been uploaded previously.

3.6 User Authentication

Both teachers and administrators must log in with their authorized credentials before accessing the system. Different user types see different features and have different permissions, ensuring that sensitive data remains protected while still being accessible to those who need it.

3.7 Reporting

The system can generate attendance reports covering daily, weekly, or monthly periods based on what teachers or administrators need. These reports can be exported as CSV files for spreadsheet analysis or as PDF documents for printing and official record-keeping.

3.8 Student Registration

During registration, the system captures several photos of each new student from different angles and lighting conditions. These images get processed to extract unique facial features, which then train the recognition model to accurately identify that student going forward.

4. Interface Requirements

4.1 User Interface Requirements

The dashboard follows a clean, uncluttered design that anyone can understand quickly. Key functions like starting attendance capture, viewing reports, managing student records, and checking sync status are all accessible from the main screen without hunting through nested menus. Since rural schools might be using older monitors or displays, all text and buttons are sized to remain clearly visible even on lower-resolution screens.

4.2 Hardware Interfaces

- Standard webcam or any compatible USB camera
- Raspberry Pi single-board computer or regular PC
- MicroSD card or traditional hard disk for storage
- Optional: Wi-Fi adapter or ethernet cable for internet connectivity

4.3 Software Interfaces

- OpenCV library handles all image processing and face detection tasks
- SQLite manages local data storage without needing a separate database server

- Cloud database (either Firebase or MySQL) provides remote backup and multi-school access
- Python programming language ties everything together and powers the user interface

5. Performance Requirements

Face recognition must complete within 1 to 2 seconds of a student appearing on camera. The system should achieve recognition accuracy above 90% when lighting conditions are reasonable. Local storage capacity should comfortably handle data for 500 to 700 students over an entire academic year. The system needs to run smoothly for 4 to 6 hours daily—covering typical school hours—without slowing down or freezing. When internet connectivity returns, cloud synchronization should finish within 30 to 45 seconds, depending on how much data accumulated offline.

6. Design Constraints

Rural environments often have limited or unreliable power supply, requiring the system to use minimal processing power and energy. Recognition accuracy naturally decreases in poor lighting conditions or when using lower-quality cameras, which must be factored into expectations. Hardware costs must stay low enough that budget-constrained schools can actually afford to implement the solution. The entire system must function perfectly without requiring constant internet access. Additionally, student biometric data raises privacy concerns, so the system must follow data protection guidelines and handle this sensitive information responsibly.

7. Non-Functional Requirements

7.1 Usability

Teachers should be able to start using the system confidently after just a brief training session. All buttons, menus, and labels use straightforward language that explains exactly what each feature does, eliminating guesswork and confusion.

7.2 Reliability

Attendance records must never disappear, regardless of power failures, system crashes, or network problems. If the system needs to restart for any reason, it should automatically resume normal operation without requiring manual intervention or data recovery procedures.

7.3 Security

Student facial data gets encrypted both when stored locally and when transmitted to the cloud. Only users who have logged in with valid credentials can view attendance records. All data transfers between the local system and cloud servers use secure protocols like HTTPS and SSL encryption to prevent interception.

7.4 Portability

The software runs equally well on Windows computers and Linux-based systems, giving schools flexibility in their hardware choices. It works with various camera models and brands without requiring specific expensive equipment.

7.5 Maintainability

The code is organized into separate modules, making it straightforward to update individual components—like swapping in an improved recognition algorithm—without rewriting the entire system. Detailed logs track system activity, helping identify and fix problems quickly.

7.6 Scalability

The cloud infrastructure supports growing from a single classroom to multiple classes and even expanding across several schools within a district. As student enrollment increases over time, the system handles the additional load without performance degradation.

8. Project Planning

8.1 Estimated Timeline

Phase	Duration
Requirement Gathering	1–2 weeks
Design & Dataset Collection	2–3 weeks
System Development	4 weeks
Testing & Debugging	2 weeks
Deployment	1 week
Final Review & Optimization	1 week

8.2 Budget Estimate

Item	Approximate Cost
Camera/Webcam	₹2,500-₹3,000
Raspberry Pi / Desktop Setup	₹4,000–₹5,000
Local Storage	₹1,000
Maintenance & Miscellaneous	₹1,000
Total Estimated Cost	**₹8,500 – ₹10,000**

Chapter 3: Class Modeling

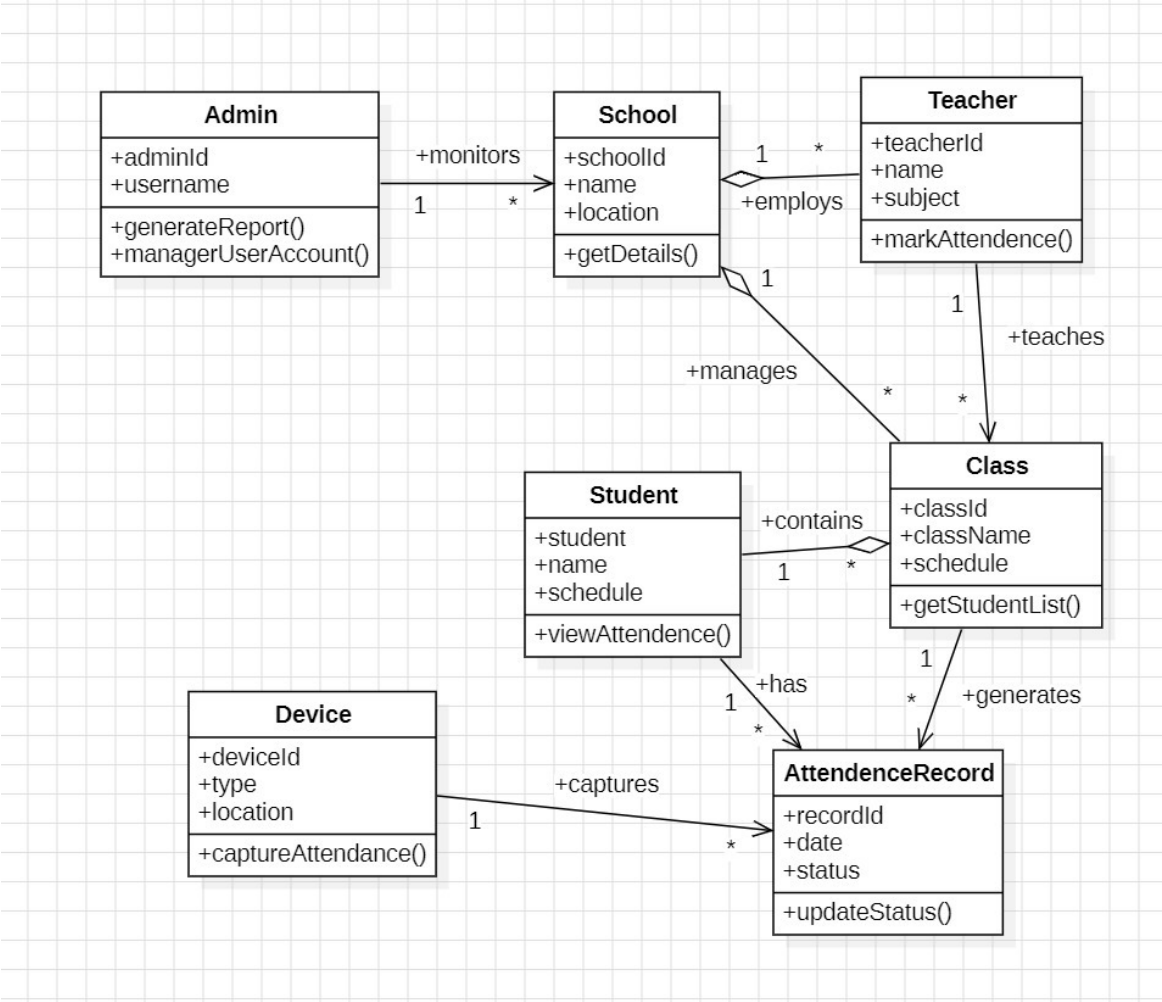


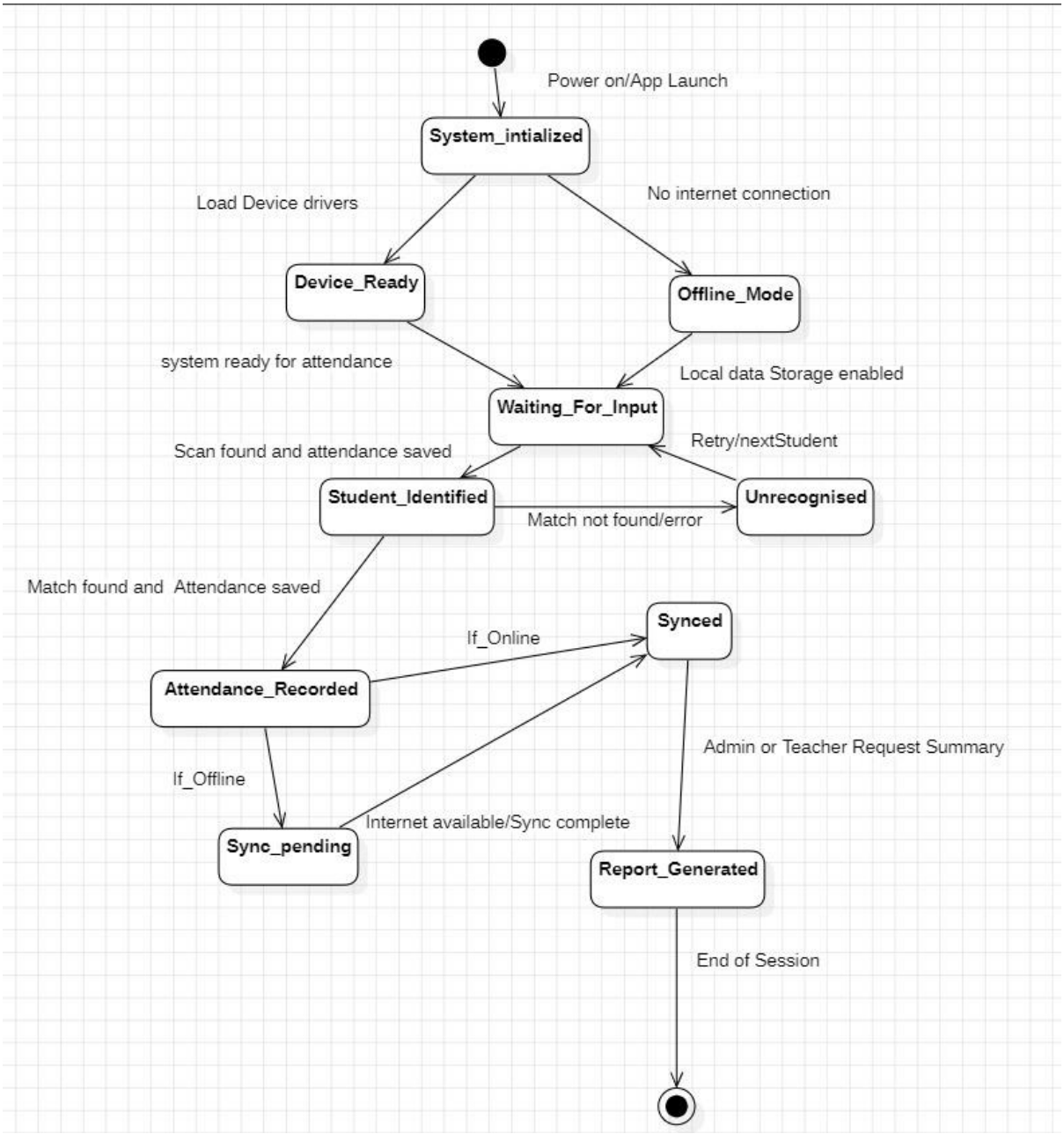
Fig 3.1 Attendance System Class Diagram

The diagram shows how different parts of a school attendance system are connected. Each box represents something important in the system, like a student, teacher, or device, and the arrows show how they interact with one another.

- **Admin**: This role manages the overall system. An admin can create reports and handle user accounts, but they don't directly deal with attendance.
- **School**: Each school has basic details like its name and location. The school is in charge of hiring teachers and overseeing how the system runs.
- **Teacher**: A teacher works for a school and is responsible for marking attendance for the classes they teach. Each teacher has an ID, name, and the subject they teach.

- **Class**: A class includes information such as its ID, name, and schedule. A class contains a group of students and is taught by a teacher. It also generates attendance records based on student participation.
- **Student**: Students belong to a class and have details like name and schedule. Students can check their own attendance information.
- **Device**: Devices (like scanners or cameras) are used inside the school to capture attendance data. They record which students are present by sending information to the system.
- **AttendanceRecord**: These are the actual logs that show whether a student was present or absent on a specific date. Records can be updated, and they are generated by classes and captured through devices.

Chapter 4: State Modeling



4.1 Attendance System State Diagram

This diagram shows how the attendance system works from the moment the app or device is turned on until the attendance session ends.

- When the system starts, it checks for the necessary device drivers. If everything loads correctly, the device becomes **ready** to capture attendance.

- At the same time, the system also checks for an internet connection. If the internet is not available, it switches to **offline mode**, where attendance can still be stored locally.
- Once the system is ready, it waits for input—usually a student scan.
    - If the system recognizes the student, it moves to **Student Identified**.
    - If not, the scan is marked as **Unrecognised**, and the user can try again.
- When a student is successfully identified, their attendance is recorded.
    - If the device is online, the record is synced immediately.
    - If the device is offline, the record is saved as **Sync Pending** until the internet becomes available.
- When syncing is complete, or when an admin or teacher asks for a summary, the system generates a **report**.
- After the report is created or the session ends, the process completes and waits for the next session.

Chapter 5: Interaction Modeling

5.1 Use Case Diagram:

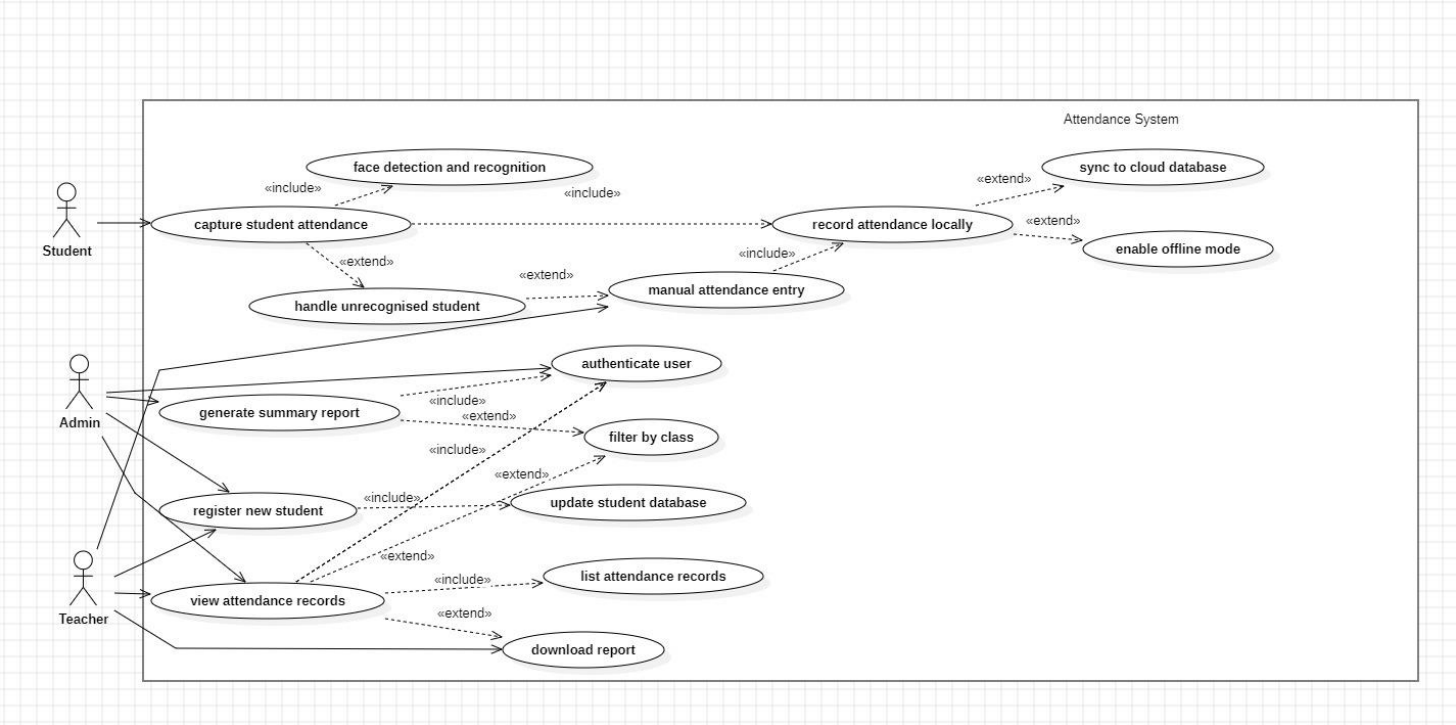


Fig 5.1.1 Attendance System Use Case Diagram

The diagram shows how different users—Students, Teachers, and Admins—interact with an automated attendance system that relies on face recognition.

Student Interactions

Students mainly interact with the system to have their attendance recorded.

- The system uses face detection and recognition to automatically capture attendance.
- If the system fails to recognise a student, it triggers a process to handle unrecognised students, which may involve a manual check.
- All attendance data is first saved locally, and then later synced to a cloud database when internet is available.
- If there's no internet, the system can switch to offline mode so attendance can still be logged.

Teacher Interactions

Teachers can log in and perform several tasks:

- They can view attendance records for their classes.
- They can also download reports if they need a copy.
- Their actions are protected by an authentication step so only authorized teachers can access data.

Admin Interactions

Admins have the most control in the system:

- They can add new students and update the student database as needed.
- They can also list or browse all attendance records, sometimes narrowing the results using a class filter.
- Like teachers, they must also authenticate before using the system.
- Admins can also produce a summary report, which gives an overview of attendance across the system.

5.2 Sequence Diagrams

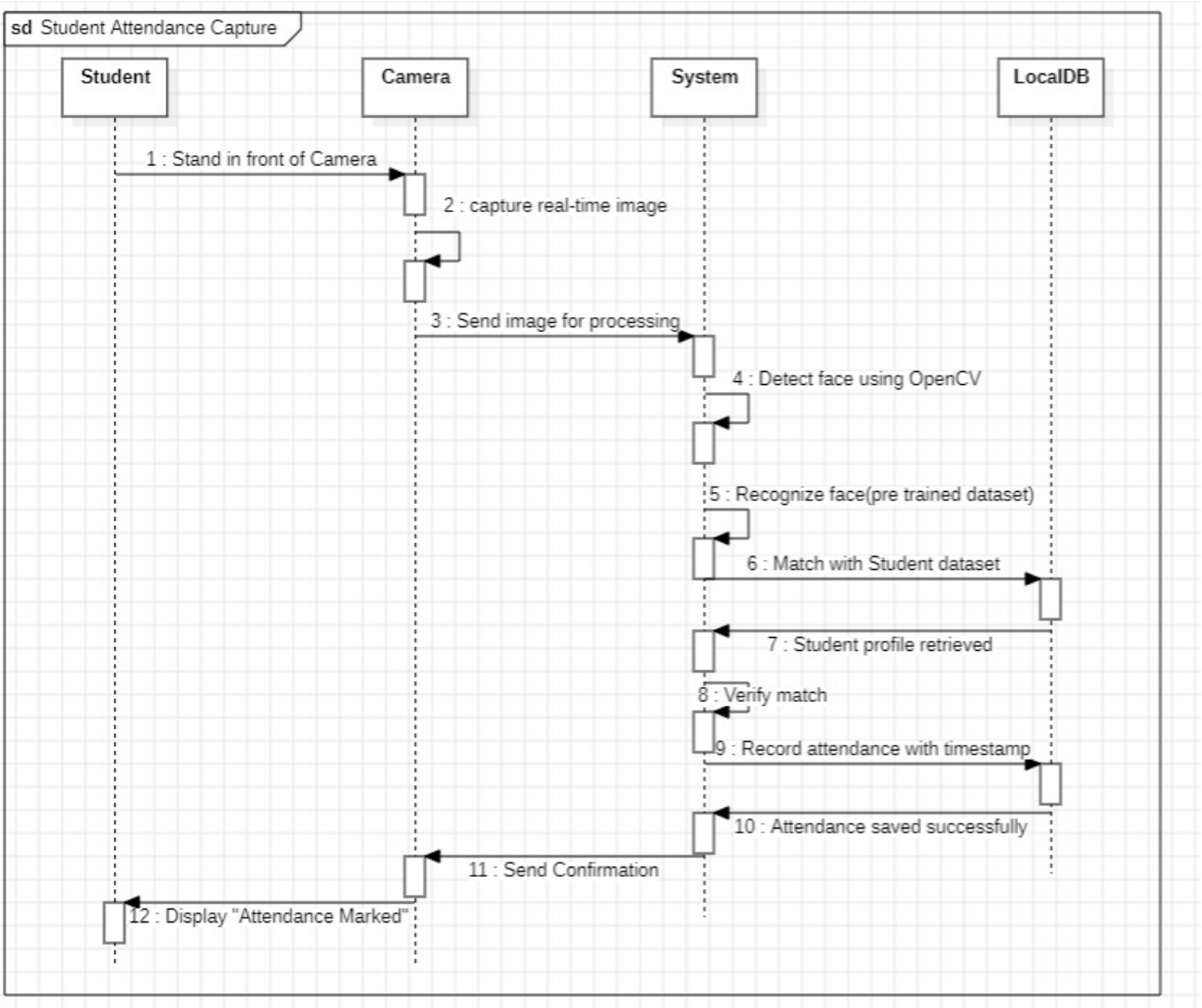


Fig 5.2.1. Student Attendance Capture

This diagram shows the step-by-step process of how a student's attendance is captured using facial recognition.

1. The process starts when the student stands in front of the camera.
2. The camera captures a real-time image of the student.
3. The captured image is then sent to the system for analysis.
4. The system first detects the face in the image using OpenCV.
5. Next, it recognizes the face by comparing it with a pre-trained dataset.
6. The system then matches the detected face with the stored student records in the local database.
7. Once the match is found, the student's profile is retrieved.
8. The system verifies the match to ensure the correct student is identified.

9. The attendance is recorded along with the timestamp**.**
10. The local database confirms that the attendance has been successfully saved**.**
11. The system sends a confirmation message back to the camera.
12. Finally, the student sees a message saying **"**Attendance Marked"**,** completing the process.

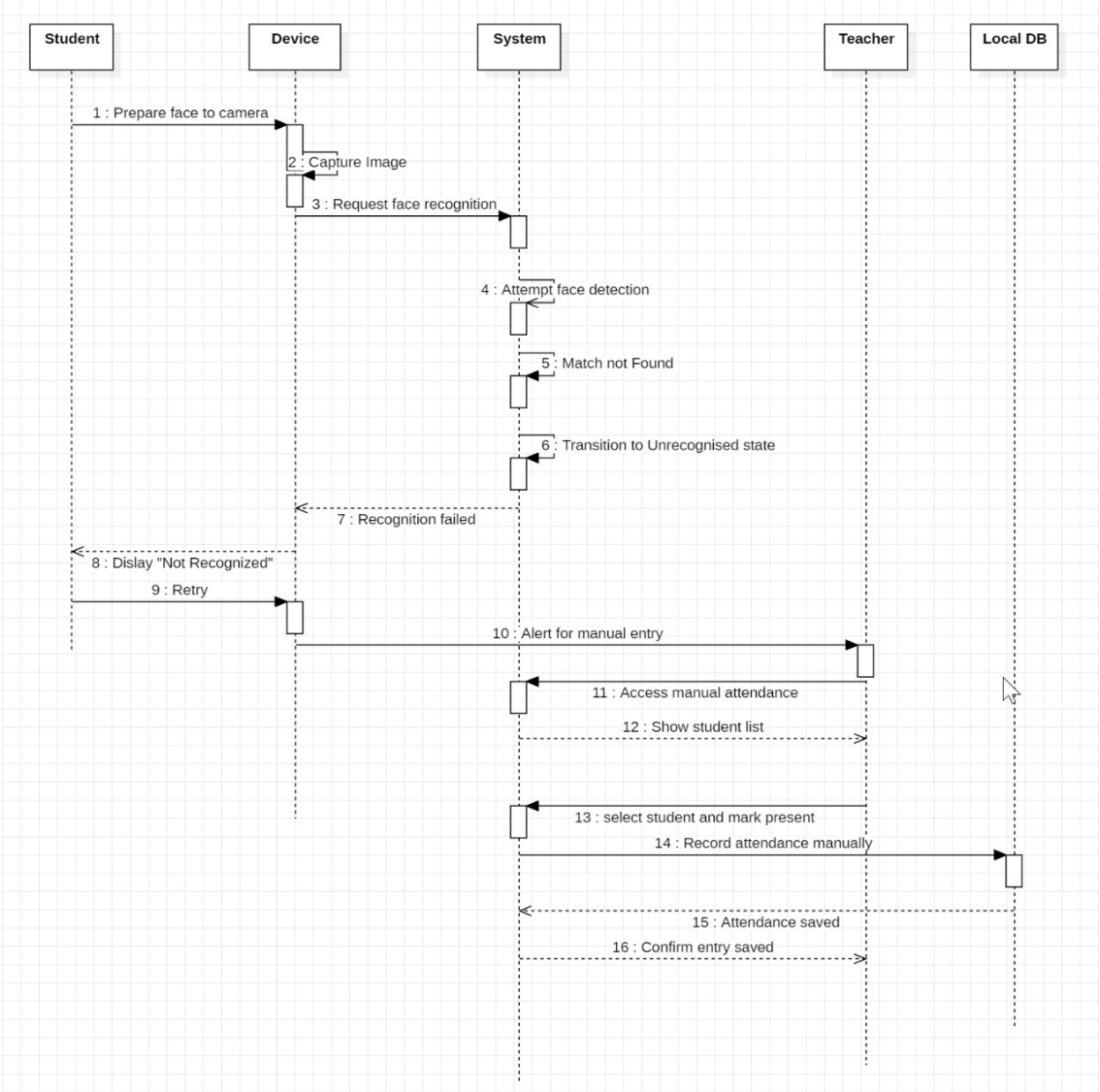


Fig 5.2.2 Handle Unrecognized Student

This diagram walks through what happens when the system cannot recognise a student's face during attendance taking.

1. The student stands in front of the camera, ready to have their attendance recorded.
2. The device captures the student's image and sends it to the system for face recognition.
3. The system tries to detect and match the face, but no match is found in the database.
4. Because the system cannot identify the student, it moves into an "unrecognized" mode.
5. The device then shows a "Not recognized" message and may try again briefly.
6. When recognition still fails, the system notifies the teacher that manual attendance is required.
7. The teacher opens the manual attendance feature, which displays a list of all students.
8. The teacher selects the correct student from the list and marks them as present.
9. This manual entry is then saved to the local database, and the system confirms the update.

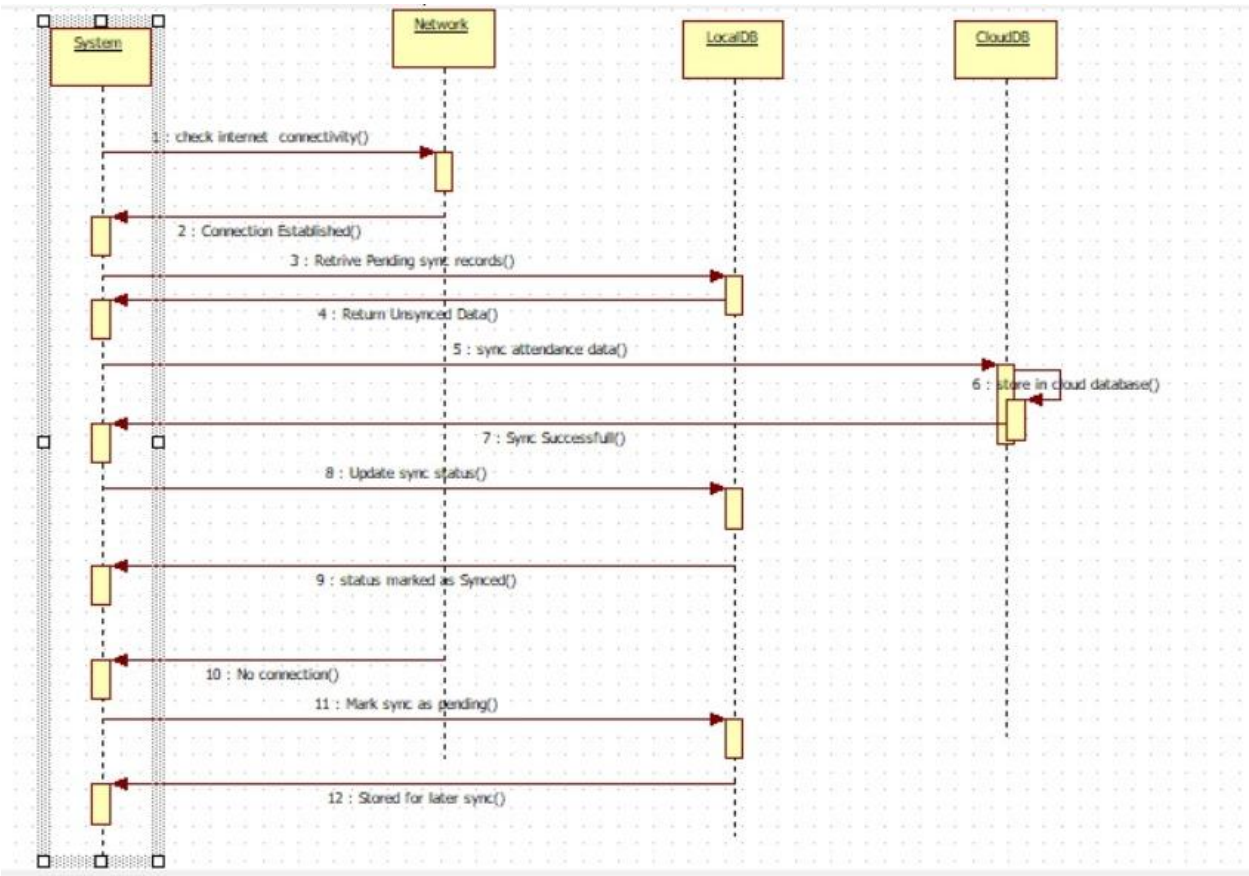


Fig 5.2.3 Sync Attendance Data

The diagram shows how the system checks for internet and syncs attendance data between the local database and the cloud.

1. The system first checks internet connection.
2. If the internet is available, it connects and asks the local database for any attendance records that still haven't been synced.
3. The local database sends back the unsynced data.
4. The system then sends this data to the cloud database so it can be stored online.
5. Once the cloud confirms that the data was saved successfully, the system updates the local database and marks those records as "synced."
6. If there is no internet connection, the system marks the records as "pending" and keeps them stored locally until the next time it can sync.

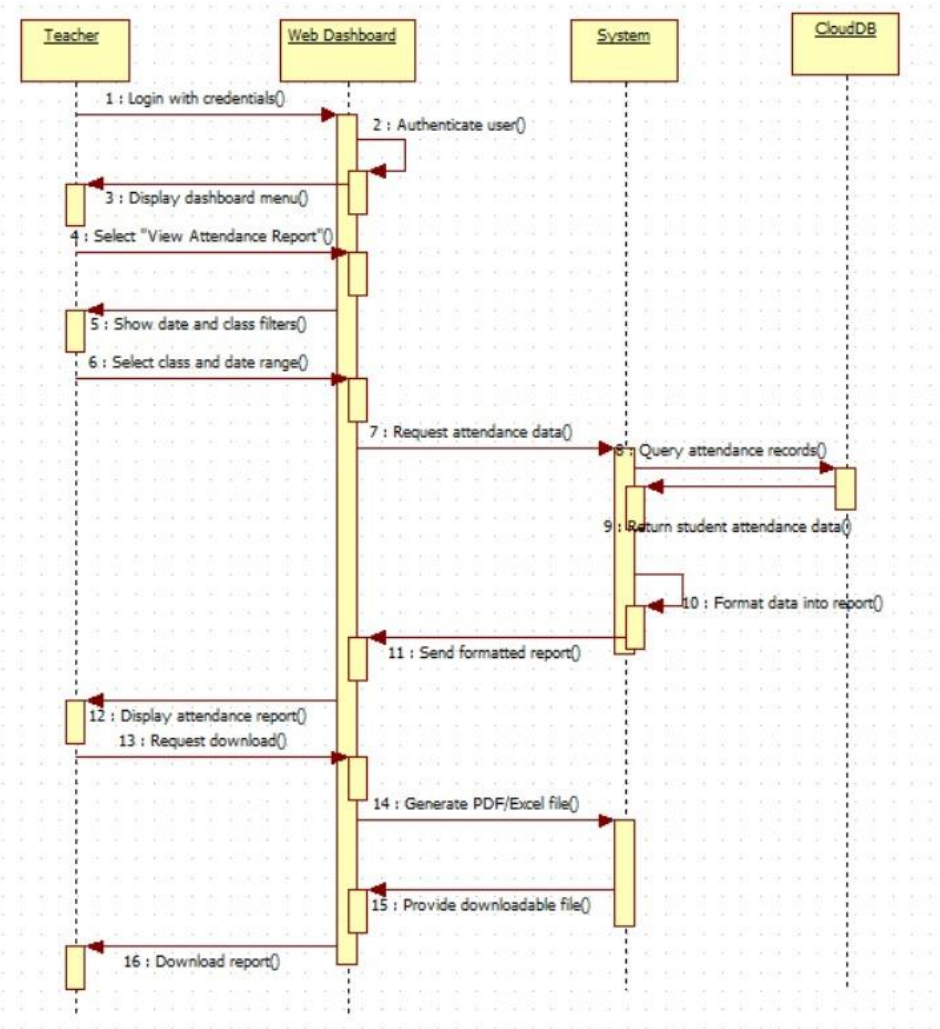


Fig 5.2.4 Teacher Views Attendance Report

This diagram shows how a teacher uses the system to view and download an attendance report.

1. The teacher begins by logging into the web dashboard using their username and password.
2. The system checks the login details and confirms the teacher's identity**.**
3. Once logged in, the dashboard opens and shows the main menu**.**
4. The teacher chooses the option to view attendance reports**.**
5. The dashboard then displays filters, allowing the teacher to pick a class and a date range.
6. After selecting the desired filters, the dashboard sends a request to the system for the relevant attendance data.
7. The system then contacts the cloud database to pull the matching attendance records.
8. The cloud database returns the data, and the system organizes it into a readable report format**.**
9. The formatted report is sent back to the dashboard, where the teacher can now see the attendance summary**.**
10. If the teacher wants a copy, they click download.
11. The system generates a downloadable file, usually in PDF or Excel format.
12. The file is made available, and the teacher downloads the report to their device.

5.3 Activity Diagram:

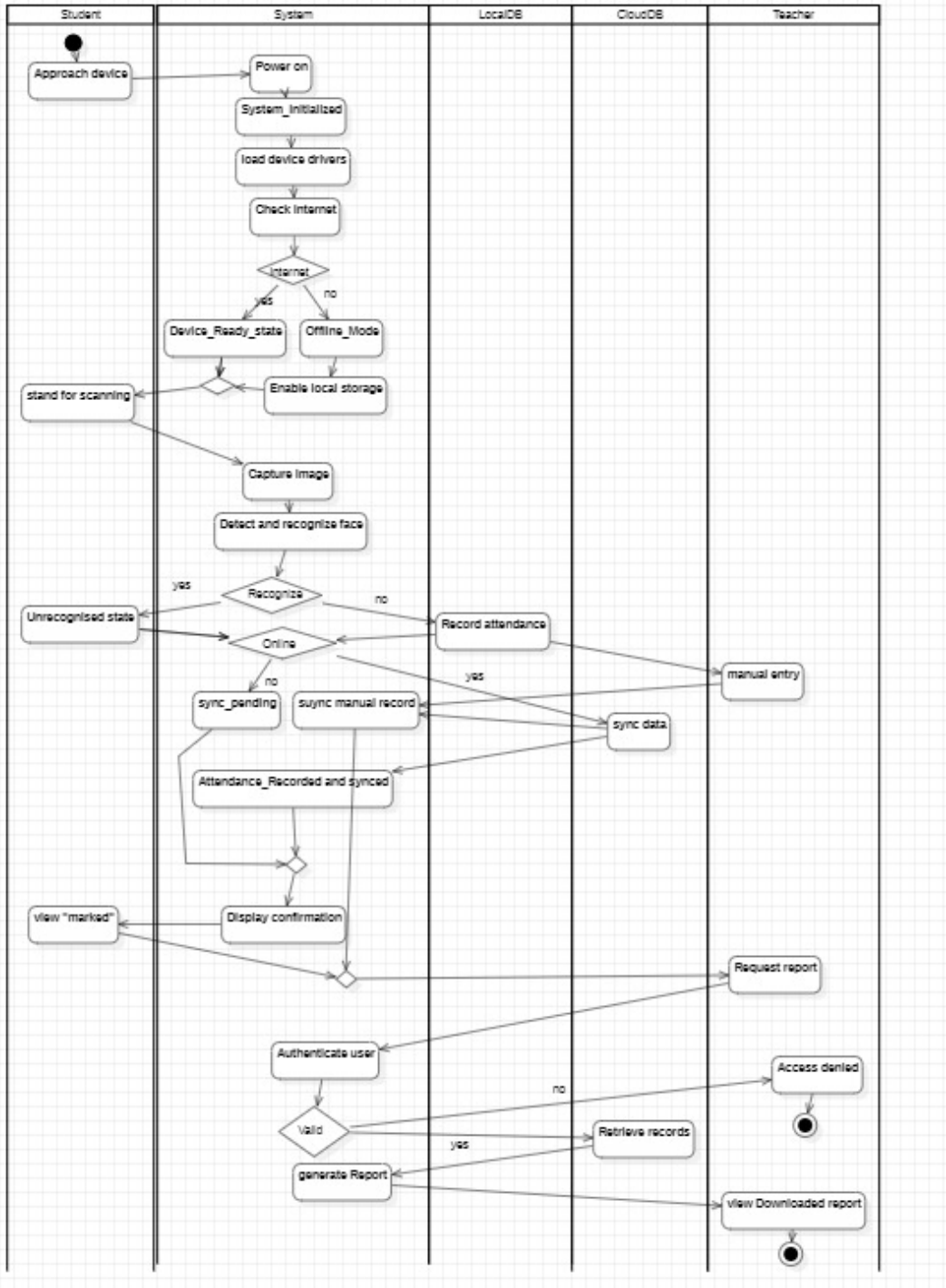


Fig 5.3.1 Complete Attendance System Flow

- Student Interaction
    - A student walks up to the attendance device.
    - The device powers on, loads its drivers, and checks for internet connectivity.
- System Readiness
    - If the internet is available, the device goes into Online Mode.
    - If the internet is *not* available, it switches to Offline Mode, enabling local storage so attendance can still be recorded.
- Capturing Attendance
    - The student stands in front of the device.
    - The system captures an image and tries to recognize the student's face.
    - If the system cannot recognize the student, they are marked as "unrecognized".
- Recording Attendance
    - If the face is recognized, the system records the attendance.
    - In online mode, the data is immediately sent to the cloud.
    - In offline mode, data is saved locally and marked for syncing later.
    - When the internet becomes available again, all pending records are synced automatically.
- Teacher Manual Entry (Fallback)
    - If recognition fails or the system requires it, a teacher can manually enter the attendance in the cloud.
- Confirmation to Student
    - After the attendance is recorded, the system shows a confirmation message indicating their attendance has been marked successfully.
- Teacher Report Access
    - A teacher can later request an attendance report.
    - They must authenticate themselves.
    - If successfully authenticated, the system retrieves the attendance records and generates a downloadable report.
    - If authentication fails, the system denies access.

Chapter 6: UI Design with Screenshots

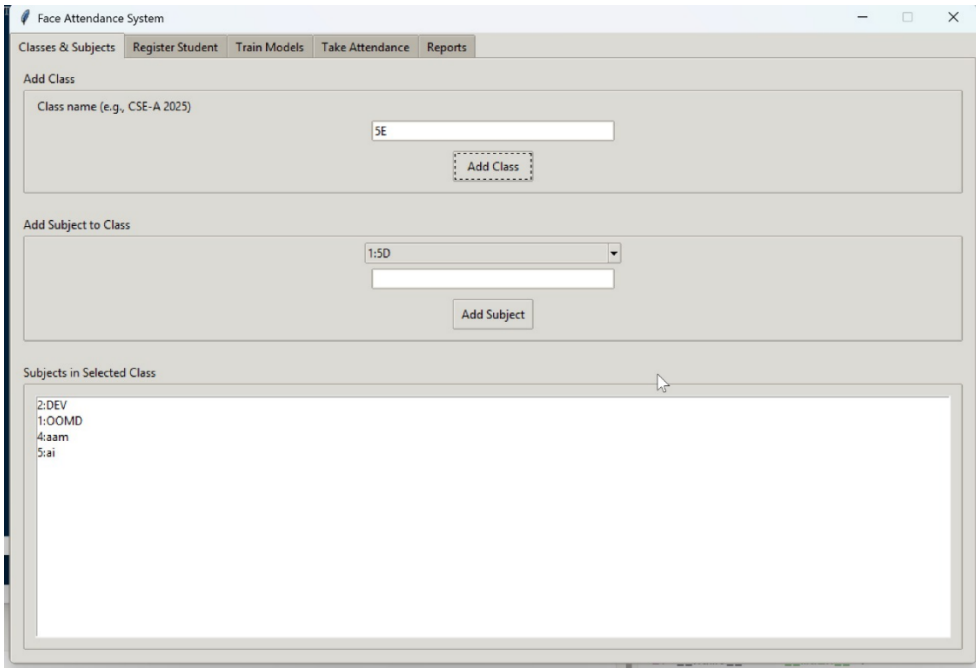


Fig 6.1 Adding a particular Class

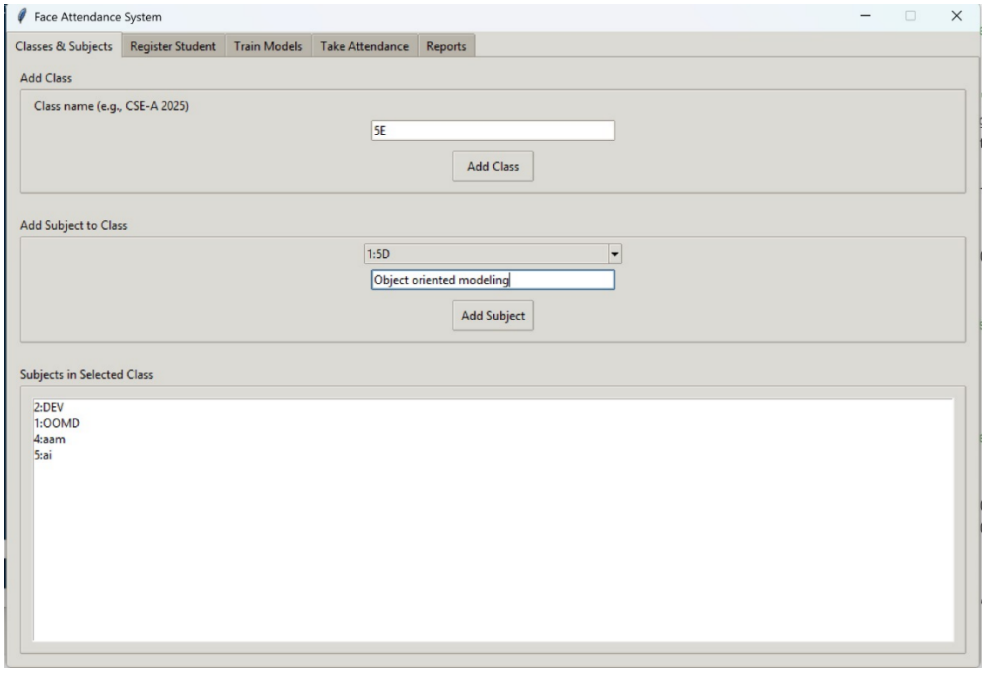


Fig 6.2 Adding Subject to Selected Class

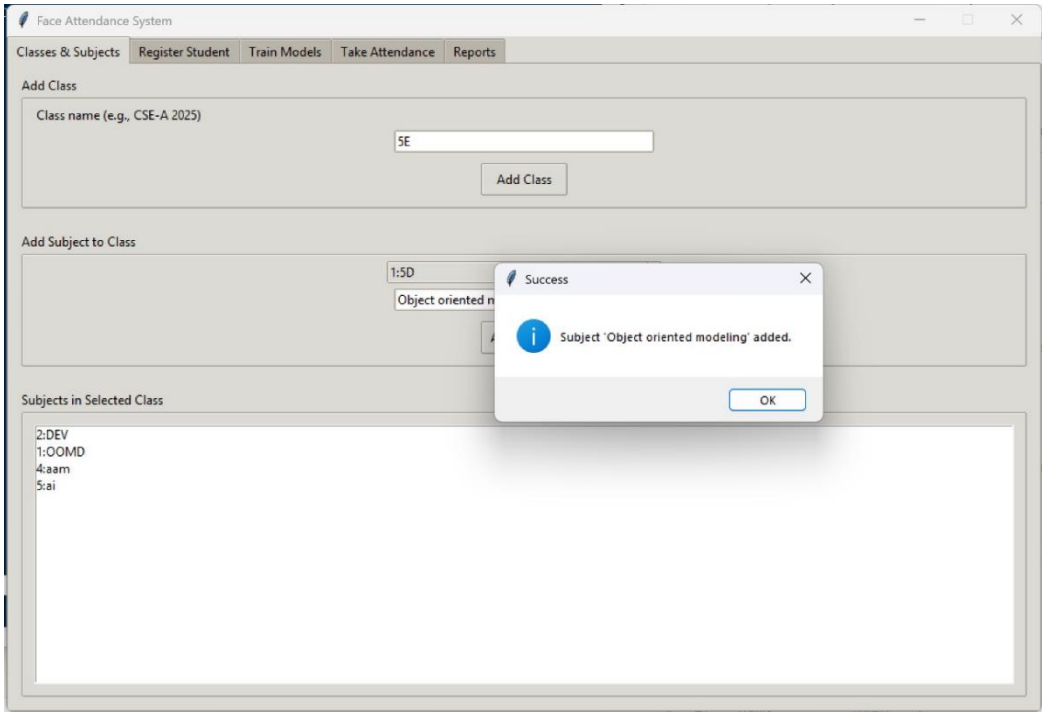


Fig 6.3 Subject Added Output Screen

Fig 6.4 Subjects in Selected Class Output Screen

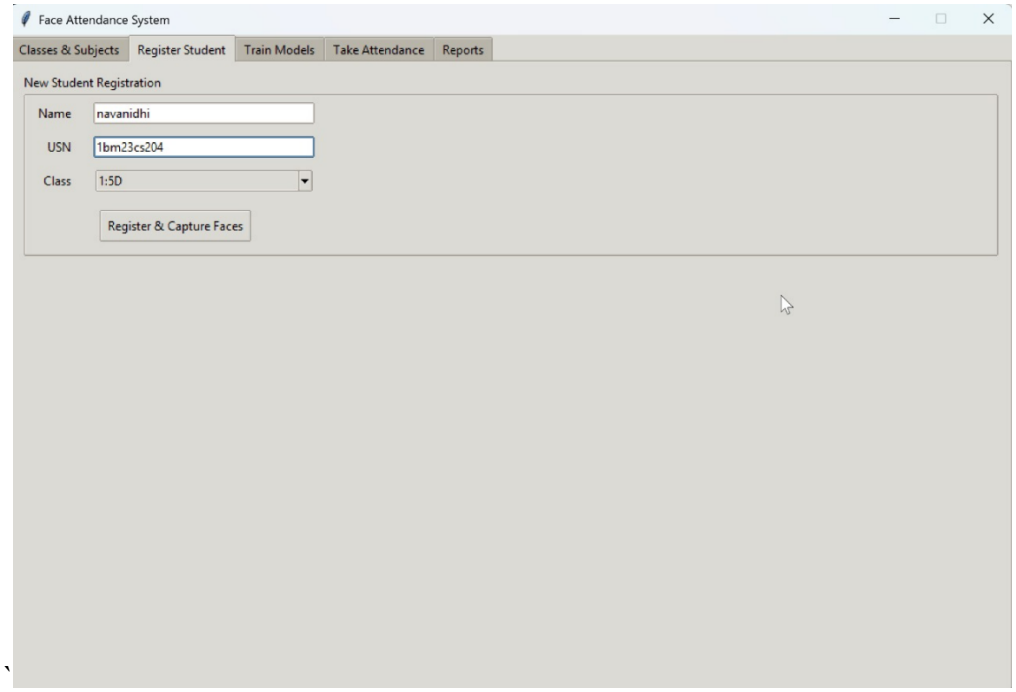


Fig 6.5 Student Registration Screen