FINAL YEAR PROJECT

EduFace

Facial Recognition for Student Attendance and Visitor Login

An offline-first system that runs on the school's own server

- Case Study: St. Charles Lwanga International Secondary School, Kakiri



Presented by Mutesi Patricia

Diploma in Software Engineering, ISBAT University

Supervisor: Mr. Tobias Kakooza

eduface.rwenzorilabs.com

Overview of the Project

The problem

Attendance and visitor registration at the school are mostly manual. One student can sign in for an absent friend, paper visitor records are slow to check, and a system that depends on the internet can fail when the connection drops.

The aim

To design and develop an offline-first facial recognition system for managing student attendance and visitor login at the school.

What it does

- Students scan a QR card or their face at a gate kiosk
- Faces are used only with guardian consent, and a head turn proves a live person
- Visitors get QR badges, a watch list and an overstay flag
- Guardians are told about late, absent or early-leaving students

QR or face
two ways to sign in at the gate

25
tables in MySQL 8

150+
automated tests in the project

Live
eduface.rwenzorilabs.com

Module Description

1 Setup and users

School setup with the standard classes for nursery, primary and secondary, and roles with sign-in lockout.

2 Students and guardians

Classes, students and guardians, CSV import, printable QR ID cards and unique issued ID numbers.

3 Scanning and attendance

Late cut-off, duplicate window, departures, corrections with a reason, and Close day.

4 Face recognition

Detects, aligns and compares faces on the server CPU. Matches only with a threshold and a margin.

5 Consent and privacy

Guardian consent first. Only encrypted templates are stored, and withdrawing consent erases them.

6 Anti-spoofing

A random head-turn challenge, and an optional image model, so a photo is rejected.

7 Visitors and badges

Check-in with QR badges, a watch list, overstay flags and an evacuation roll call.

8 Gate kiosks and messages

Paired gate tablets with a fixed direction, and a guardian message queue sent by one tap.

9 Reports, backups and audit

CSV and print reports, encrypted backups, data retention, a system check and a tamper-evident audit log.

Operational Environment



Software

- Python 3.12 with FastAPI and Uvicorn
- MySQL 8.0 for the database
- ONNX Runtime for the face and liveness models
- HTML, CSS and JavaScript in any modern browser
- Windows: started with run.bat

Hardware

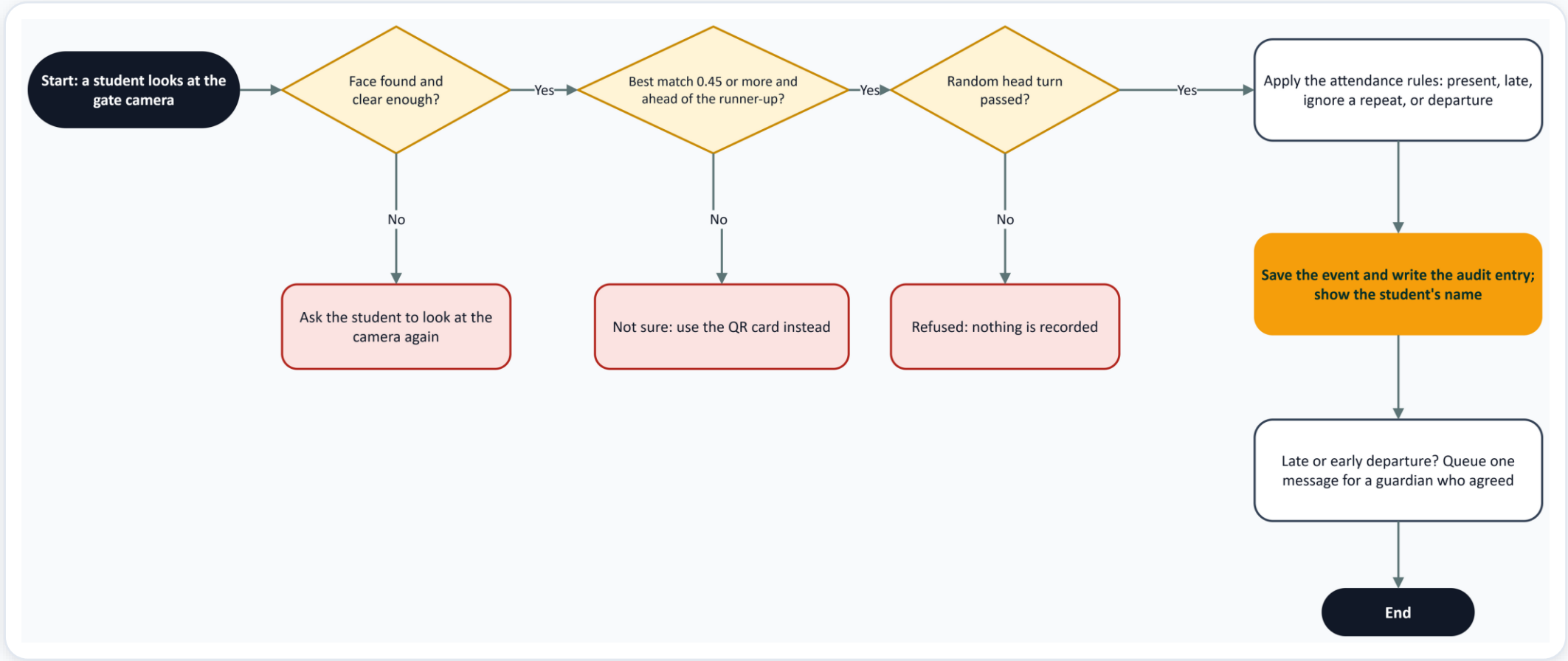
- A school server or computer; faces run on the CPU, about 0.1 seconds per frame
- A tablet or computer with a camera at each gate
- Local network and backup power
- Storage for encrypted backups

Runs online on

- Hostinger VPS with Ubuntu 24.04
- Docker: the app and MySQL 8
- Caddy gives automatic HTTPS, which the camera needs
- eduface.rwenzorilabs.com

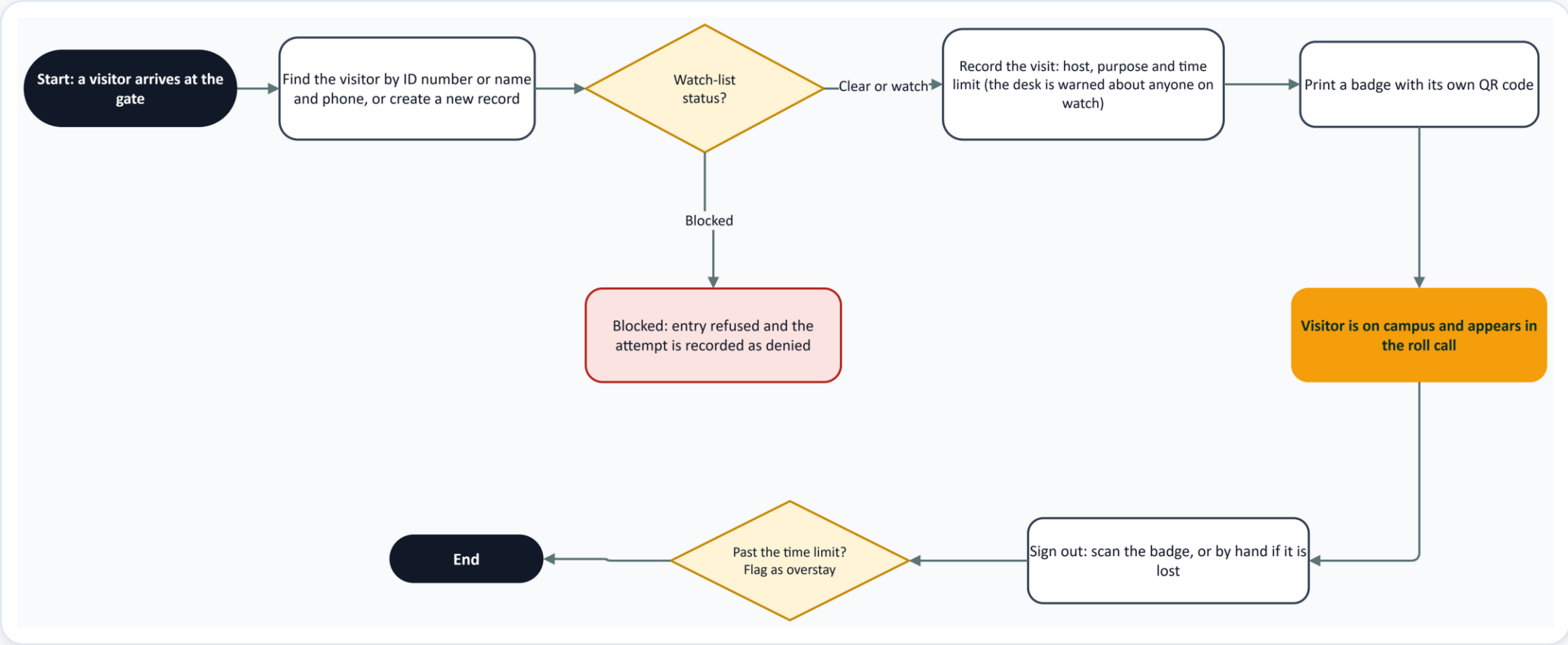
Security: Argon2id passwords, AES-256-GCM face templates, no face images stored, CSRF and CSP protection, encrypted backups, hash-chained audit log

Flowchart: Face Scan at the Gate



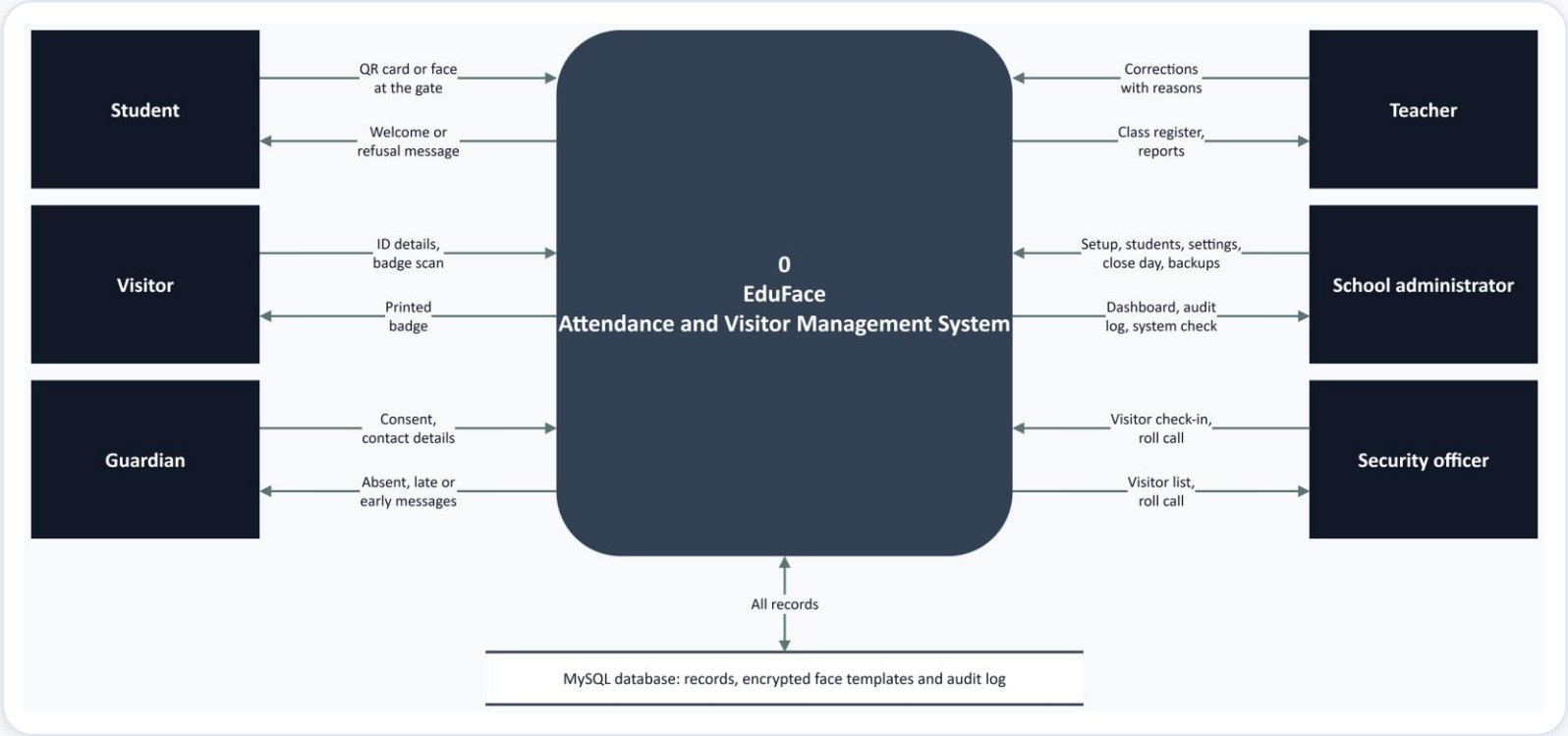
A face is accepted only if it is clear, clearly the best match, and a live head turn passes. Otherwise the student uses the QR card.

Flowchart: Visitor Check-in and Check-out



Blocked visitors are refused and the attempt is still recorded. A badge QR code can never sign a student out, and a student card can never sign a visitor out.

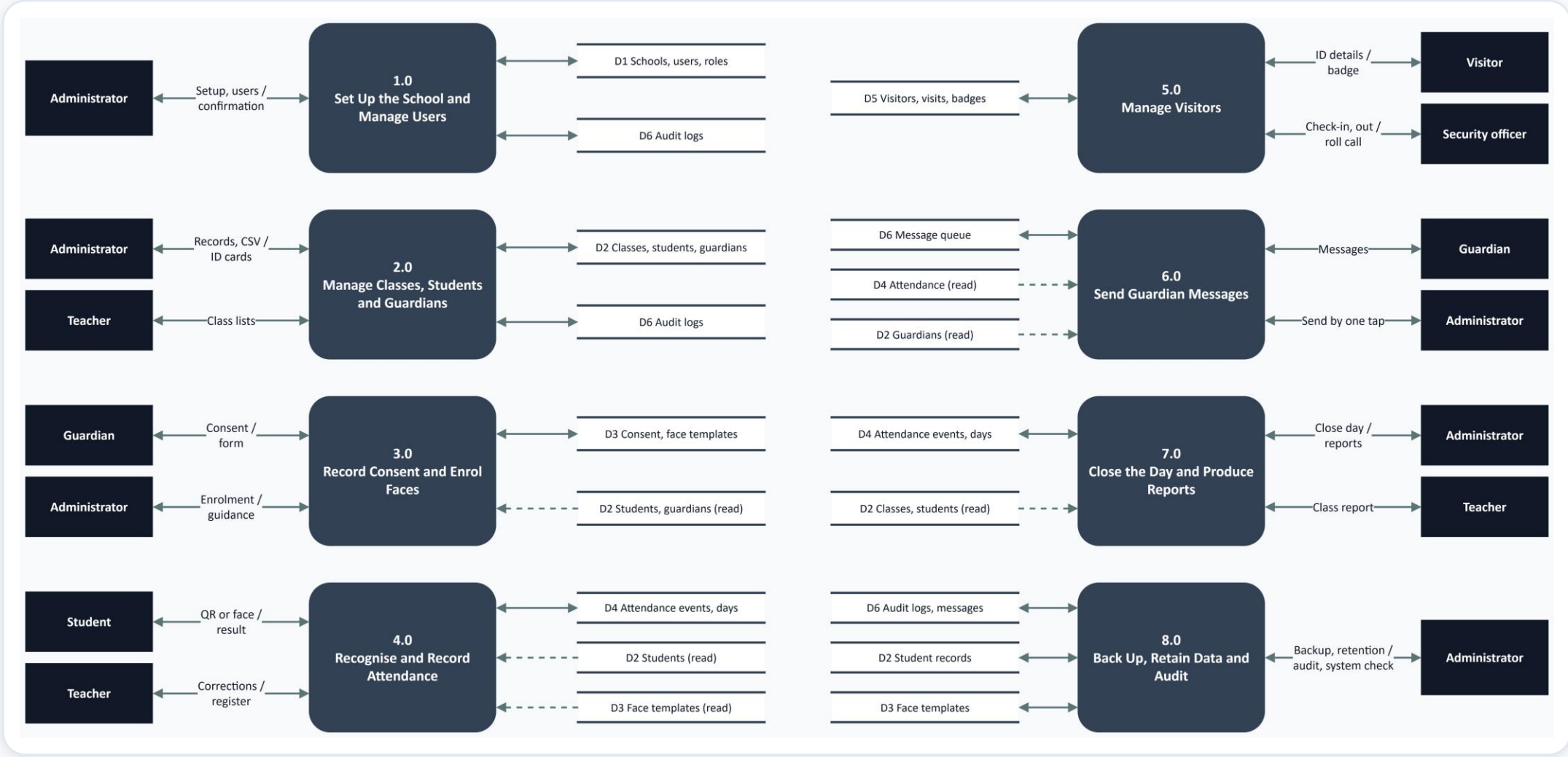
Data Flow Diagram: Level 0



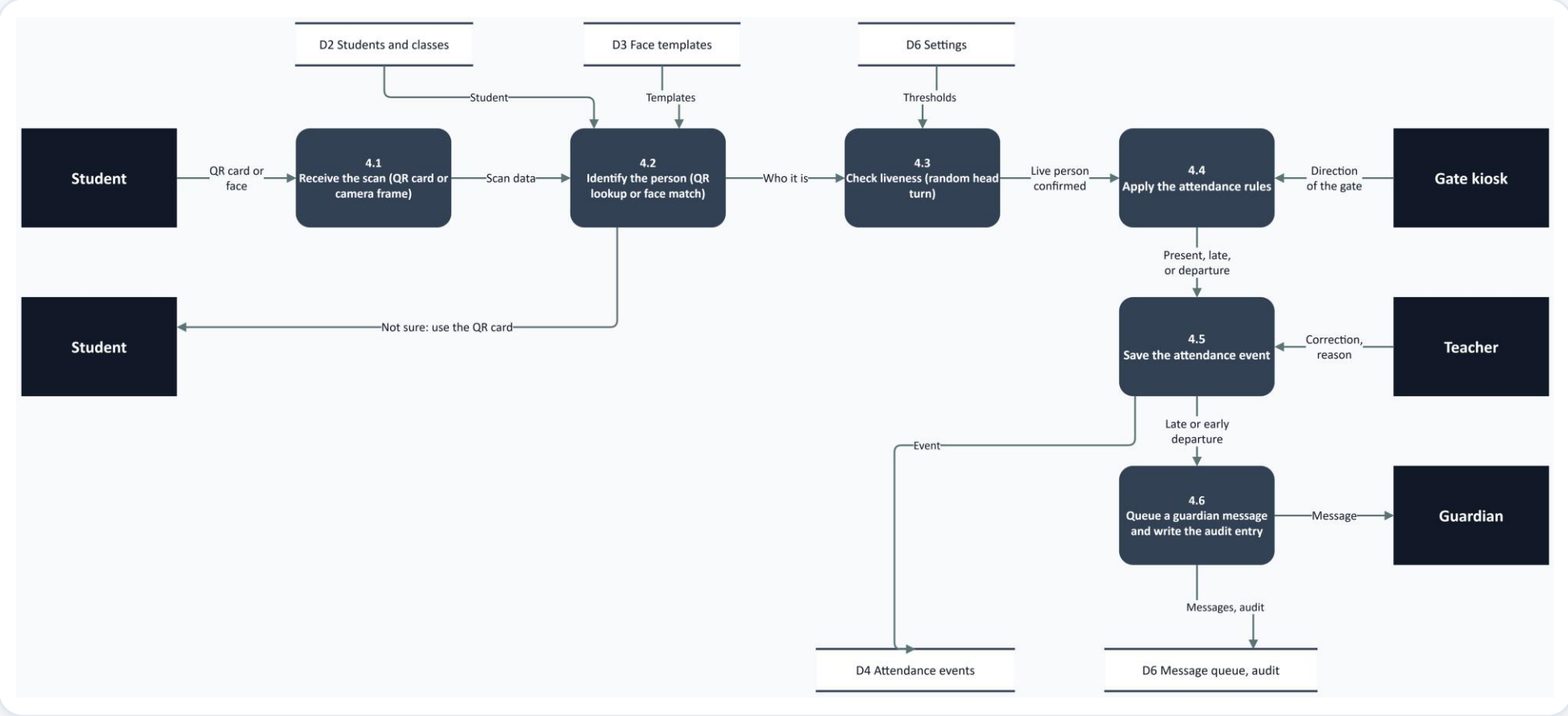
Outside party Process Data store Data flow

Level 0 shows the whole system as one process, with six kinds of people and one database.

Data Flow Diagram: Level 1

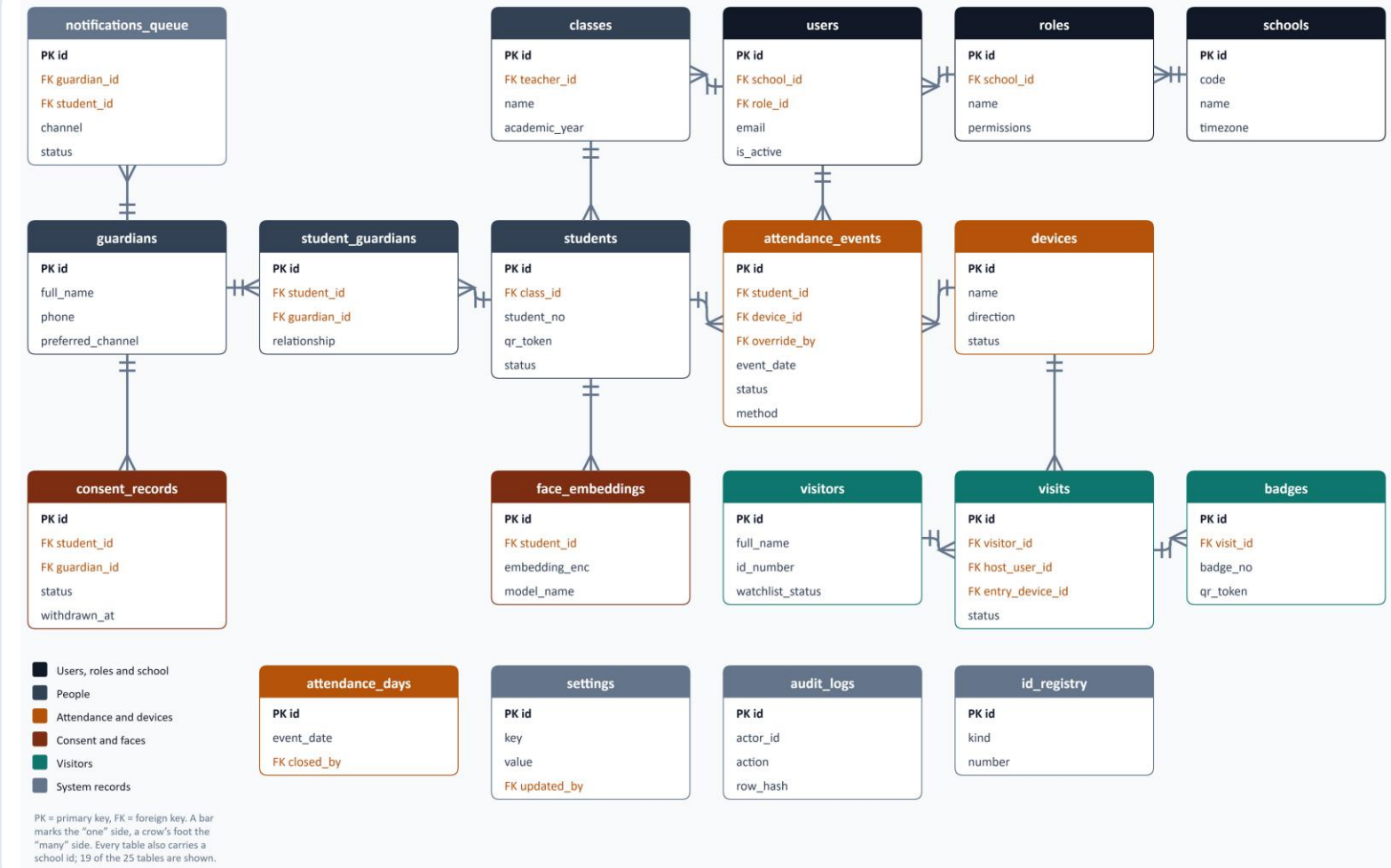


Data Flow Diagram: Level 2



Process 4.0, recognise and record attendance, in six steps. A face passes the match and liveness checks before the attendance rules run.

E-R Diagram



25

tables in MySQL 8, with unique ID numbers issued by one registry

Append-only

attendance events are never edited; corrections add a new row

Encrypted

face templates are stored as ciphertext and no face image is kept