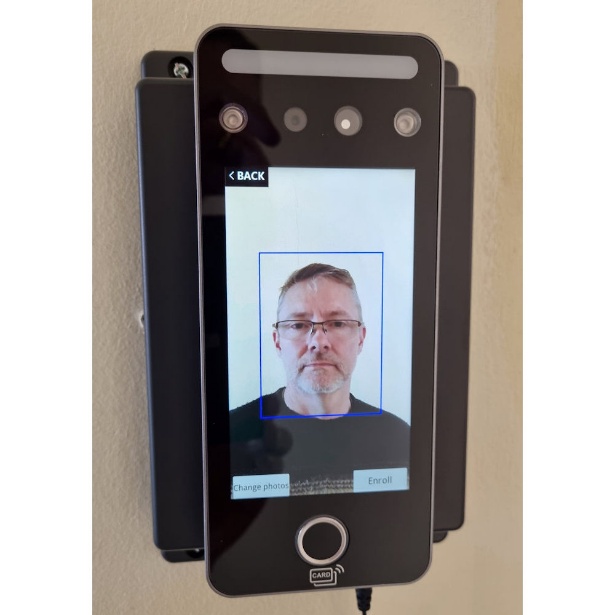
# **Project Overview: Face Recognition Attendance System**

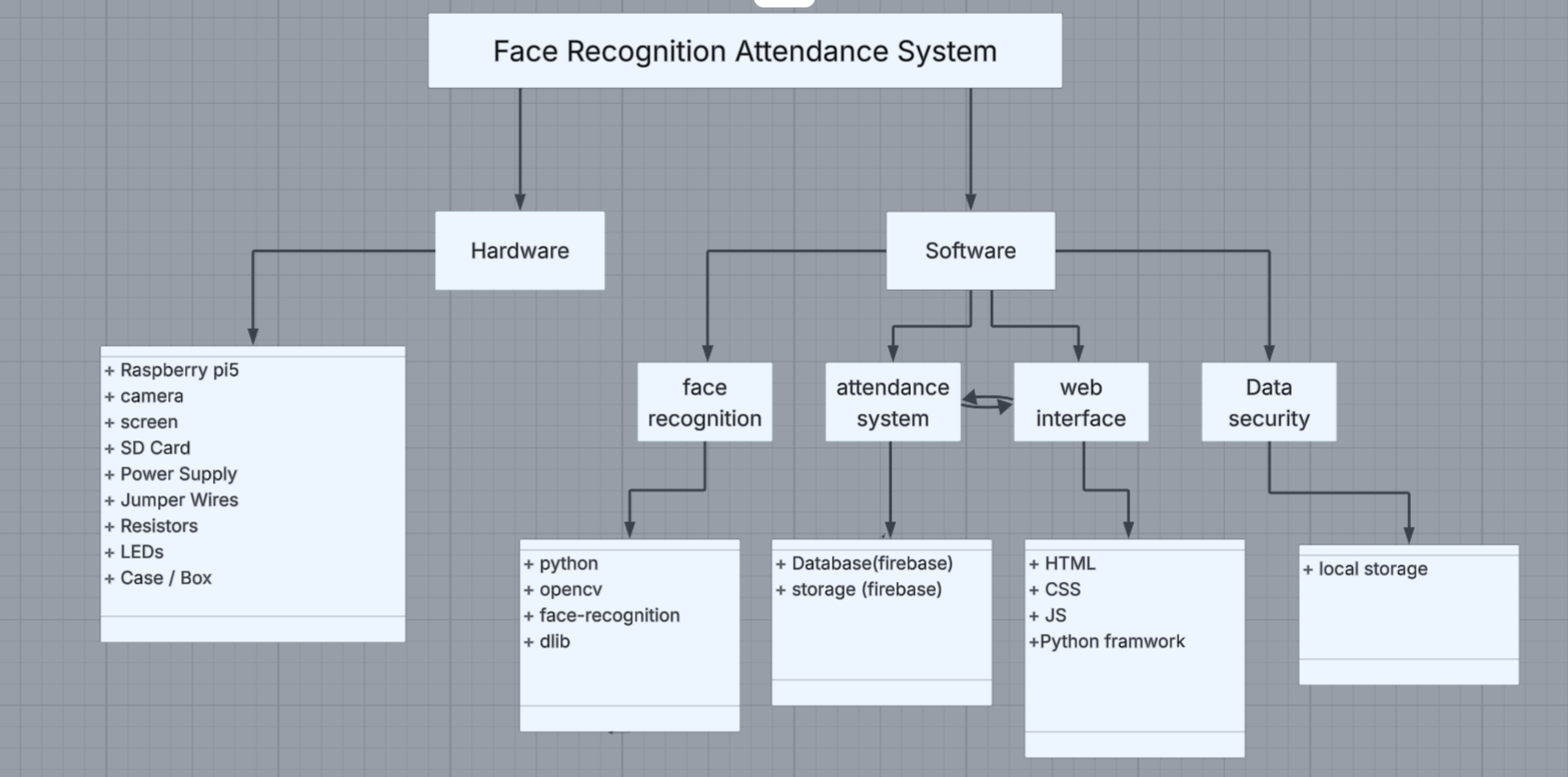
## **Main Idea:** A smart attendance system that automatically records students’ attendance using **face recognition** through a **Raspberry Pi 5** and a **camera**, connected to **Firebase** for database and storage management. The system includes a **web interface** with two panels — one for **admins** and one for **students.**

* **Hardware:** 
  + **Raspberry Pi 5** – main processing unit.
  + **Camera Module** – captures the student’s face at the entrance.
* **Software & Libraries:**
  + **Programming Language:** Python.
  + **Libraries:**
    - opencv – for image processing and camera control
    - dlib and face\_recognition – for facial feature extraction and recognition.
* **Database & Storage:**
  + **Firebase Realtime Database**: Stores student data such as name, student ID, attendance count, and absences.
  + **Firebase Storage**: Stores face images organized by student ID. Each student can update their photo every semester or every year to keep the database up to date.
* **Web Interface:**
  + **Admin Panel:** Displays statistics such as:
    - Number of students present today.
    - Attendance percentage per lecture.
    - Students with high absence rates (warnings).
    - Allows data visualization and reporting.
  + **Student Panel:** 
    - Login with username and password.
    - Upload a new profile photo each semester or year.
    - View attendance records and receive warnings if absence limits are reached.
* **Fallback Mechanism:** If the system fails to recognize a student’s face, it allows verification using **student ID.**

The figure below illustrates an **approximate design of the proposed device**. This image is taken from an external source and is used for illustrative purposes only. The actual prototype may differ in terms of components, external case, or wiring layout.



**General Idea**

****

**Full Process**

When a student approaches the lab door, the camera captures their face, the Raspberry Pi processes the image using AI algorithms, and then determines whether the student is registered.

If recognized → "Pass" is displayed on the LCD and the green LED turns on.

If not recognized → "Fail" is displayed on the LCD and the red LED turns on.

**Workflow**

1. System Power-On
   1. The Raspberry Pi is powered using a 5V–5A supply.
   2. The micro SD card contains the OS (Raspberry Pi OS) and the Python attendance program
2. Image Capture
   1. The camera (Pi Camera Module) captures an image of the student at the door
3. Image Processing
   1. The Raspberry Pi runs Python code with OpenCV and face\_recognition libraries to detect and compare the face against a local database stored on the SD card.
4. Verification
   1. If the face matches → result is "Pass"
   2. If not recognized → result is "Fail"
5. Output Display
   1. The LCD (16x4) displays the student’s name and status (Pass/Fail).
   2. Green LED lights up for success.
   3. Red LED lights up for failure.
6. Data Storage
   1. Attendance logs (student name, date, and time) are stored in **Firebase Database**, and student images are uploaded to **Firebase Storage** for later verification and web access