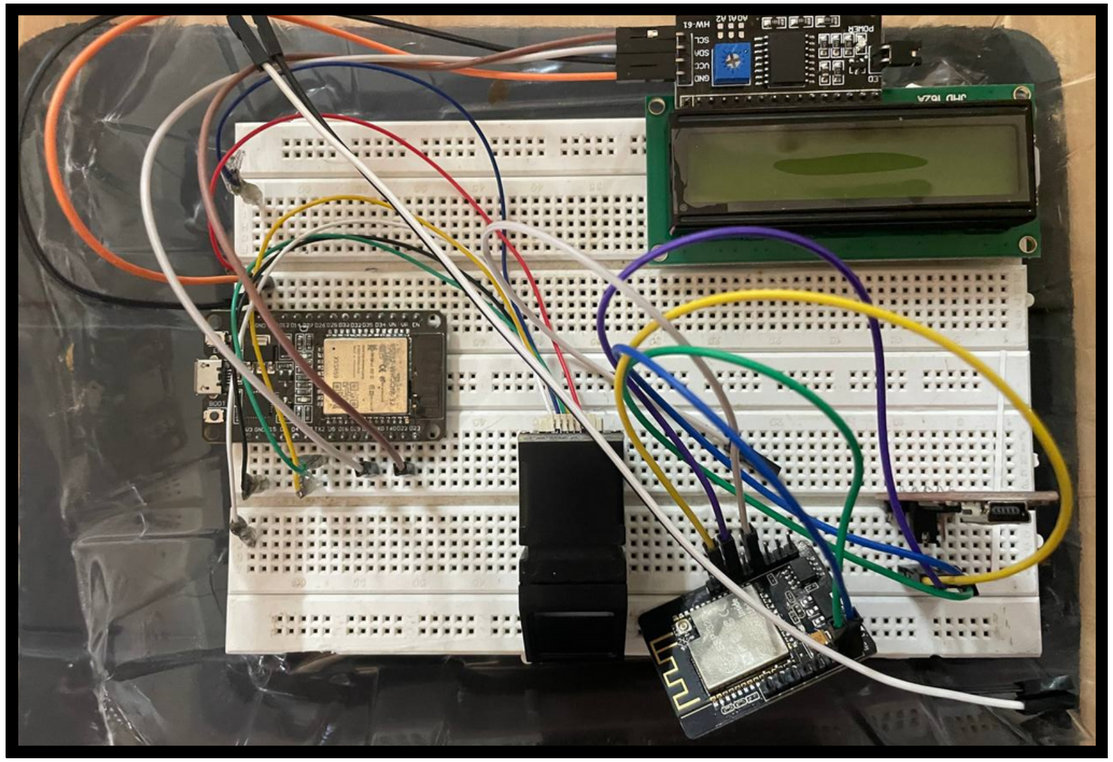
Smart Attendance System

A Combination of Hardware(Iot), Web Service, App Service, Machine Learning, ComputerVision, Biometric Detection



Features

- ML Based Face Recognition
- Optical Fingerprint Sensor For Biometric Detection
- Website for easy access
- Application based Services
- Advanced Authentication Enhanced Security
- Cross Platform Support
- Rapid Data Transfer
- Integrated Display
- User Friendly UI/UX

Components And Services Used

Hardware

- Esp32
- FTDI Module
- R307 Fingerprint Sensor
- Esp32 Cam-module
- LCD Display
- I2C Convertor
- C++
- Embeded C

Components And Services Used

Web

- HTML
- CSS & BootStrap
- JAVA Script
- Google Sheets
- Supabase Authentication
- REACT.JS
- VITE.JS

Components And Services Used

Application

- Flutter
- DART
- Android Studio
- FireBase
- Google Sheets
- Google Drive API

Components And Services Used

Machine Learning

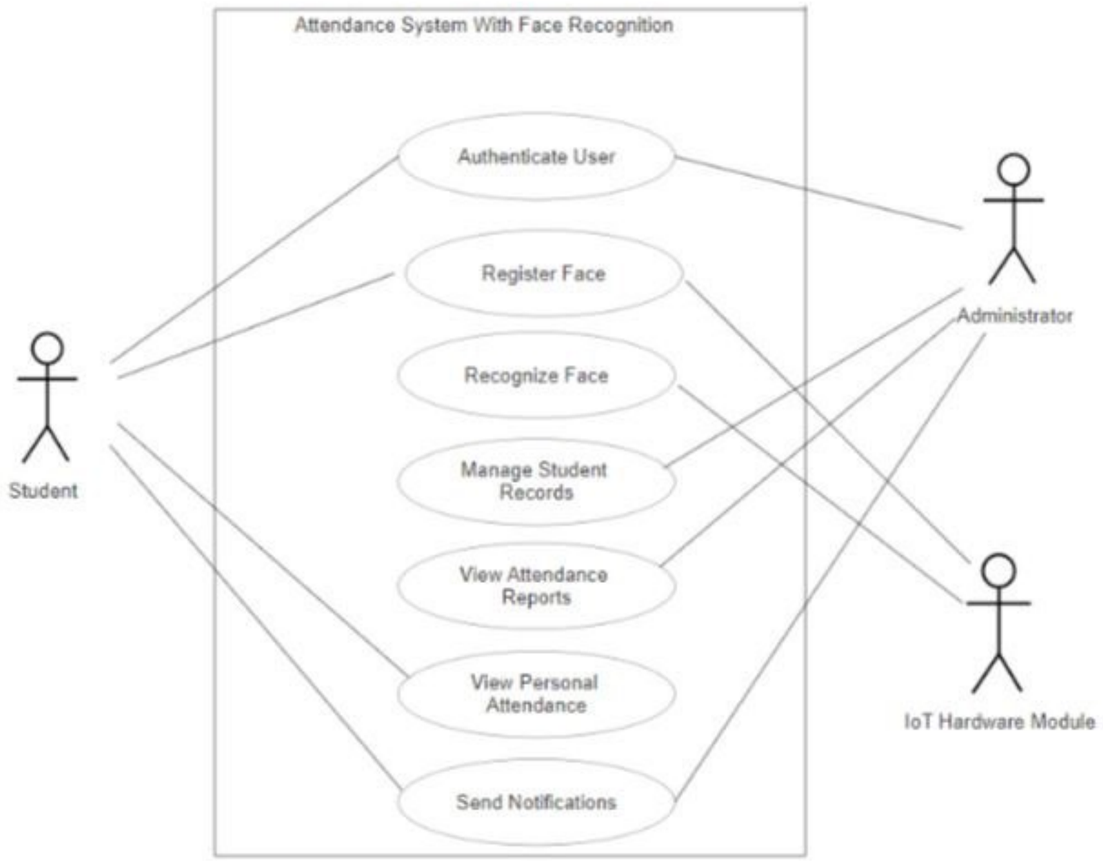
- Python3
- OpenCV
- Face_Recognition
- Google Collab

Components And Services Used

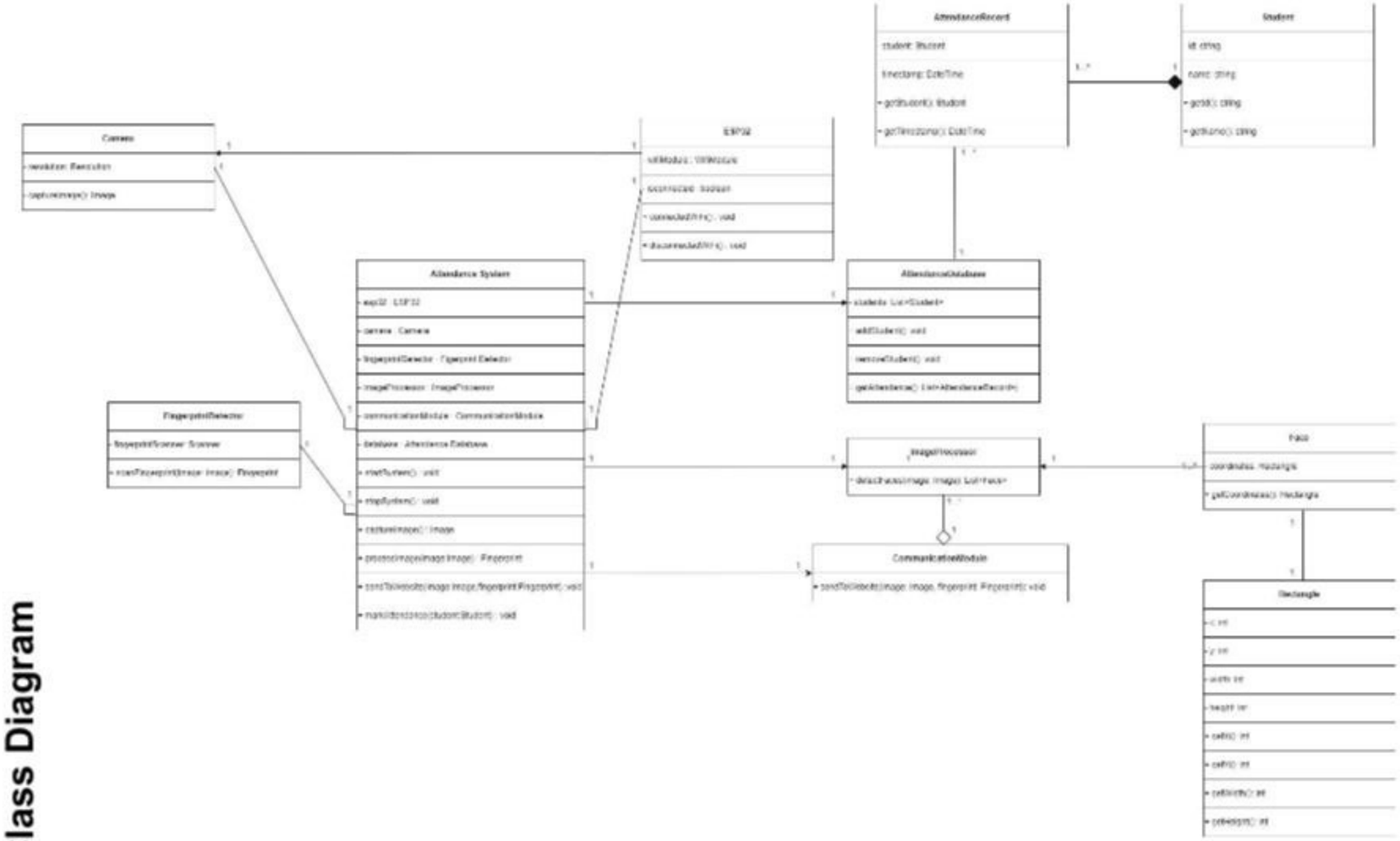
API

- IFTTT APIs
- Google Drive API
- Google Sheets API
- Supabase API

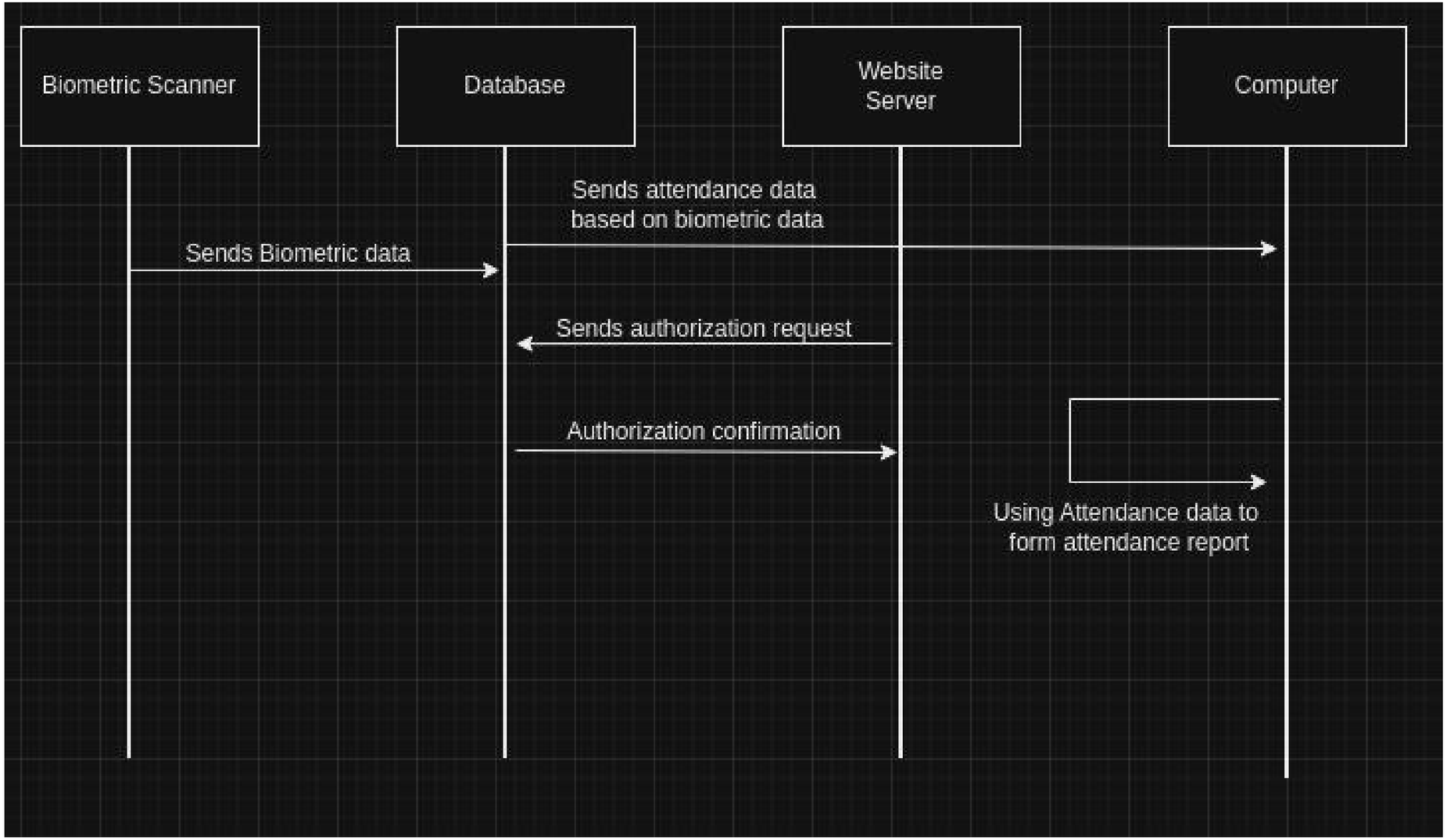
Use Case Diagram



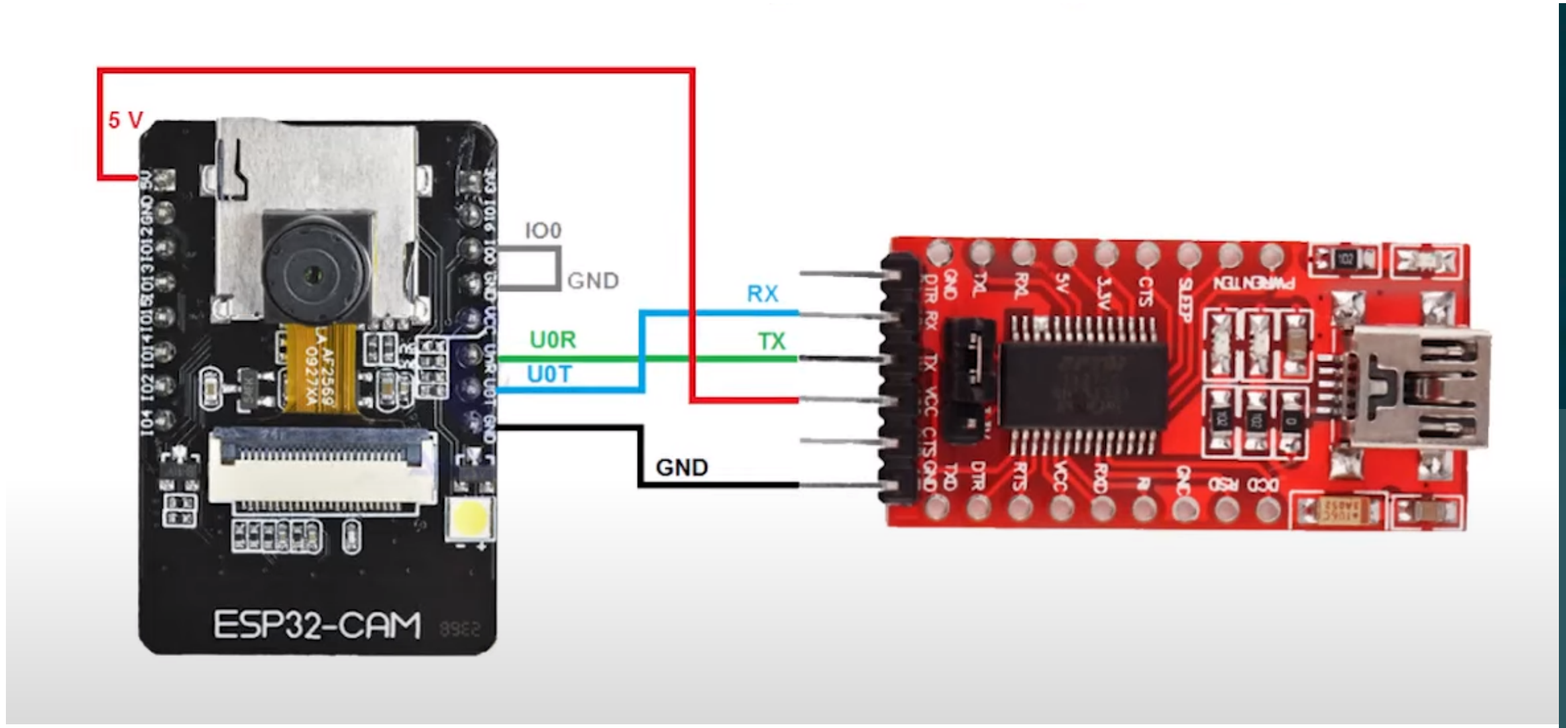
Class Diagram

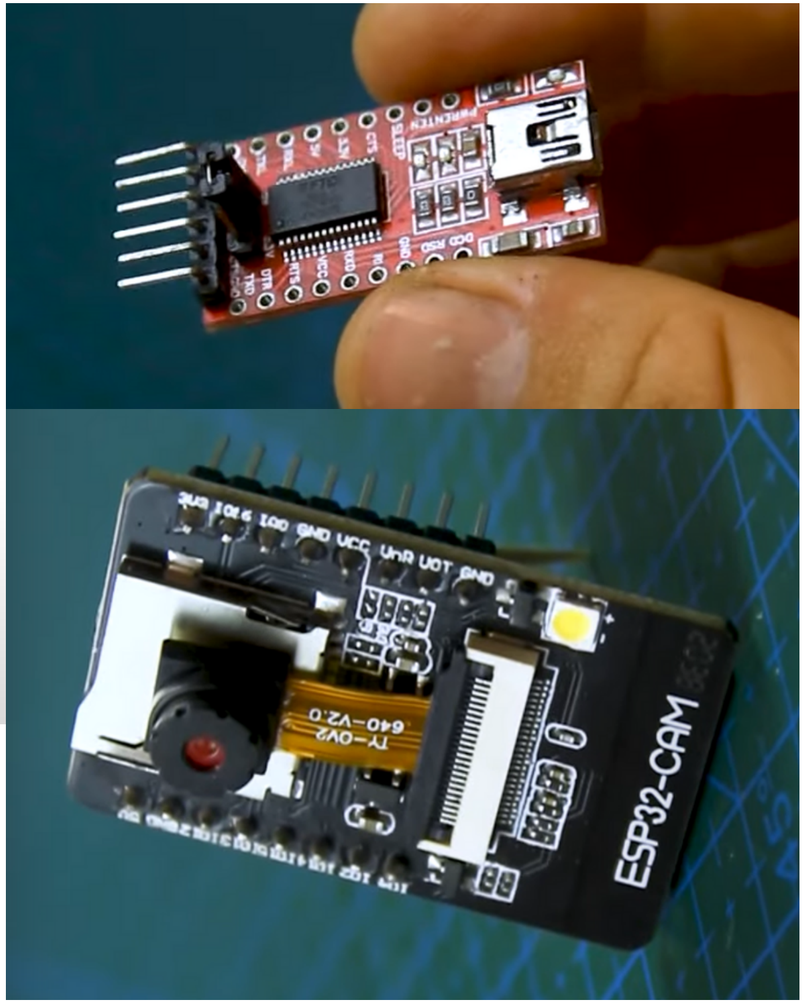


Sequence Diagram

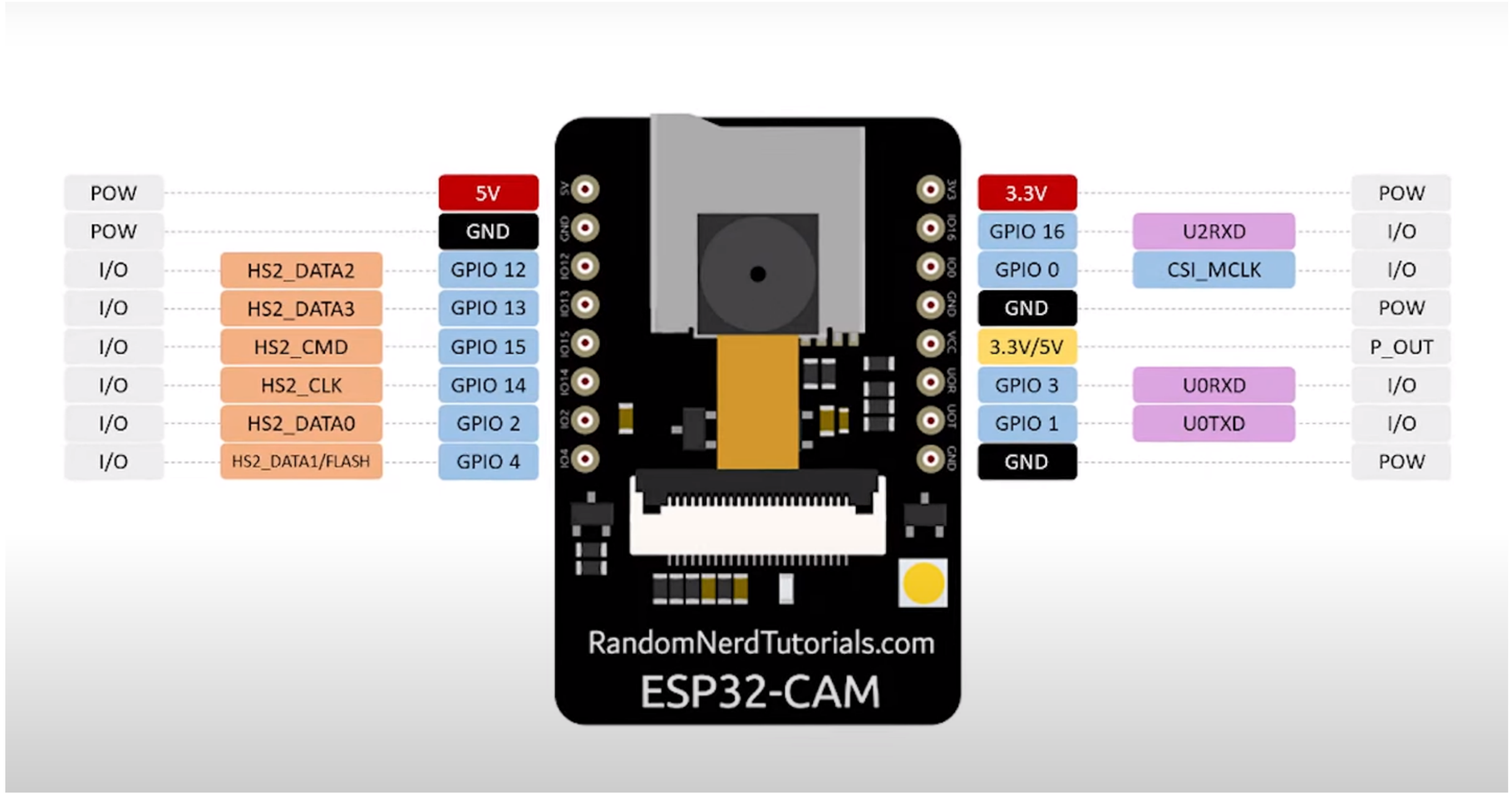


Face Recognition





Face Recognition



Libraries Used

OpenCV
requests
numpy
pandas
face_recognition
cmake tools
dlib

Usage of OOPs

Class Instances: The code defines and uses instances of classes. For example:
pd.
DataFrame is a class instance used to create a DataFrame (df).
cv2.VideoCapture is a class instance used to capture video frames (cap).
face_recognition methods are used through instances, like face_recognition.face_encodings and face_recognition.face_locations.

Encapsulation: The functions findEncodings and markAttendance encapsulate specific functionalities, making the code modular and readable.

Inheritance: There are no explicit examples of inheritance in the code.

Polymorphism: Polymorphism allows objects of different types to be treated as objects of a common type. While not explicitly shown in the code, some functions and methods, like cv2.cvtColor and cv2.waitKey, might exhibit polymorphic behavior, accepting arguments of different types.

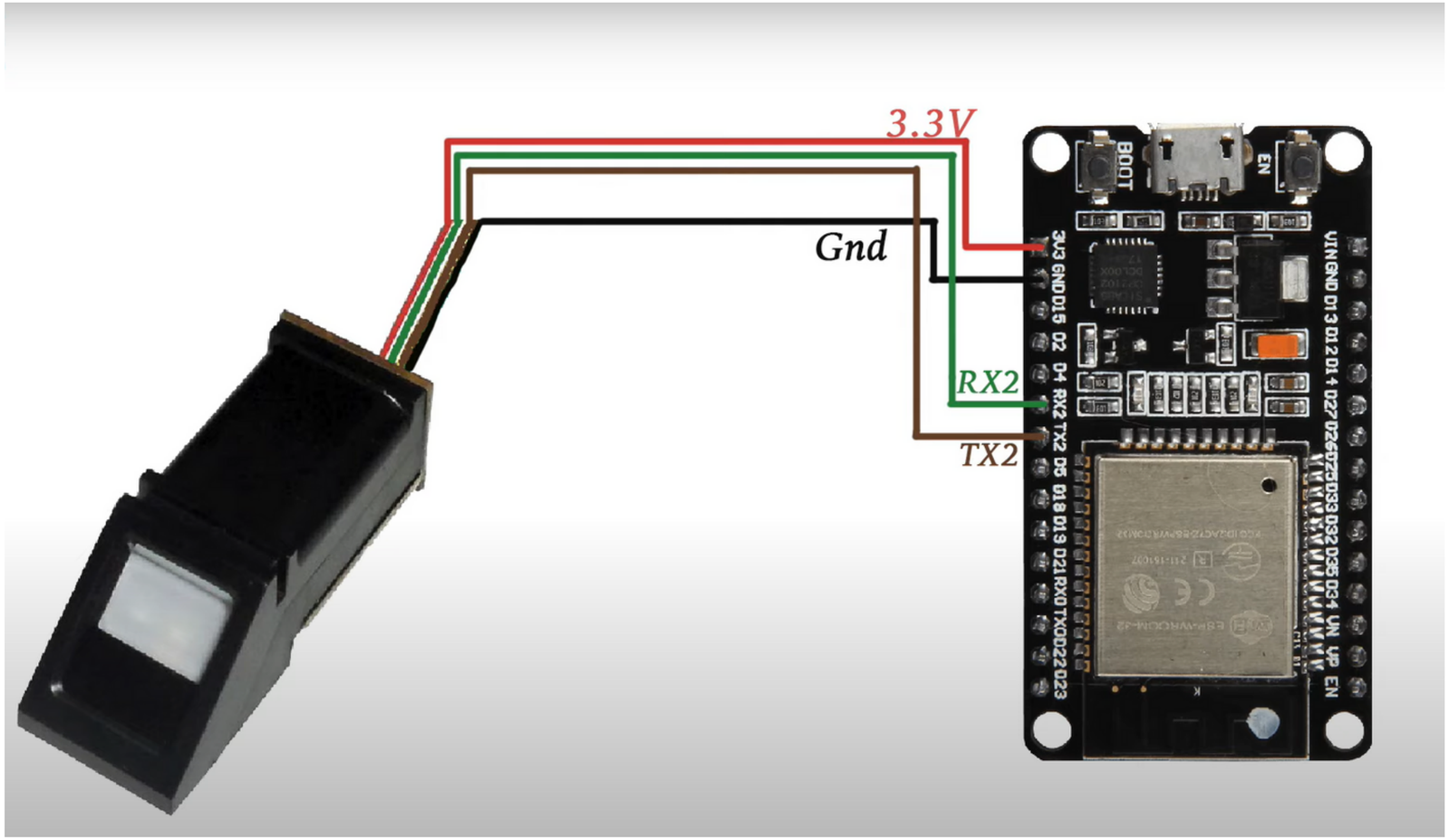
Abstraction: The code uses various libraries and functions that abstract low-level details, providing higher-level functionalities. For example, the cv2 library abstracts image processing operations.

File Handling: The code uses file handling in an object-oriented manner. It interacts with the file "Attendance.csv" to read and write attendance data.

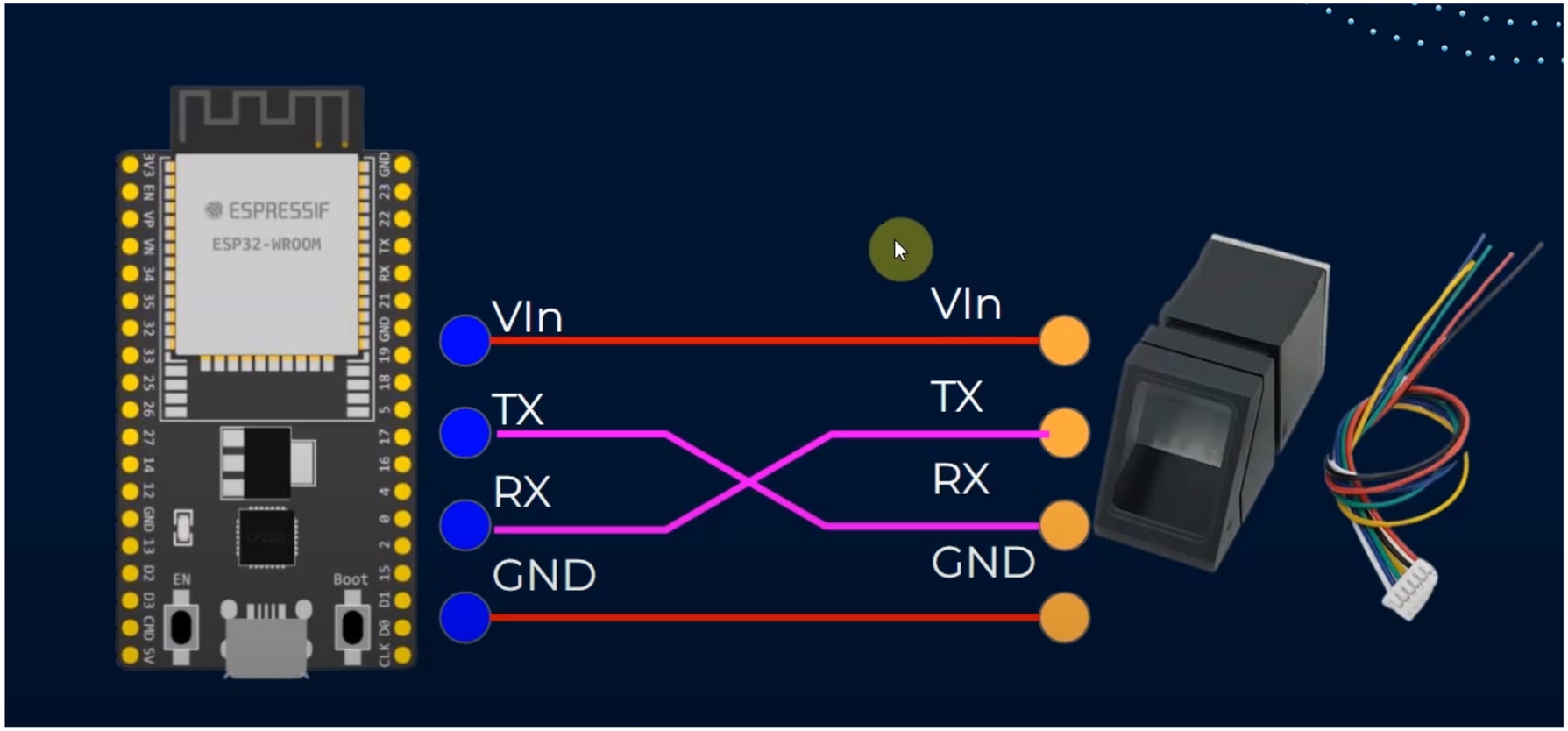
Data Structures (List): The code uses lists (images, classNames, encodeListKnown, myDataList) to store and manipulate data.

DateTime Object: The datetime class from the datetime module is used to create a datetime object (now) for timestamping attendance entries.

Fingerprint Sensor



Fingerprint Sensor



Libraries Used

- asdafruit fingerprint
- fingerprint.h
- wire.h
- wifi.h
- lcd.h
- esp system lib

Website

The data received form cam sensor and fingerprint sensor is sent to the website server via IFTTT api using google sheets

Backend

- Authorization Using Login Credentials
- Stores Email and DATA
- Verifies Login through 2FA
- Use of tokens and cookies for Enhance security

Application

- Working on flutter app proved cross platform support
- Easy Access of Attendance Data
- Quick Render
- Ease In biometric upload
- Safe Storage of Data using fire base

Usage of OOPs

- Encapsulation by bundling code into widgets
- Inheritance
- Polymorphism through build method
- Abstraction

Error Faced During Deployment

Error Due to Dlib
Error in Creating Wheels
Error Due To Mismatched Pyhton Version i.e. Python 3.8.10
Error in Connection of Hardware (Burnout Error)
Server Error