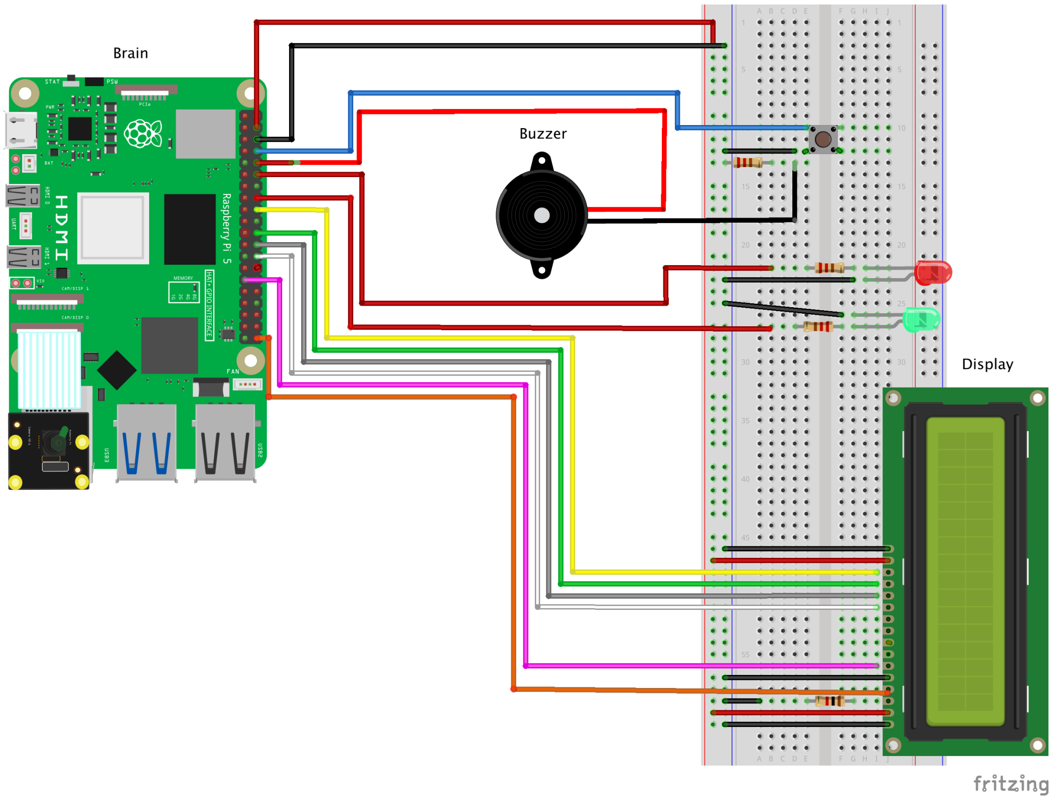
Design

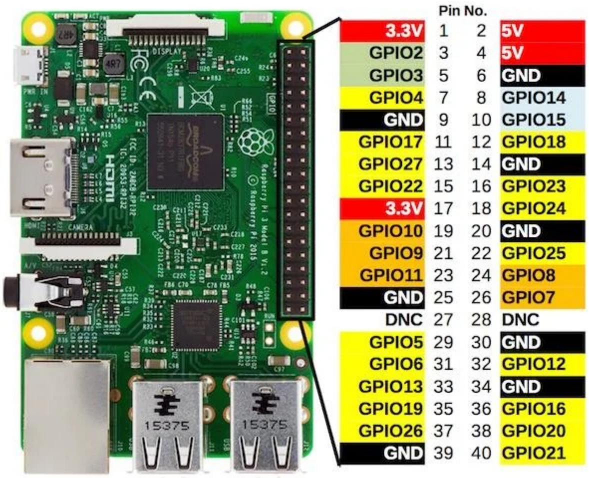
Raspberry setup



About Hardware:

The Circuit would consist off:

- Raspberry Pi 5 :
    1. This would work as the brain of the circuit where every hardware part would contain as a body part.
    2. Raspberry pi would store the instruction which would be taken to the parts to operate



	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Raspberry pi 4B similar GPIO pin as Raspberry pi 5

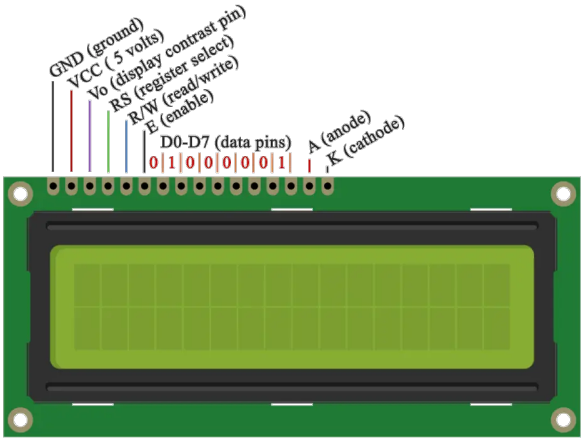
Image from: https://www.reddit.com/media?url=https%3A%2F%2Fpreview.redd.it%2Fpowering-raspberry-pi-5-with-gpio-v0-9sk0kmbr6cac1.jpg%3Fwidth%3D600%26format%3Dpjpg%26auto%3Dwebp%26s%3Dbba66153b2b9ec5128240bd86f81fe4f5450caae

- Pi Camera Module 3 : The Camera is a 120 degree wide angled camera which would work perfect to detect objects and people around

- Passive Buzzer: This component would work as an alarm which would only go off if the camera using face recognition algorithm detected someone who is suspicious person. The buzzer can be turned off:
  1. If the user or user-related people come and push the pushbutton
  2. Can be automatically turned off after 2 min

- LEDs: This would be the indicator of people which shows that if a familiar face stored or learned by the algorithm (This will be indicated by Green LED) and Unfamiliar face or Suspicious faces (this would be indicated by Red LED)

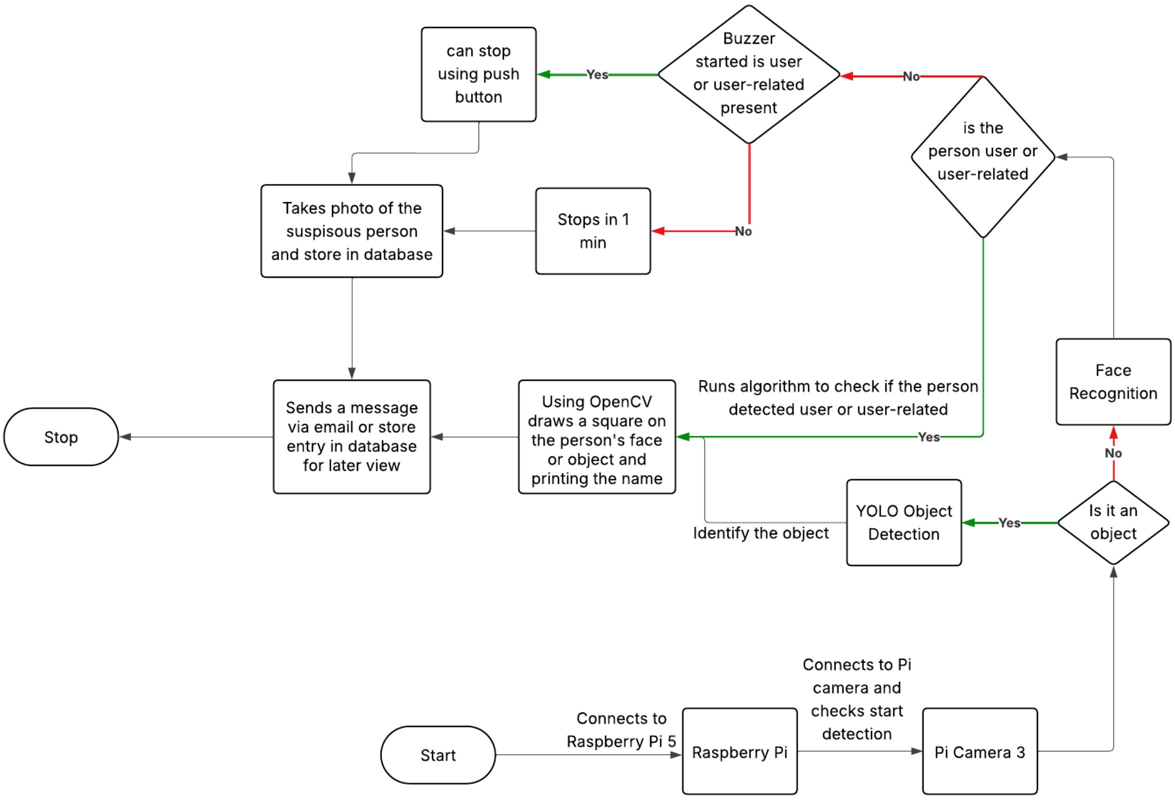
- LCD 16 pin Display: The LCD would work as a Display which would print the name of the user or user-related as stored in algorithms



LCD 16 pin Display

Image From: https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/

Algorithm Design


can stop using push button
Yes
Buzzer started is user or user-related present
No
is the person user or user-related
Takes photo of the suspisous person and store in database
Stops in 1 min
No
Face Recognition
Runs algorithm to check if the person detected user or user-related
Yes
Stop
Sends a message via email or store entry in database for later view
Using OpenCV draws a square on the person's face or object and printing the name
No
Is it an object
Yes
YOLO Object Detection
Identify the object
Connects to Pi camera and checks start detection
Connects to Raspberry Pi 5
Start
Raspberry Pi
Pi Camera 3

The program would first start with the pi camera v3 (module 3) turning on with raspberry pi 5 and connect to it. The camera would search the full range of its vision to see if there is any object or person which could be detected. After detection there are only three things that can happen:

1. Object Detected by the YOLO object detection algorithm

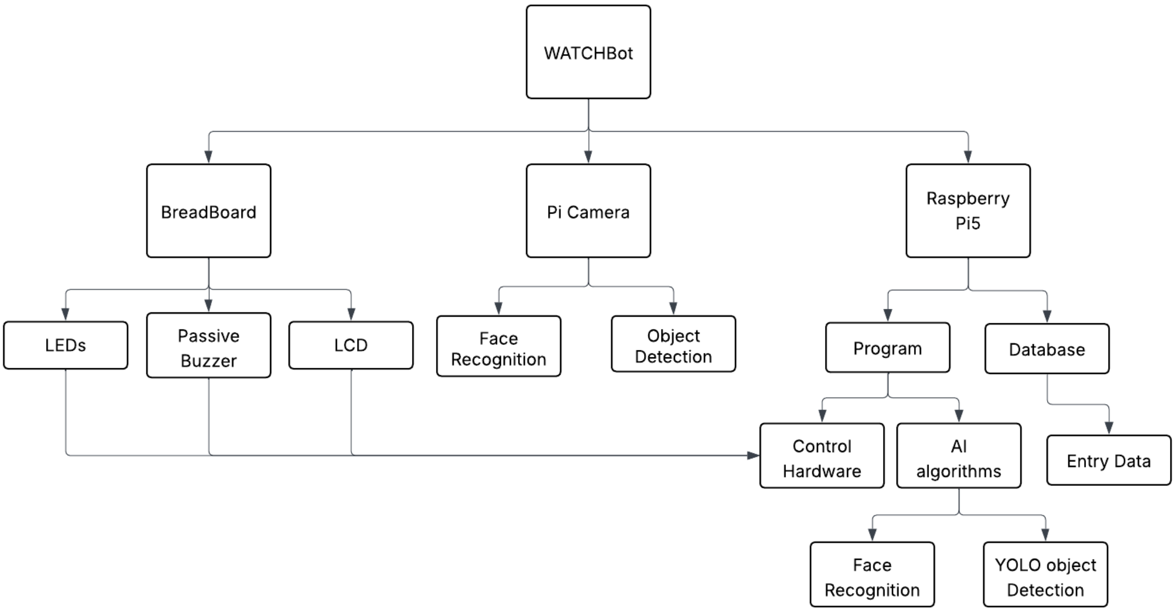
2. Person Detected by the face recognition algorithm by FaceNet(Google), from this can could be either two scenario where:
    1. Person Detected could be user or user related which would indicate it by the turning on of Green LED
    2. Person Detected is unknown: this where there could be chances where the person detected can be be either be unknown or suspicious person (this could be detected using the Red LED turning on)

3. There many be chances where there are people or object in the camera's range.

The following pseudocode describes the logical flow of how WATCHBot detects and identifies people, using the Pi Camera and facial recognition algorithms to control hardware outputs:

```
BEGIN PROGRAM

INITIALISE camera
INITIALISE database of known faces
INITIALISE trained model for suspicious detection
INITIALISE LEDs, buzzer, LCD display

LOOP forever
    CAPTURE frame from camera
    DETECT objects in frame using YOLO algorithm

    IF object detected IS person THEN
        IDENTIFY face using facial recognition

        IF face matches known user THEN
            DISPLAY user name on LCD
            TURN ON green LED
            TURN OFF red LED
            TURN OFF buzzer

        ELSE
            CHECK if person matches suspicious pattern (e.g., mask
detected)

            IF person IS suspicious THEN
                DISPLAY "Suspicious Person!" on LCD
                TURN ON red LED
                ACTIVATE buzzer
                LOG event in database as "Suspicious"
                WAIT 2 minutes
                TURN OFF buzzer

            ELSE
                DISPLAY "Unknown Person" on LCD
                TURN ON red LED
                LOG event in database as "Unknown"
            ENDIF
        ENDIF
    ENDIF
ENDLOOP

END PROGRAM
```

Structure/Hierarchy Diagrams



The structure and hierarchy diagram represents the overall architecture of WATCHBot, showing how the hardware and software components interact to perform detection and response tasks.

At the top level, WATCHBot is composed of three main parts — the Breadboard, Pi Camera, and Raspberry Pi 5 — each responsible for a specific set of functions.

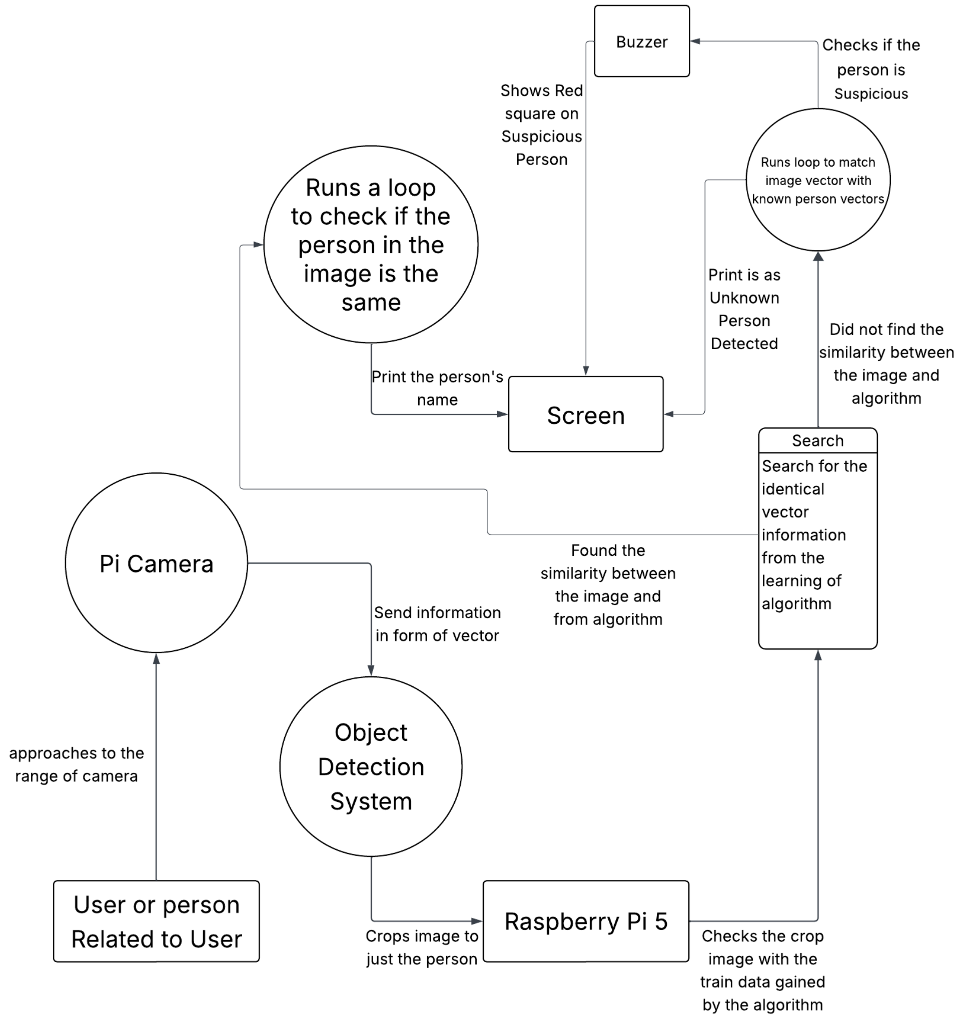
The Breadboard handles the output hardware, including the LEDs, passive buzzer, and LCD display, which provide visual and auditory feedback during detections. These components are directly controlled by the Raspberry Pi to indicate different system states such as suspicious detection or normal operation.

The Pi Camera serves as the system's visual input device, capturing real-time footage and passing it to the Object Detection and Face Recognition modules. These modules work together to identify and classify people or objects within the camera's range.

The Raspberry Pi 5 acts as the core processing unit, managing both the program logic and the database. The program contains the control instructions for hardware activation and integrates AI algorithms such as YOLO Object Detection and Facial Recognition. The database component stores entry data, including detection results and timing information, enabling record-keeping and analysis.

This hierarchical structure ensures smooth coordination between hardware and software, allowing WATCHBot to detect, identify, and respond efficiently to different scenarios in real time.

Data Flow Diagram



Buzzer
Checks if the person is Suspicious
Shows Red square on Suspicious Person
Runs loop to match image vector with known person vectors
Runs a loop to check if the person in the image is the same
Print is as Unknown Person Detected
Did not find the similarity between the image and algorithm
Print the person's name
Screen
Search
Search for the identical vector information from the learning of algorithm
Found the similarity between the image and from algorithm
Pi Camera
Send information in form of vector
Object Detection System
approaches to the range of camera
User or person Related to User
Crops image to just the person
Raspberry Pi 5
Checks the crop image with the train data gained by the algorithm

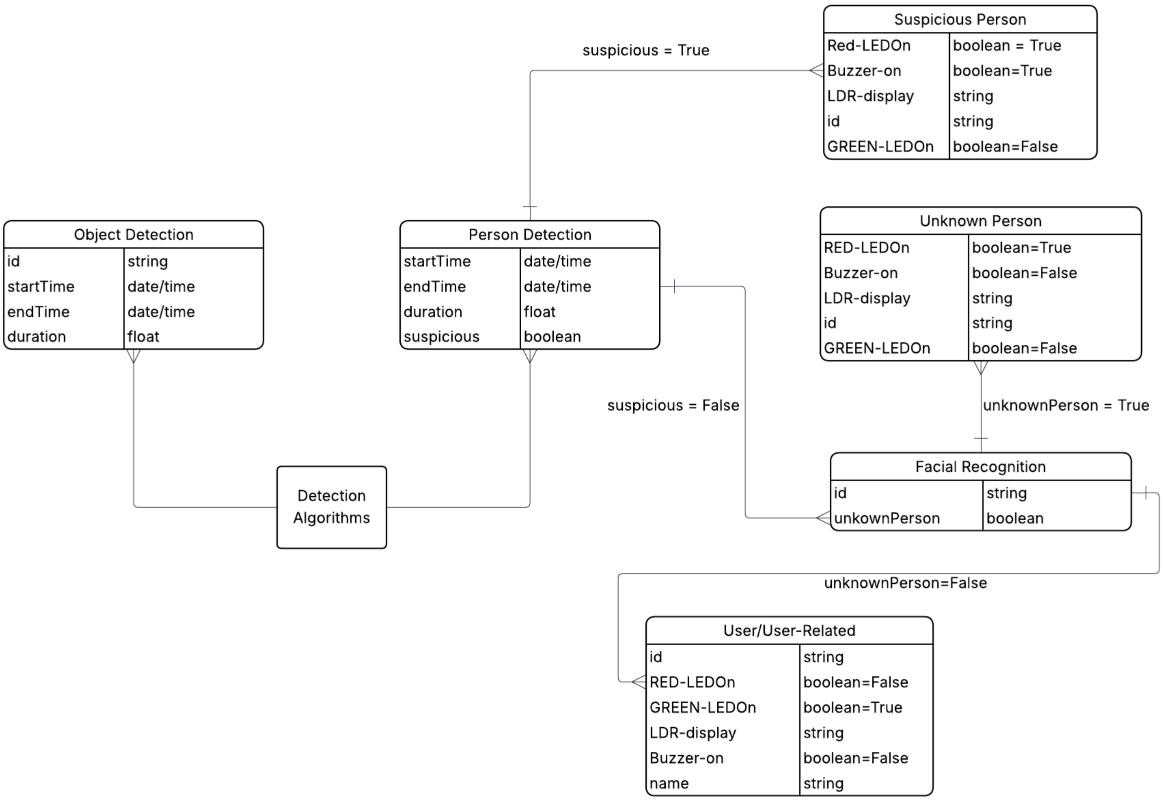
The Data Flow Diagram (DFD) illustrates how information moves through the detection system, starting from the Pi Camera and ending with the output on the screen and buzzer.

When a user or any person enters the range of the camera, the Object Detection System identifies the presence of a person and crops the image to focus only on them. This cropped image is then processed by the Raspberry Pi 5, which converts it into a vector form and checks it against the stored training data of the algorithm.

If a match is found between the new image and an existing known vector, the system prints the person's name on the screen, identifying them as a known user. If no match is found, the algorithm labels the person as Unknown and displays an alert message. In case the system detects a suspicious person, it activates the buzzer and displays a red square on the screen to highlight the detection.

The system continuously runs a loop to verify whether the person remains the same or if a new face has entered the frame. This real-time flow of data between the camera, object detection module, and Raspberry Pi ensures quick and accurate facial recognition and response through both visual (screen) and auditory (buzzer) alerts.

Entity Relationship Diagram



The Entity Relationship Diagram (ERD) illustrates the database structure used in the detection system. It consists of two main detection types — Object Detection and Person Detection — which are both managed by the central Detection Algorithms entity.

In the case of Person Detection, the system determines whether a person is suspicious or not suspicious. If the individual is marked as suspicious, details such as LED and buzzer status are stored in the *Suspicious Person* table.

If the person is not suspicious, the record is passed to the Facial Recognition module, which further classifies the person as either Unknown or User/User-Related. Each category maintains information such as LED activation, buzzer status, and LDR display data, which represent the hardware's real-time response to the detection.

This relational design allows the system to efficiently log, categorise, and respond to detections while maintaining a clear separation between object detection and facial recognition processes.

Database Design

1. Object Detection

Field Name	**Data Type**	**Description**	**Constraints**
id	STRING	Unique ID of the detection	PRIMARY KEY
startTime	DATETIME	Time detection started	NOT NULL
endTime	DATETIME	Time detection ended	NOT NULL
duration	FLOAT	Duration of detection (in seconds/ minutes)	

2. Person Detection

Field Name	Data Type	Description	Constraints
id	STRING	Unique ID	PRIMARY KEY
startTime	DATETIME	Detection start time	NOT NULL
endTime	DATETIME	Detection end time	NOT NULL
duration	FLOAT	Detection duration	
suspicious	BOOLEAN	True if person is suspicious	NOT NULL

3. Suspicious Person

Field Name	Data Type	Description	Constraints
id	STRING	Foreign key from PersonDetection	PRIMARY KEY, FOREIGN KEY (id) REFERENCES PersonDetection(id)
RED_LEDOn	BOOLEAN	True when red LED is ON	DEFAULT TRUE
GREEN_LEDOn	BOOLEAN	True when green LED is ON	DEFAULT FALSE
LDR_display	VARCHAR(100)	Sensor reading for display	
Buzzer_on	BOOLEAN	True when buzzer is ON	DEFAULT TRUE

4. Unknown Person

Field Name	Data Type	Description	Constraints
id	STRING	Foreign key from FacialRecognition	PRIMARY KEY, FOREIGN KEY (id) REFERENCES FacialRecognition(id)
RED_LEDOn	BOOLEAN	True when red LED is ON	DEFAULT TRUE
GREEN_LEDOn	BOOLEAN	True when green LED is ON	DEFAULT FALSE
LDR_display	VARCHAR(100)	Display data for LDR	
Buzzer_on	BOOLEAN	True when buzzer is ON	DEFAULT FALSE

5. Facial Recognition

Field Name	Data Type	Description	Constraints
id	STRING	Unique ID	PRIMARY KEY
unknownPerson	BOOLEAN	True if unknown person detected	NOT NULL

6. User/User-Related

Field Name	Data Type	Description	Constraints
id	STRING	Unique ID	PRIMARY KEY
name	VARCHAR(100)	Name of the known user	
RED_LEDOn	BOOLEAN	True when red LED is ON	DEFAULT FALSE
GREEN_LEDOn	BOOLEAN	True when green LED is ON	DEFAULT TRUE
LDR_display	VARCHAR(100)	Display sensor info	
Buzzer_on	BOOLEAN	True when buzzer is ON	DEFAULT FALSE

7. Detection Algorithms

Field Name	Data Type	Description	Constraints
id	STRING	Unique ID	PRIMARY KEY
objectDetection_id	INT	Foreign key to ObjectDetection	FOREIGN KEY (objectDetection_id) REFERENCES ObjectDetection(id)
personDetection_id	INT	Foreign key to PersonDetection	FOREIGN KEY (personDetection_id) REFERENCES PersonDetection(id)