

# 3.1 Requirement Analysis Document — PiGuard

## 1. Project Summary

**PiGuard** is an AI-powered smart security prototype designed to enhance safety in environments such as homes or offices. The system runs on a **Raspberry Pi 4** equipped with a **camera module** that captures real-time video, detects, and recognizes faces using **computer vision** algorithms.

If a detected face is not recognized, it is automatically added to a “**strangers**” table in the database. A **Flutter mobile application** allows the user to view unknown individuals, manage registered users, and receive real-time notifications when a stranger is detected.

PiGuard integrates **AI, embedded systems, and mobile development technologies** to provide a **low-cost, intelligent, and efficient security solution**.

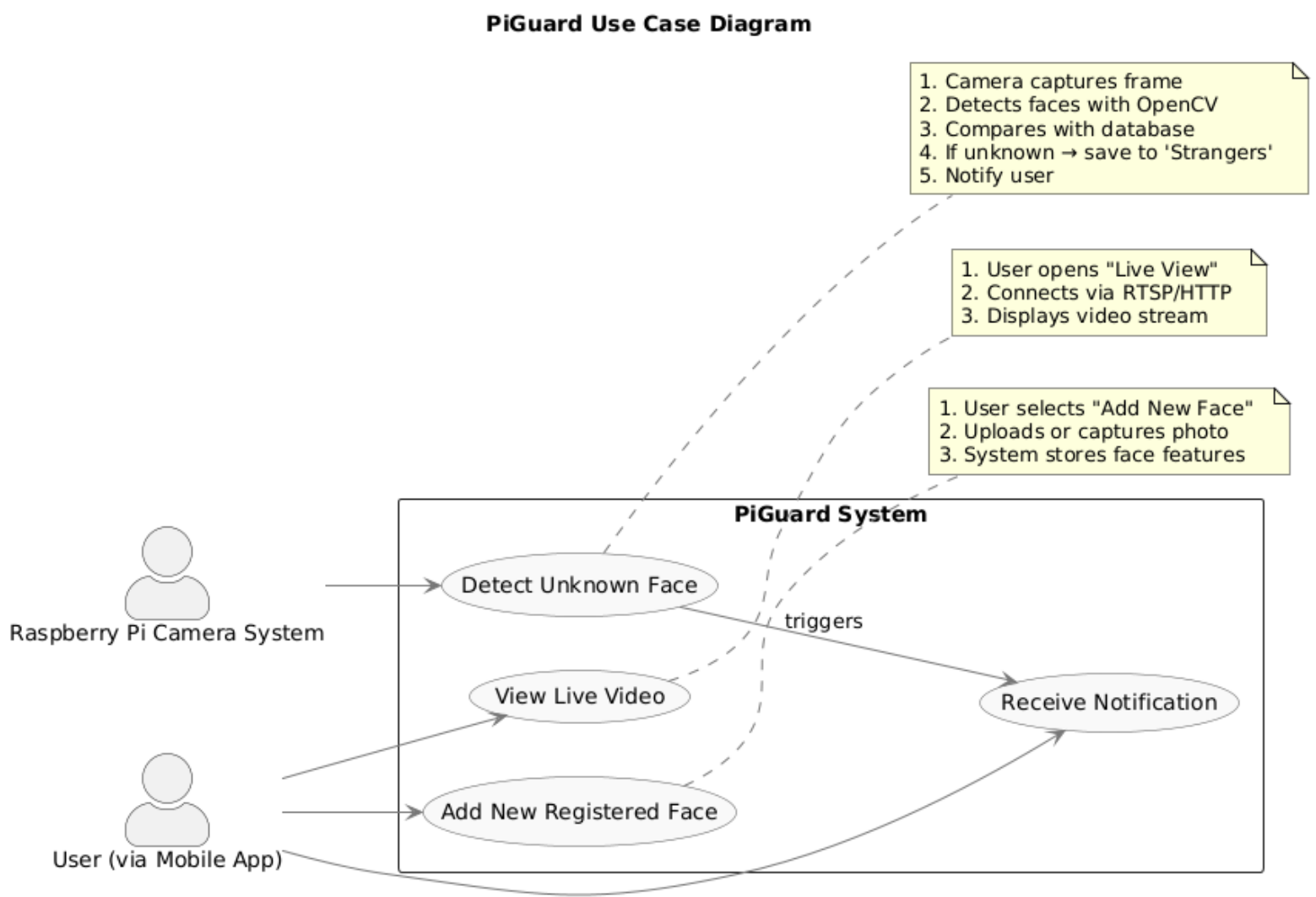
## 2. Functional Requirements

ID	Requirement	Description
F1	Face Detection	The system detects faces in the live camera feed using OpenCV.
F2	Face Recognition	Detected faces are compared with registered users in the database.
F3	Stranger Recording	Unrecognized faces are saved automatically in the “strangers” table.
F4	Stranger List Display	The mobile app displays a list of all detected strangers.
F5	Real-Time Notification	Sends a push notification to the user when a stranger is detected.
F6	Manage Registered Faces	The user can add or delete known faces via the mobile application.
F7	Live Camera View	The user can access a live camera feed through the mobile app.
F8	Alert System	When an unknown face is detected, an alert sound or visual signal is triggered.

### 3. Non-Functional Requirements

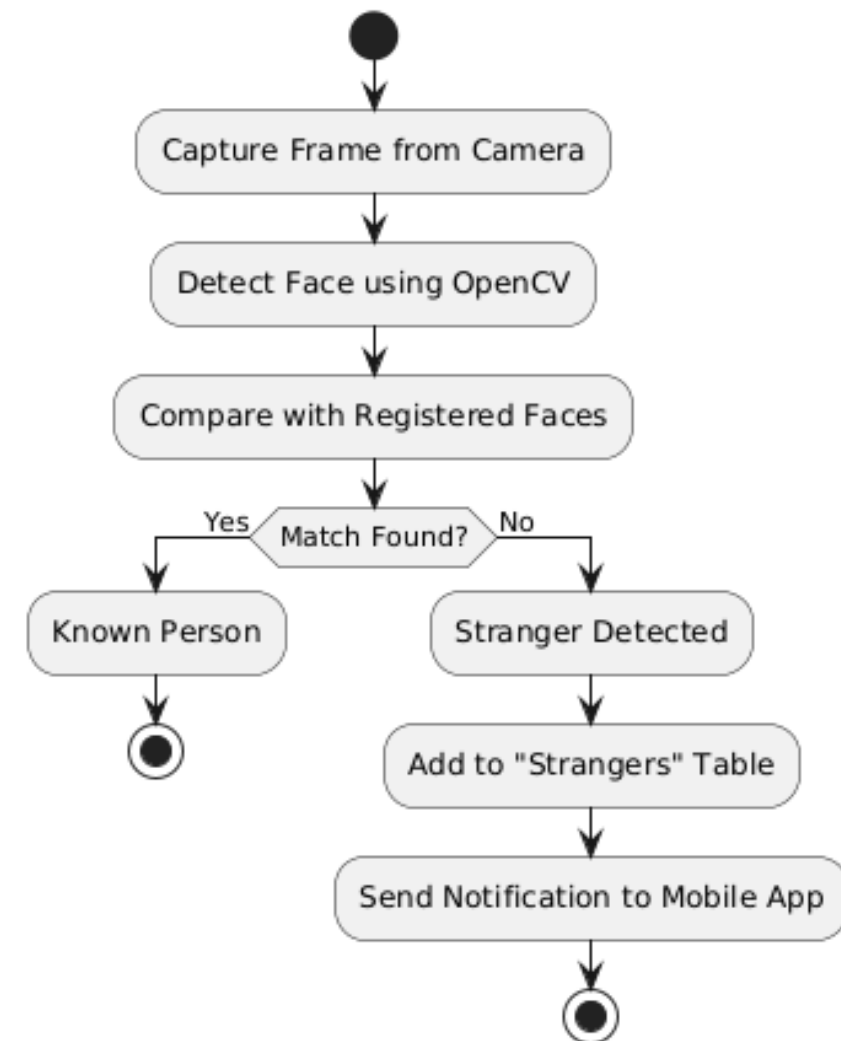
ID	Requirement	Description
NF1	Performance	The system should detect and recognize faces in less than 1 second per frame.
NF2	Security	Database and API access must be restricted with authentication.
NF3	Portability	The system must run reliably on Raspberry Pi 4 hardware.
NF4	Usability	The mobile application interface must be simple and user-friendly.
NF5	Scalability	Adding new faces should not degrade system performance.
NF6	Reliability	The system should automatically restart the camera module if it disconnects.
NF7	Compatibility	Python code runs on Raspberry Pi OS; Flutter app supports both Android and iOS.

### 4. User Scenarios (Use Cases)

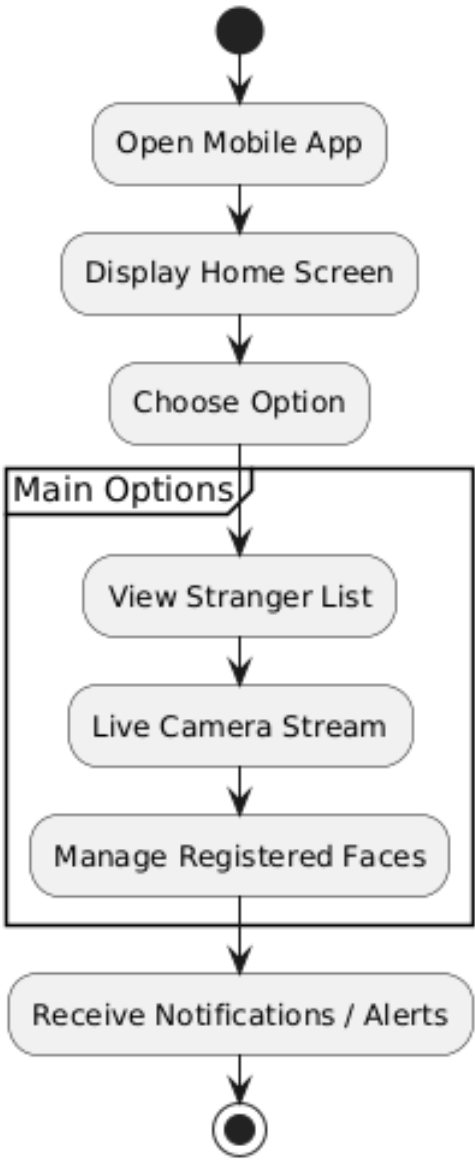


## 5. Workflow Diagrams

### 5.1 Face Detection & Recognition Workflow



5.2 Mobile Application Workflow



6. Conclusion

This **Requirements Analysis Document** defines the core and extended behaviors of the **PiGuard** system. It ensures the software meets real-time performance, security, and reliability goals while maintaining usability and scalability. The combination of **Python + OpenCV + Flutter + Raspberry Pi** provides a cost-effective and intelligent prototype for small-scale smart surveillance systems.