SMART CITY PROJECT

****

A Lab Project Report

# Software requirement specifications (SRS)

Under the supervision of

# Prof. Mohammad Sarosh Umar

&

# Mr. Misbahur Rahman Warsi

Submitted by: Team of

M-Tech Computer Engineering 2024-25

# Introduction

The Smart City Project aims to enhance urban security, surveillance, and access control through four integrated modules:

1. **Automatic Number Plate Recognition (ANPR)**
2. **Surveillance System Using Raspberry Pi**
3. **Face-Recognition-Based Smart Entry System**
4. **Centralized Dashboard for Module Management**

Each module contributes uniquely to creating a secure and efficient smart city infrastructure, with the centralized dashboard providing seamless control, monitoring, and data integration across all modules.

# Automatic Number Plate Recognition (ANPR)

### Introduction

* + 1. Purpose

The purpose of this document is to provide a comprehensive description of the software

requirements for the development of an Automatic Number Plate Recognition (ANPR) system.

This document will outline the functionalities, constraints, and specifications necessary to design

and implement a robust ANPR system.

### Scope

The ANPR system aims to automatically recognize and interpret license plate information from

images and video streams. It will be utilized for various applications, including law enforcement,

parking management, and traffic monitoring.

### Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, and documentation

writers. It provides an overview of the ANPR system's requirements, organized to cater to each

reader type.

### System Overview

* + 1. System Description

The ANPR system will capture images or video frames containing vehicle license plates,

process the visual data, and extract relevant information such as license plate number.

### Key Features

* + - 1. License Plate Recognition: Accurate extraction and recognition of license plate numbers.
      2. Database Integration: Storing and retrieving recognized license plate information from a database.
      3. Alerts and Notifications: Generating alerts or notifications for specific events or matches.

### Functional Requirements

* + 1. Image/Video Input
       1. The system should accept images or video streams containing vehicles with visible license plates.
       2. Images may be captured by surveillance cameras, CCTV systems, or other image sources.

### License Plate Recognition

* + - 1. Implement an algorithm for accurate license plate recognition.
      2. Recognize alphanumeric characters from various plate formats.

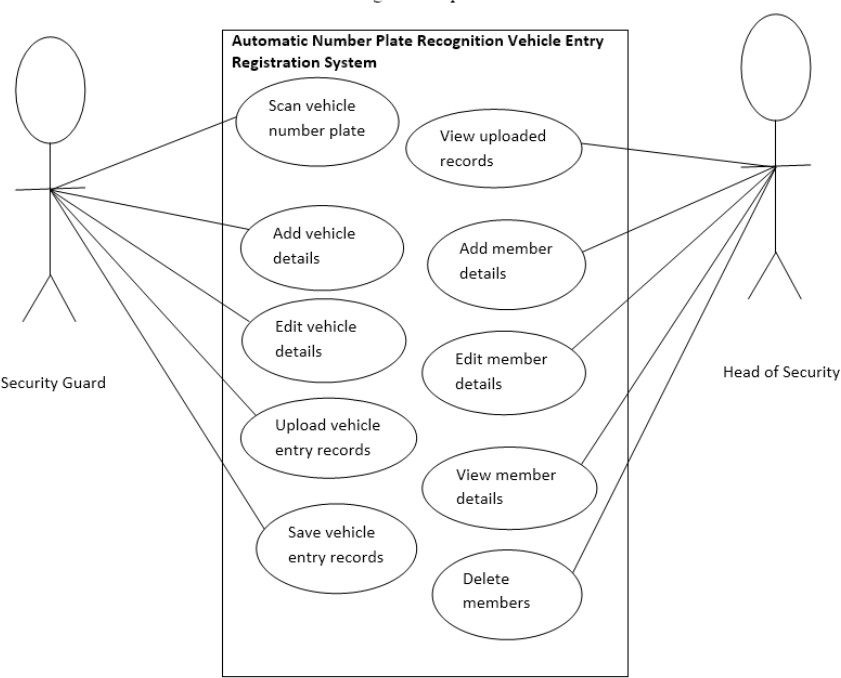
### Database Integration

* + - 1. Create and maintain a database to store recognized license plate information.
      2. Enable efficient retrieval of information based on queries.

### Non-functional Requirements

* + 1. Performance
       1. The system should process images in real-time, with minimal latency.
       2. Support recognition in various lighting and weather conditions.
    2. Reliability
       1. Ensure a high level of accuracy in license plate recognition.
       2. Implement error-handling mechanisms for exceptional cases.

### System Architecture



* 1. Interfaces

#### API

* + - 1. Define an API for integration with dashboard.

### Constraints

* + 1. The system's accuracy may be affected by factors such as image quality, lighting conditions, and plate variations.
    2. Legal and privacy considerations must be taken into account when implementing the system.

# Face-Recognition-Based Smart Entry System

### Introduction

#### Purpose

The purpose of this project is to enhance security and streamline access control by implementing a face-recognition-based smart entry system. The system aims to replace

traditional access methods like keycards and passwords with AI-driven facial authentication, ensuring quick and secure entry for authorized individuals while preventing unauthorized access.

#### Scope

This system will be implemented in organizations, residential complexes, government offices, and high-security zones to allow access only to authorized individuals.

### Definitions, Acronyms, and Abbreviations

* + 1. FRBSES: Face-Recognition-Based Smart Entry System
    2. AI: Artificial Intelligence
    3. GUI: Graphical User Interface
    4. DBMS: Database Management System

### Overall Description

#### Product Perspective

* + 1. The Face-Recognition-Based Smart Entry System will replace traditional authentication methods with facial recognition to enhance security and user convenience.

#### Product Functions

* + 1. Capture and process live facial images.
    2. Compare captured images with stored facial templates.
    3. Grant or deny access based on authentication results.
    4. Maintain a log of all entry attempts.
    5. Notify administrators in case of unauthorized attempts.

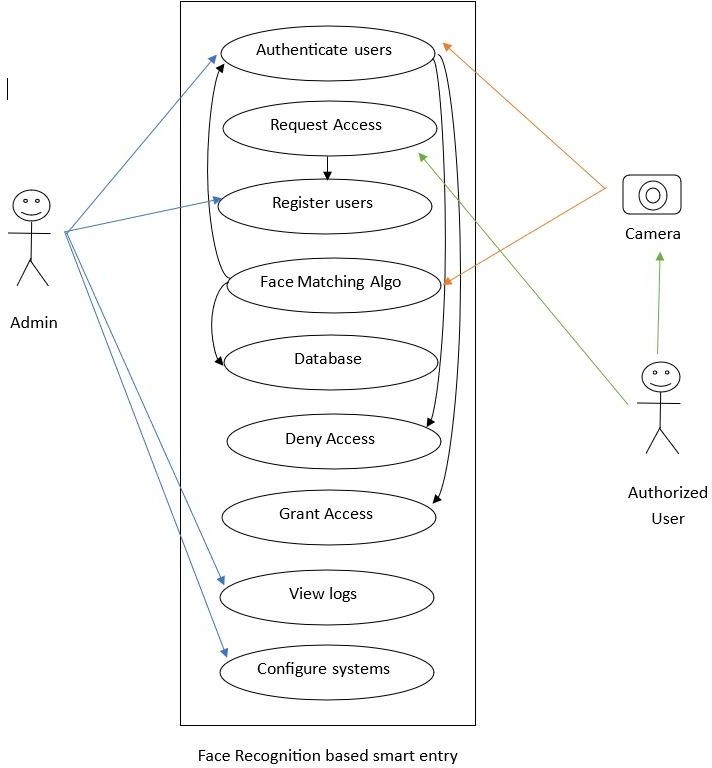
### Specific Requirements

#### Functional Requirements

* + 1. User Registration
       1. The system shall allow administrators to register new users by capturing and storing their facial data.
       2. Users shall be able to update or delete their facial data upon request.

### Face Recognition Authentication

* + - 1. The system shall capture and process the user's face for authentication.
      2. It shall compare the captured image with stored templates and grant access if a match is found.



**Figure 1 Use Case Diagram**

### Conclusion

* 1. This document outlines the requirements for a Face-Recognition-Based Smart Entry System. Future improvements may include voice recognition and cloud-based authentication solutions.

# Surveillance System Using Raspberry Pi for A Smart City

### Introduction

#### Purpose

This project aims to develop a real-time surveillance system using Raspberry Pi with storage capacity and computer vision to enhance security in smart cities.

#### Scope

-It will capture and stream live video from Raspberry Pi cameras.

-It will detect motion and trigger recording.

-It will send alerts to authorities for unusual behaviors.

### System Overview

#### Components

* + 1. Raspberry Pi, camera module, memory card, motion sensors.

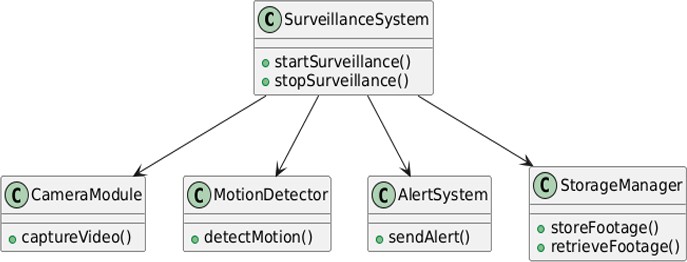
#### Key Features

* + 1. Live streaming at dashboard.
    2. Motion detection and automatic recording.
    3. Instant alert sends to authorities.

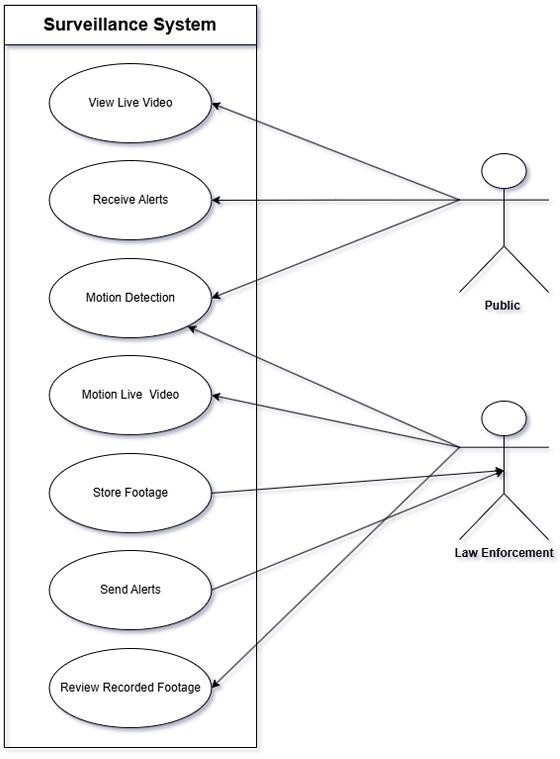
### System Requirements

* 1. Functional Requirements
     1. Capture and stream live videos from Raspberry Pi camera module.
     2. Detect motion and trigger automatic recording.
     3. Store footage in a memory card.

### UML Diagrams



**Figure 2 Class Diagram**



**Figure 3 Use Case Diagram**

# Centralized Dashboard for Module Management

Purpose: To provide a unified platform for monitoring, controlling, and analyzing data

### Scope:

* Centralized Monitoring of ANPR, Surveillance, and Face Recognition Systems
* Real-Time Alerts and Notifications
* Advanced Data Analytics and Reporting
* User Role Management and Access Control

## System Architecture

The Smart City Project integrates all modules through a centralized dashboard, enabling seamless monitoring, data sharing, and real-time alerts. The architecture is designed for scalability, high availability, and secure data transmission.

Components:

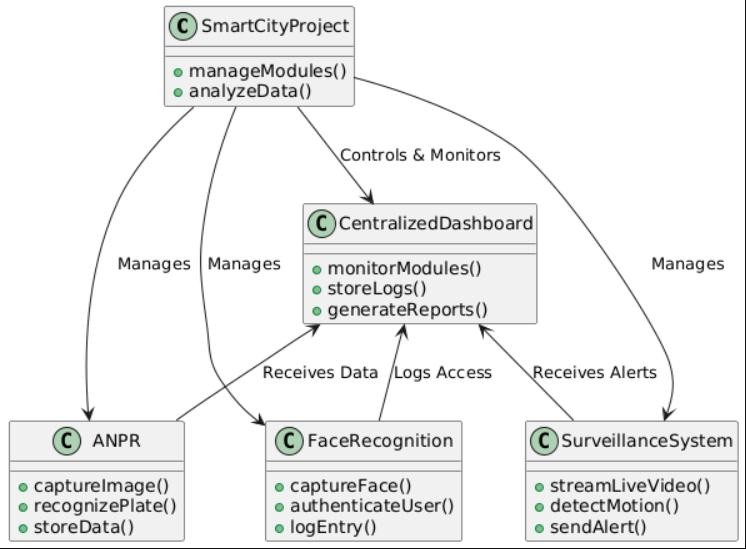
* + ANPR Cameras and License Plate Database
  + Raspberry Pi Surveillance Nodes
  + Face Recognition Access Points
  + Central Monitoring Dashboard
  + Cloud Infrastructure for Data Backup and Analytics

1. ***Technology Stack:***

* **Backend:** Python with FastAPI for high-performance API development.
* **Frontend:** React.js or Vue.js for an interactive and responsive dashboard.
* **Database:** PostgreSQL or MongoDB for efficient data storage.
* **Cloud Infrastructure:** AWS, Google Cloud, or Azure for data backup and analytics.

#### **Security:** OAuth 2.0 and JWT for user authentication and access control.

* **IoT Integration:** MQTT or WebSockets for real-time communication with surveillance nodes.



**Figure 4 Class Diagram for smart city project**

## 4. Conclusion

The Smart City Project leverages **surveillance, ANPR-based vehicle tracking, and biometric access control** to enhance urban security. The centralized dashboard provides **real-time monitoring, predictive analytics, and automated emergency responses.**

With a **scalable and modular architecture**, future enhancements could include **IoT integrations, AI-based anomaly detection, and automated security interventions**, making urban environments safer and more intelligent.