**Software Requirements Specification**

**For**

**Campus Permit Parking System**

Version 1.0

Faculty Advisor: Dr. Truong Tran

Ryan Brennan

Tyler Wallace

Brendan Gaffney

Jeffrey Tetkoskie

Alan Baxley

Table of Contents

[1. Introduction 3](#_Toc153527947)

[1.1 Purpose 3](#_Toc153527948)

[1.2 Document Conventions 3](#_Toc153527949)

[1.3 Intended Audience and Reading Suggestions 3](#_Toc153527950)

[1.4 Project Scope 3](#_Toc153527951)

[2. Overall Description 3](#_Toc153527952)

[2.1 Product Perspective 3](#_Toc153527953)

[2.2 Product Features 3](#_Toc153527954)

[2.3 User Classes and Characteristics 3](#_Toc153527955)

[2.4 Operating Environment 4](#_Toc153527956)

[2.5 Design and Implementation Constraints 4](#_Toc153527957)

[3. System Features 4](#_Toc153527958)

[3.1 License Plate Recognition 4](#_Toc153527959)

[3.1.1 Functional Requirements 4](#_Toc153527960)

[3.2 Vehicle Registration Management 4](#_Toc153527961)

[3.2.1 Functional Requirements 4](#_Toc153527962)

[3.3 Database Interaction 4](#_Toc153527963)

[3.3.1 Functional Requirements 4](#_Toc153527964)

[4. External Interface Requirements 5](#_Toc153527965)

[4.1 User Interfaces 5](#_Toc153527966)

[4.2 Hardware Interfaces 5](#_Toc153527967)

[4.3 Software Interfaces 5](#_Toc153527968)

[4.4 Communications Interfaces 5](#_Toc153527969)

[5. Other Nonfunctional Requirements 5](#_Toc153527970)

[5.1 Performance Requirements 5](#_Toc153527971)

[5.2 Safety Requirements 5](#_Toc153527972)

[5.3 Security Requirements 5](#_Toc153527973)

[5.4 Software Quality Attributes 5](#_Toc153527974)

[6. Other Requirements 5](#_Toc153527975)

[Appendix A: Glossary 6](#_Toc153527976)

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of the Campus Parking Permit System (CPPS). It will explain the system’s functionality, the interfaces between the system and the users, and the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system.

## 1.2 Document Conventions

This SRS follows the IEEE-830 standard for SRS formatting.

## 1.3 Intended Audience and Reading Suggestions

The primary audience for this document is the campus parking services staff, students, faculty, and the development team. The document is organized by system functionality and priority.

## 1.4 Project Scope

The CPPS is designed to manage vehicle registration and automate parking enforcement on a college campus. The system will replace physical parking permits with a digital license plate recognition system, reducing costs and improving enforcement efficiency.

# 2. Overall Description

## 2.1 Product Perspective

The CPPS is an independent system but will interface with the college’s existing student and faculty databases for authentication and registration information.

## 2.2 Product Features

Automatic license plate recognition and retrieval of vehicle details from a database.

Web portal for vehicle registration by students and faculty.

Database management for registered and unregistered vehicles.

Support for multiple vehicles per user with variable pricing.

## 2.3 User Classes and Characteristics

Parking Services Staff: Need to verify registration status and enforce parking regulations.

Students/Faculty: Need to register vehicles and ensure their vehicles are recognized as permitted.

## 2.4 Operating Environment

The CPPS will run on Android devices and a web-based version will be developed using Django/Python. The system will use Microsoft SQL Server for database management.

## 2.5 Design and Implementation Constraints

The system must operate with high accuracy in various lighting and weather conditions.

User interfaces must be accessible and compliant with ADA standards.

The system must comply with data protection laws for handling personal information.

# 3. System Features

## 3.1 License Plate Recognition

Description: The system must automatically recognize and decode license plate information using the device’s camera.

## 3.1.1 Functional Requirements

The system shall capture license plate images using CameraX API.

The system shall process images to retrieve license plate characters using ML Kit.

The system shall cross-reference license plate information with the vehicle registration database.

## 3.2 Vehicle Registration Management

Description: The system provides a web interface for vehicle registration.

### 3.2.1 Functional Requirements

The system shall allow students and faculty to register multiple vehicles.

The system shall provide tiered pricing options for multiple vehicle registrations.

The system shall store unregistered vehicle data for a specified period.

## 3.3 Database Interaction

Description: The system interacts with the database to retrieve and store vehicle data.

### 3.3.1 Functional Requirements

The system shall retrieve vehicle make, model, and registration status upon query.

The system shall update the registration status of the vehicle based on user input.

The system shall maintain historical data for unregistered vehicles.

# 4. External Interface Requirements

## 4.1 User Interfaces

Mobile and web-based interfaces for registration and enforcement.

Real-time display of recognized license plate information for enforcement officers.

## 4.2 Hardware Interfaces

Compatibility with Android device cameras for mobile app.

Support for LPR cameras for future integration.

## 4.3 Software Interfaces

Database: Microsoft SQL Server.

Mobile App: Android Studio with Kotlin.

Web Version: Django with Python.

## 4.4 Communications Interfaces

RESTful API for database interaction.

Secure HTTPS connections for web-based interactions.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The system shall process license plate information within 2 seconds of image capture.

The database shall handle simultaneous queries from multiple users.

## 5.2 Safety Requirements

The system shall not store images of license plates longer than necessary to prevent misuse.

## 5.3 Security Requirements

The system shall comply with GDPR and other relevant privacy regulations.

The system shall implement role-based access control for sensitive data.

## 5.4 Software Quality Attributes

Reliability: The system shall have an uptime of 99.9%.

Usability: The system shall be usable with minimal training.

Maintainability: The system shall be designed for easy updates and maintenance.

# 6. Other Requirements

Subject to change per project needs.

# Appendix A: Glossary

CPPS: Campus Parking Permit System.

LPR: License Plate Recognition.