Software requirement specification (SRS) document template

Project name: License Plate Scanner App

Date: November 30, 2023

Version: 1.0

By: William McMains & PlateMate

Revision history			
Version	**Author**	**Verson description**	**Date completed**
1.0	William McMains	First Release	November 30, 2023

Review history			
Approving party	**Version approved**	**Signature**	**Date**
Backend	1.0	Backend	November 29, 2023
Frontend	1.0	Frontend	November 29,2023
Project Manager	1.0	Project Manager	November 29, 2023

Approval history			
Reviewer	**Version reviewed**	**Signature**	**Date**
Backend	0.9	Backend	November 25, 2023
Frontend	0.9	Frontend	November 25,2023
Project Manager	0.9	Project Manager	November 25, 2023
Project Manager	1.0	Project Manager	November 28, 2023

Table of contents

1

Introduction

Describe the purpose of the document.

1.1 Product scope	List the benefits, objectives, and goals of the product.

The benefits of this project will be in that it will be able to centralize information that has to do with parking and vehicle registration. The objective is to create a central database where users can keep track of their permits and companies/enforcers can feel secure in the fact that they have a central secure database that keeps track of their consumers.

1.2 Product value	Describe how the audience will find value in the product.

The average person will find this product useful because they won't have to wait for their parking permits to come in the mail to hang them on their mirror on window. They will simply buy their permit and it will be valid from that moment on and they can park in case of emergency or just for convenience.

1.3 Intended audience	Write who the product is intended to serve.

The product is mainly intended to serve people who regularly deal with parking permissions. That being businesspeople in large cities with paid parking, students going to college who need to pay for parking, and enforcers who need to check someone's permit by checking their license plate.

1.4 Intended use	Describe how will the intended audience use this product.

The intended use of this is to keep track of the permits that people have and allow for the speedy execution of the checking of those permits by parking officials.

1.5 General description	Give a summary of the functions the software would perform and the features to be included.

The project will be able to take pictures of someone's license plate and will interpret the letters combined with manual input to keep in a database. This will allow their name and plate to be registered in the secure database. This database will then be able to be accessed by companies so that when someone buys a permit it will attach itself to them in the app. Then when an enforcer goes around to check permits in a place where there are no meters for example, they will be able to quickly scan a license plate where it will match through the database to see if they are permitted to be there.

Functional requirements

List the design requirements, graphics requirements, operating system requirements, and constraints of the product.

This project will be implemented with HTML for the website and so will run with HTML5 and CSS. It will not be graphics heavy and so it will run on any modern system. It will be coded in SQL using MongoDB and Java as a frontend so it will be compatible with PC and Android when it gets ported over so It will not be compatible with Apple.

3

External interface requirements

3.1 User interface requirements	Describe the logic behind the interactions between the users and the software (screen layouts, style guides, etc).

The user interface will be basic because the application itself is meant to be informational in scope. They will have a profile page which they will be able to create and add information to. Through this profile page they will be able to enter a license plate to register where they will take a picture of the license plate as well as enter it manually to add redundancy. The enforcement/business side will also have a business profile where they can add subordinate accounts for traffic enforcers to access the database and use it as needed.

3.2 Hardware interface requirements	List the supported devices the software is intended to run on, the network requirements, and the communication protocols to be used.

The applicable devices will be website form with a mobile site to allow for on the go use. A mobile app will be implemented if time allows for convenience. The network requirements will be minimal due to mainly being a text-based database with pictures added onto separate profiles for license plate info. It will use SQL and .net for basic internet communication and database handling.

3.3 Software interface requirements	Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.

The frontend of the product will use HTML5, CSS, and JavaScript for implementation into a website form. This will keep its data stored in a backend SQL database with Python being used to handle the frontend to backend data cross. This will be handled through MangoDB.

3.4 Communication interface requirements	List any requirements for the communication programs your product will use, like emails or embedded forms.

The embedded forms will be handled through the backend using Python and sent to the SQL database to store as encrypted information. This will handle the username and info such as their profile info, email, license plate, and car information.

4

Non-functional requirements

4.1 Security	Include any privacy and data protection regulations that should be adhered to.

The data will be encrypted using 256-bit encryption to make sure that people license plate, credit card info, and permit information is safe and won't be stolen.

4.2 Capacity	Describe the current and future storage needs of your software.

The storage needs won't be large even with many users and will be able to use a cloud server from Google with only a 5Tb need as a guess overall.

4.3 Compatibility	List the minimum hardware requirements for your software.

Due to its minimal nature, it will not need a high-quality computer to gain access. Any mobile device with access to Google Play will be able to download it on a mobile device. For a computer it will work on any device that can run HTML5 and probably a minimum of i3 CPU processor.

4.4 Reliability	Calculate what the critical failure time of your product would be under normal usage.

Using the average amount of time for our supposed assets and the failure rate while using redundancy on a system using the Cloud our CFT will be around 230 hours of run time before needed upkeep.

4.5 Scalability	Calculate the highest workloads under which your software will still perform as expected.

The highest workload where it will still be working is 2000 requests per second before the read/write speed starts to faulter due to the functionality of the server. Though due to the minimal nature of the project this will likely never happen.

4.6 Maintainability	Describe how continuous integration should be used to deploy features and bug fixes quickly.

Through the use of Git and clone instances new feature will be able to be implemented easily as our frontend and backend developers continue to create the best possible product available.

4.7 Usability	Describe how easy it should be for end-users to use your software.

The ease of use will be near perfect in the fact that through minimal use of flashy objects the only things that a use will need to worry about is creating their profile, inputting their license plate thorough a short tutorial that allows them to enter the information, and then simply going to their profile to view or enter any documents connected to their profile. Where as the business side will be able to see a list of all users that have one of their permits and will be able to automatically verify and add permits to a profile through their administrator profile.

4.8 Other	List any additional non-functional requirements.

Definitions and acronyms

