Software Requirements Specification

For

Vehicles Parking Space Finder App

Prepared by

Md:Robiur Rahman (193014049)*

Hasin Imam Mehran (193014020)

Asib Shikdar Pranto (193014067)

< Friends Circle Company>

Date Created: 23.10.23

Table of Contents

Introduction

Purpose

The purpose of the "Vehicles Parking Space Founder" app is to provide a convenient and efficient solution for users to locate available parking spaces for their vehicles. This app aims to address the following purposes

- Because of that there are no towing problems.
- And the vehicle has been parked as a secure condition.
- There is no risk for the vehicle owner for parking the car.

Document Conventions:

The main chapter headings are given with font. Times and font size of 18.the subheadings of each chapter are given with font. Times and size of 14 Emphasize is given in Italy's. Critical information in specific scenarios is given in Bold Words.

Abbreviation	**Explanation**
API	Application programming interface
SPMS	Smart parking management system.
GUI	Graphical user interface
IDE	Integrated development environment
IT	Information technology
OS	Operating system
SQL	Structure query language

Intended Audience and Reading Suggestions:

Developers: to be sure, they are developing the right project that fulfills requirements provided in this document.

User: to get familiar with the idea of the project and suggest other features that would make it even more functional.

Product Scope

In the modern age many people have vehicles .Vehicles are now a basic need. Every place is under the process of urbanization. There are many corporate offices and shopping centers, hospitals, schools, government organizations etc. These systems might be computerized or non-computerized with the help of computerized system. We can deliver a good service to customers who want to park their vehicle into any place.

References:

1. www.google.com.
2. www.w3schools.com
3. www.DocFoc.com
4. www.slideshair.com
5. www.vehicle project.com

Overall Description

Product Perspective:

SPSFA ought to have the option to give an essential and simple exchange of data.

For example: it ought to have the option to eliminate the correspondence holes between a representative and the client.

Product Functions:

There are a lot of functions in parking space finder apps .these function can be placed into broader categories as follows:
1. Efficiency and functionality.
2. Security over performance.
3. Lightweight and elegant.

User Classes and Characteristics:

A user can just have his/her worker Id number or vehicle plate number as username so assuming he joins the parking then no one but he can login.

Operating Environment:

There are many OS environment where our products compatible version of OS as describes in hardware and software requirements, so it does not need to rely on any other operating system. Our

product will provide the best features of both Android system.so it will have to provide complete support to old and new applications designed for both Android systems.

Design and Implementation Constraints:

Servers and the numbers of client can get to or can be online immediately. More is the quantity of client more will be the organization traffic and the server arrives in a down state .As an operating system the major constraint for parking space will be hardware and its performance there are several trivial and non-trivial constraint which are from both development perspective and context perspective. Some of them are listed as follows here;
1. Memory of the device.
2. Cope with boot up procedures of different manufacturers.
3. Prevent deadlock during application executions.

User Documentation:

First of all, the end users will be required to sign our terms and conditions while first running the Parking space finder product in their devices. These terms and conditions will contain and agreements relevant to the features implemented in the parking space finder product.
Once the parking space has successfully run up for the first time, a detailed documentation page will be shown right on the device, which can be skipped as of now, upon clicking skip, the user will get instant no how to access the detailed documentation inside .

Assumptions and Dependencies:

Parking space finder apps should work even when the organization traffic is high .Server ought to have a power reinforcement just as a data set reinforcement. Parking space ought to be viable with a large portion of the network frameworks for example past and most recent one.

External Interface Requirements:

GUI;

User Interfaces:

This interface should be profoundly natural or intuitive on the ground that there won't be a help for the client who is working the Framework .at most of the spots assist work area with being accommodation client's comfort. The pin and secret word classification ought to be kept up this should be possible by utilizing reference marks at the secret phrase board. Legitimate security message ought to be shown all things consideration of the spots.

Hardware Interfaces:

The parking space apps will be interfacing with hardware all the time and the real challenge during development is to make most efficient possible usage of hardware components. Otherwise there will be no market of new parking space that does not even provide better performance .The parking space will be capable of running on all devices whether that be media and online servers and operating system.it will be interfacing with all the hardware components.

Software Interfaces:

- The last application should be bundled in a setup program with the goal that the items can be effectively introduced on machines.
- For the information base taking care of MYSQL should be introduced .These items are open sources items.

System Features:

As discussed in section 2.2.the parking space apps will provide all the features that predecessor parking software offer along with new features to improve the quality and performance of the system.

System Feature 1:

Accessibility

Description and Priority

Accessibility is one of the highest priority features of the system analysis this accessibility not only includes better UI

Stimulus/Response Sequences

Normal controls accessibility is enhanced by adding clearer, simple buttons, representing whatever the specific section does.

4.1.3 Functional Requirements

REQ-1: Hardware monitor
REQ-2: Back end SQL

System Feature 2 (and so on)

Other Nonfunctional Requirements

User Requirements:

- Need for an application that makes communicating easy and comfortable.
- An application that enables user to park a vehicle with safe and secure
- Need for an application that is easy to use and widely available and hence a web application

Security Requirements:

The system security problem can be divided into following related issues. They determines the file structure, data structure and access procedures. System Security and system is responsible for controlling access to system resources. This will include sensitive data. The system must therefore include a certain amount of protection for such data .system security in concerned with all aspects of these arrangements

Software Quality Attributes :

Following the attributes here:
1. It should be equipped with a current and archive database.
2. All records can easily be updated.
3. It should have its own firewall.

Business Rules:

Some business rules following here:
1. Project manager will be able to lead the development team apart from accessing the code.
2. Developers can access both backend code and frontend of the operating system.
3. End users can only use the front end of the system and use its features. They are not allowed to and will not be given access to the code of the system.

Appendix A: Glossary:

Terms	**Definitions**
API	Application programming interface
Availability	The quality of being able to be used obtained
Components	A unique, identifiable constituent part of a content item composed
GUI	Graphical user interface
Hardware	The machine's wiring and other physical components of a computer's other electronic system.
IDE	Integrated development environment
JAVA	Programming language

Appendix B: Analysis Models:

Process model:

Iterative model:

The iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented.

An iterative life cycle model does not attempt to start with a full specification of requirements instead. The process is then repeated, producing a new version of the software for each cycle of the model.

Why not the Waterfall model:

Waterfall is a linear, sequential model where the next phase starts only after the completion of the previous phase. The waterfall model can be used when requirements are not changing frequently. Besides high amounts of risk and uncertainty poor model for long and ongoing projects So waterfall models is not appropriate for our parking management system project.

Below is our class diagram of this app: