SOFTWARE REQUIREMENTS SPECIFICATION

“Project name”

THE SMART PARKING SYSTEM

1. Introduction

The purpose of this Software Requirements Specification SRS document is to provide a detailed overview of our Smart parking system software, its parameters and goals. The product whose software requirements are specified in this document is an android based simulation system that locates and tracks available parking spots near CC3 building in IIIT Allahabad campus.

1.1. PURPOSE

The ever-increasing traffic has enforced the use of Smart Parking Systems. This application will aid in the optimization of parking spaces, regulation of traffic flow, and improve the efficiency of parking operations.

From Driver’s point of view :-

* Driver’s are unknown whether the slot of parking is available or not. So they need to travel the full area and if all slots are occupied then a traffic like situation is created.
* It's difficult to see the lines of parking slots and sometimes people on purpose or by mistake park their car in a wrong way which also occupies another slot.
* Few driver’s leave their car/bike for a prolonged period.

From Admin’s point of view :-

* The management is unaware of the status whether a vehicle is parked properly or not.
* It's difficult to keep a record of all complaints.
* When a vehicle obstructs another vehicle it leads to arguments and complaints.
* It is hard to keep track of people entering and leaving the building.
* The trivial parking system takes a lot of space due to improper alignment of vehicles.
* Less security is given to the parking spaces.

This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2. Document Conventions

MAIN SECTION TITLES

Font: Arial Face: None Size: 20

Subsection Titles

Font: Arial Face: None Size: 16

Standard Text

Font: Arial Face: None Size: 12

1.3 Intended Audience and Reading Suggestions

The audiences for this document include the system developers and the users. The system developer uses this document as the authority on designing and building system capabilities. The users review the document to ensure the documentation completely and accurately describes the intended functionality.

This version – version 1.0 - provides general descriptions of the system. The system developer should review the document to ensure there is adequate information for defining an initial design of the system. The users should review the document to affirm the features described are needed, to clarify features, and to identify additional features needed within the system.

1.4 SCOPE:

<Project\_name> will allow the person to save their time in hectic and trivial parking, thus allowing a person to do something more productive, which makes them happy and mindful. We have mentioned briefly the features that are In-scope and Out-of-Scope using our application :

In-Scope:

* Efficient parking space division as the sensors will guide the driver to park the vehicle properly.
* The driver can update the extra amount of time he will be occupying the parking space in the application so that the new user is updated with the waiting time.
* It assists the user in finding a parking spot in a matter of minutes.
* With the Smart Parking System, it is almost impossible to steal a vehicle. Thus, giving extra security.
* There is a good chance for the accidents to reduce, considering that there will be fewer distracted drivers on the road looking for open parking spaces. Thereby improving road safety.
* The Smart Parking System helps the management to have a record of people entering and leaving the building.
* It also results in fewer parking complaints.

Out-of-Scope:

* Restricting the parking time.
* Always finding an empty parking spot.

1.5 REFERENCES:

IEEE SRS Format.

2. Overall Description

2.1. Product Perspective:

<Project\_name> is aimed to increase efficiency in the parking system near CC3 building , IIIT Allahabad. This project aims at overcoming all the problems faced by both management and drivers in the traditional parking system. This will save time for the management and drivers in parking their vehicle correctly.

2.2. Product Functions:

<Project\_name> has following use cases:

|  |  |
| --- | --- |
| Use case | Description of use case |
| Admin: | |
| Launch | Allows admin to launch the app and displays landing page |
| Admin | User enters the admin mode and is asked to login |
| Login | Allows admin to login using the credentials |
| Display Panel | Displays the details of the parking space to the admin |
| Configuration Button | Allows admin to update different details of the parking space |
| Notify user as timer ends | Allows admin to notify the user on registered mail or phone to vacate the space or increase the timer. |
| Update FAQ | Allows admin to see and update the FAQ |
| View Complains | Allows to admin to see all the complains and can also be resolved |
| Quit | Quit the system |
| User: | |
| Launch | Allows user to launch the app |
| User | User enter the user mode |
| Display Panel | Displays the different details of the parking space |
| Register Vehicle | The user can register the vehicle and enter the time they require in the parking space. |
| Update timer | User can extend the timer |
| Vacate space | User specify that he is going to vacate the space |
| View FAQ | Allows user to see the FAQ |
| Complain | The user can register any complaints or improvements in the parking system. |
| Quit | Quit the system |

2.3. User Characteristics:

* The user should be familiar with android devices and language.
* The user must have an internet connection.

2.4. Principal Actors:

There are two principal actors in our <project\_name> “Admin” and “User”

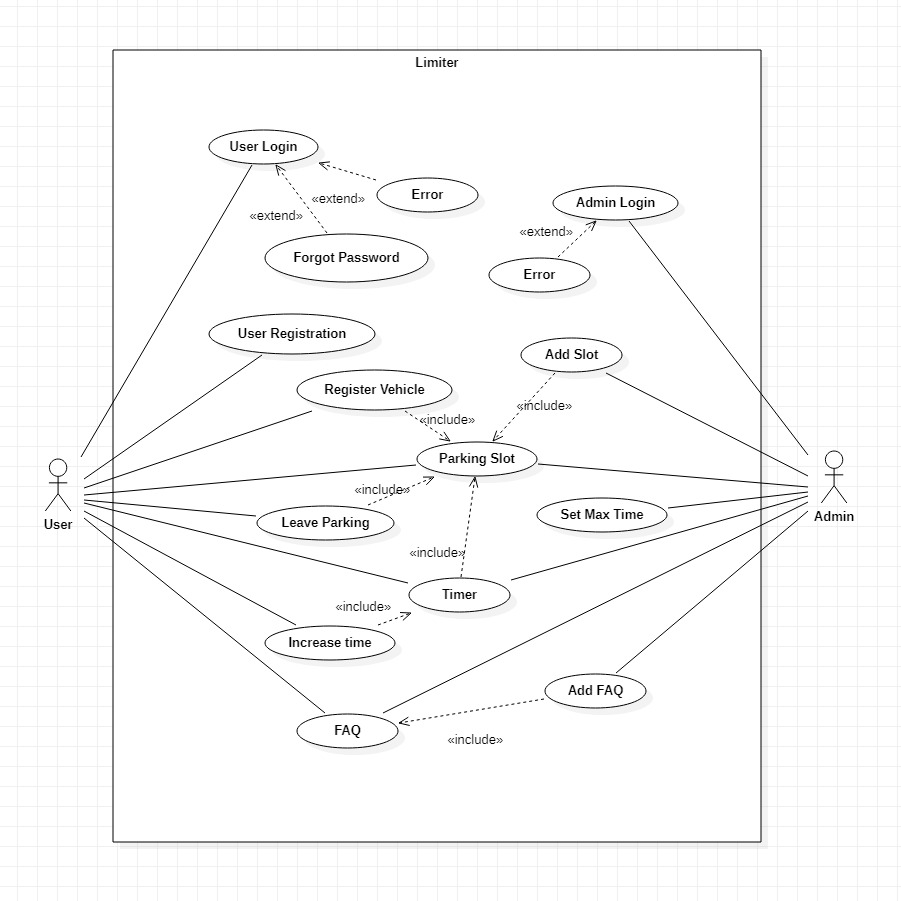
2.5. General Constraints:

* Working on the project requires internet connection.
* It is a single user application.

2.6. Assumptions and Dependencies:

* As said, working on the app requires an internet connection.
* Admin should be active and sensors should be regularly monitored for correct data.

Appendix A : Use Case Diagram



3. Specific Requirements:

3.1. Functional Requirements:

We describe the different use cases to provide a detailed description of functional requirements.

**Use Case 1:**

Name: User Login

Summary: Allows users to enter the application.

Actors: User

Pre-Condition:

* Internet Connectivity
* Email id and password should be known

Main Success Scenario:

* User enters the credentials.
* Clicks submit

Extension:

If the entered password is wrong, the user is notified through a dialog box. The user may click on the forgot password to change the password if that is the case.

Post-Conditions:

Main screen is displayed if the credentials are correct.

**Use Case 2:**

Name: Forgot Password

Summary: Allows users to change the password.

Actors: User

Pre-Condition:

* Internet Connectivity
* Email id should be logged in.

Main Success Scenario:

* User enters the email.
* Clicks forgot password
* Finds OTP in his mail
* Enters OTP in dialog box displayed
* Changes the password
* Clicks submit.

Extension:

If the email is not entered or the OTP is not correct, the user is notified through a dialog box.

Post-Conditions:

Login page is displayed for the user to login again.

**Use Case 3:**

Name: Admin Login

Summary: Allows admin to enter the application.

Actors: Admin

Pre-Condition:

* Internet Connectivity
* Email id and password should be known

Main Success Scenario:

* Admin enters the credentials.
* Clicks submit

Extension:

If the entered password is wrong, the admin is notified through a dialog box.

Post-Conditions:

Main screen is displayed if the credentials are correct.

**Use Case 4:**

Name: User Registration

Summary: Allows a new user to register into the application.

Actors: User

Pre-Condition:

* Internet Connectivity
* Knowledge of how to access the app

Main Success Scenario:

* User enters the credentials.
* Clicks submit

Extension:

If the email or mobile number is not valid , the user is notified through a dialog box.

Post-Conditions:

Main screen is displayed if the credentials are correct.

**Use Case 5:**

Name: Vehicle Registration

Summary: Allows a new user to register his vehicle at a particular spot.

Actors: User

Pre-Condition:

* Internet Connectivity
* User should know the required credentials

Main Success Scenario:

* User enters the credentials
* Clicks submit

Extension:

If the vehicle ID is not valid or the type of vehicle does not match, the user is notified through a dialog box.

Post-Conditions:

Vehicle gets registered if credentials are correct.

**Use Case 6:**

Name: Parking slot

Summary: Allows a user to see the parking slot and how much it is occupied.

Actors: User, Admin

Pre-Condition:

* Internet Connectivity
* Knowledge of how to access the app

Main Success Scenario:

* User clicks on display parking button on the navigation bar
* A screen appears displaying the slots which are occupied, going to be occupied and free. The occupied slots have a timer running which tells the time after which it is going to be free.

Extension:

Register Vehicles for the user.

Post-Conditions:

Display panel can be seen. Moreover a window appears for registration as soon as a car is parked in the slot to register the vehicle for the user.

**Use Case 7:**

Name: Add slot

Summary: Allows admin to add new parking slots to the existing one.

Actors: Admin

Pre-Condition:

* Internet Connectivity
* Updated data which needs to be changed.
* Admin must be successfully logged in.

Main Success Scenario:

* Admin enters the slot ID of the slot which he tries to add
* Clicks submit

Extension:

If data is not saved by any chance it prompts the admin through an error page.

Post-Conditions:

Dialog box notifying update successful.

**Use Case 8:**

Name: Leave Parking

Summary: Allows a user to leave the parking slot before their slot time is up.

Actors: User

Pre-Condition:

* Internet Connectivity
* User must be successfully logged in
* Vehicle must be registered

Main Success Scenario:

* User clicks the option and the spot turns yellow indicating it is going to be free.

Extension:

Parking spot changes color in the display panel.

Post-Conditions:

Display panel can be seen with yellow color in that particular parking spot.

**Use Case 9:**

Name: Set Maximum Time

Summary: Allows admin to set maximum time a vehicle can stay in any of the parking slots

Actors: Admin

Pre-Condition:

* Internet Connectivity
* Updated data which needs to be changed.
* Admin must be successfully logged in.

Main Success Scenario:

* Admin enters the new maximum time of the slot which he tries to add
* Clicks submit

Extension:

If data is not saved by any chance it prompts the admin through an error page.

Post-Conditions:

Dialog box notifying update successful.

**Use Case 10:**

Name: Timer

Summary: Allows user and admin to see how much time is left for each booked slot to become available.

Actors: Admin, User

Pre-Condition:

* Internet Connectivity
* Admin or User must be successfully logged in.

Main Success Scenario:

* Shows the amount of time a vehicle will be staying at a given slot

Extension:

If the time is up a notification will be sent to the registered email ID and mobile number of the user.

Post-Conditions:

The time will be changing on the parking slots accordingly.

**Use Case 11:**

Name: Increase timer

Summary: Allows user to increase the time he can stay at a given slot

Actors: User

Pre-Condition:

* Internet Connectivity
* User must be successfully logged in.

Main Success Scenario:

* User enters the extra time he will need
* Clicks submit

Extension:

If data is not saved by any chance it prompts the admin through an error page.

Post-Conditions:

Dialog box notifying update successful and the user gets extra time.

**Use Case 12:**

Name: FAQ

Summary: Allows admin and users to view FAQs.

Actors: Admin, User

Pre-Condition:

* Internet Connectivity
* Admin/User must be successfully logged in.

Main Success Scenario:

* Click the FAQ section.
* FAQs are displayed.

Extension:

If any error occurs in the process it prompts the admin through an error page.

Post-Conditions:

Answers to their questions are displayed.

**Use Case 13:**

Name: Add FAQ

Summary: Allows admin to add new FAQs.

Actors: Admin

Pre-Condition:

* Internet Connectivity
* Updated data which needs to be changed.
* Admin must be successfully logged in.

Main Success Scenario:

* Admin enters the new FAQ which he tries to add
* Clicks submit

Extension:

If data is not saved by any chance it prompts the admin through an error page.

Post-Conditions:

Dialog box notifying update successful.

3.2 Non-Functional Requirements:

* Admin can only login with a pre-registered username.
* Admin should be able to push notification when application is under maintenance or any other technical difficulties are faced.
* One can be directed to a parking spot only when the user opens the display panel.

3.3 Hardware Requirements:

Should run on an android device.

3.4 Software Requirements:

TBD

3.5 Design Constraints:

* Security: The files which contain the information of the user should be secured against malicious deformations.
* Fault Tolerance: Data should not become corrupted in case of system crash or power failure.

4.System Features

4.1 Admin Login

4.1.1 Description :

This feature helps the admin to login using their mail id and password.

Data required:

* Email ID of admin\*
* Password\*

4.1.2 Stimulus:

When the user clicks on the “login” button he will be redirected to the dashboard from where he can use all the features of this application.

4.1.3 Functional Requirements:

REQ -1 : The necessary fields should be filled in without being left blank.

REQ - 2 : Validation of email id and password given by the user.

4.2 Configuration

4.2.1 Description:

With the help of this feature the admin will be able to edit the number of slots present in the parking area and give them ID’s. The admin can also set the maximum time that each vehicle can be parked in a particular parking slot.

Data required:

* Max slots\*
* Each slot ID\*
* Max time per vehicle

4.2.2 Stimulus:

When the user clicks on the “submit” button the new data will be set and he will be directed to the home page.

4.2.3 Functional Requirements:

REQ -1 : The necessary fields should be filled in without being left blank.

4.3 Display

4.3.1 Description:

This feature shows the occupied and available parking slots distinctly using different colors. It will also display the timer showing how much time the slot is expected to be occupied.

4.3.2 Stimulus:

When the user clicks on the “Display parking” button the page will redirect here where we can see all the parking slots.

4.4 Registration

4.4.1 Description:

The user is expected to register using this feature which will help the user book that particular parking slot for a given amount of time.

Data required:

* Vehicle ID\*
* Parking ID\*
* Name\*
* Contact Number\*
* Email ID\*

4.4.2 Stimulus:

When the user clicks on the “submit” button the data will be set and that particular slot will be allocated to the user. Then the page will redirect to home.

4.4.3 Functional Requirements:

REQ -1 : The necessary fields should be filled in without being left blank.