

ONLINE PARKING SYSTEM

CH-1 INTRODUCTION

1.1. Problem Statement

To automate the existing private parking system

Parking is the act of stopping and disengaging a vehicle and leaving it unoccupied. Parking on one or both sides of a road is often permitted, though sometimes with restrictions.

Vehicles continue to outnumber existing parking spaces, thus clogging roads. Incidences of violence over occupancy, deformed cars due to a space crunch, and overcharging for parking are some problems that result.

A large number of vehicles come out on the streets during festivals, exerting immense pressure on parking spaces. This means more cruising, chaos, quarrels and long queues than usual.

- Public parking is managed by local government authorities and available for all members of the public to drive to and park in.
- Private parking is owned by a private entity. It may be available for use by the public or restricted to customers, employees or

residents.



1.2. Limitations of Current System

In so many places, we see lots and lots of different vehicles, belonging to different people of different classes parked on the streets. It's nothing new for people to choose to park on the side of the road.

Existing solutions:

- Only parking at the time of arrival is available.
- There are no good and efficient solutions to the existing problems.

Drawbacks

- Parking is often metered. It will cost you money and you have to work with a time limit.
- Being parked on the street heightens the risk of your vehicle getting knocked up by a vehicle that's trying to pull in at the front or at the rear of you. You never know when a vehicle may collide with the side of your car either.

- Parking on the curb probably means that you're not directly in front of your destination. This can be inconvenient for those who are carrying loads or who are late for a specific event.
- In most cases, especially in areas where parking is scarce, one must pay to park in a parking lot. Entry and exit access are often controlled at these types of lots to ensure those parking pay the required fee. The types of products used to enforce payment are called access controls.
- In a very large parking field, it is easy to get lost or have trouble finding one's vehicle.
- It requires a maintenance contract with the supplier.

1.3. Proposed System Solution

To the above-mentioned Problems/Disadvantages, the following are the proposed solution:

- It will provide the user-interactive interface that helps the user to reserve the parking space for their vehicle according to their comfort.
- It will provide the layout of the whole parking lot which shows every available and non-available space.
- It will give notifications regarding the time limit of the vehicle in the parking, i.e., 30 mins before release of the parking, so that they may extend if wanted.
- It would also show the digital fee receipt according to the time for which the vehicle is parked.
- The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities.

- Multiple options of payment for parking services, which include mobile payment.

1.4. Advantages of Proposed Solution

- **Digital tokens**
- **No more big queues.**
- **It tells whether the parking space is reserved or not.**
- **It will reduce man-power.**
- **Time-saving** – It will help to manage your precious time.
- **Optimized parking** – Make parking easier and more efficient.
- **Reduced pollution** – Searching for parking burns around one million barrels of oil a day.
- **Improved Safety** – Safety and security can be achieved easily by smart parking.

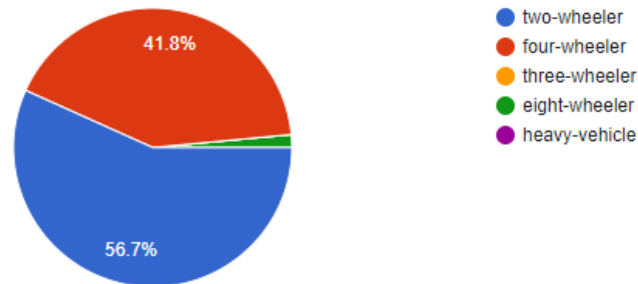
1.5 Google form link

<https://forms.gle/uSKfLZy5XoXdfixg9>

Google form responses

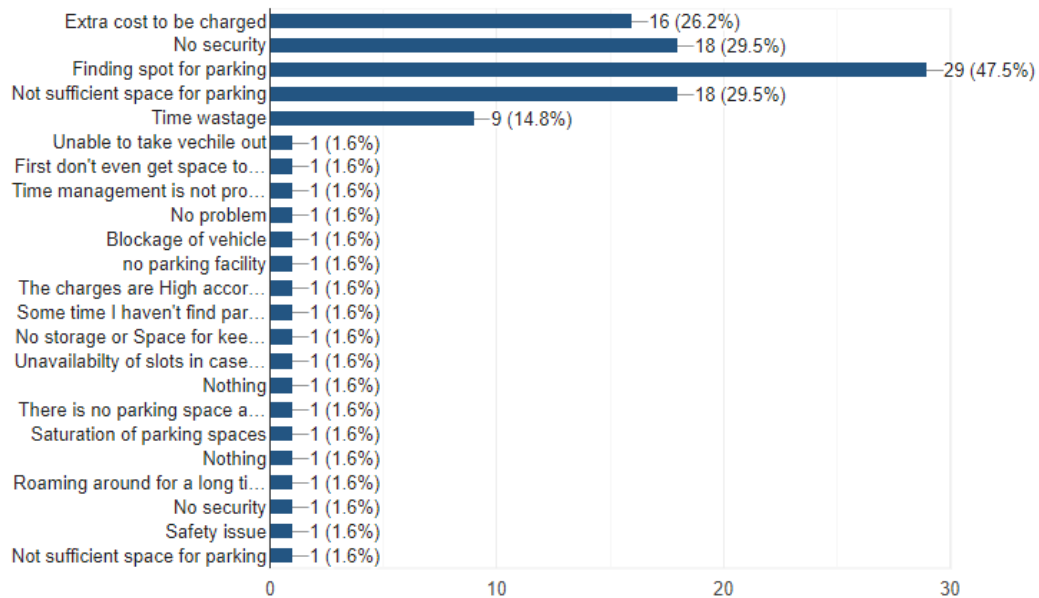
1. What type of vehicle you have?

67 responses



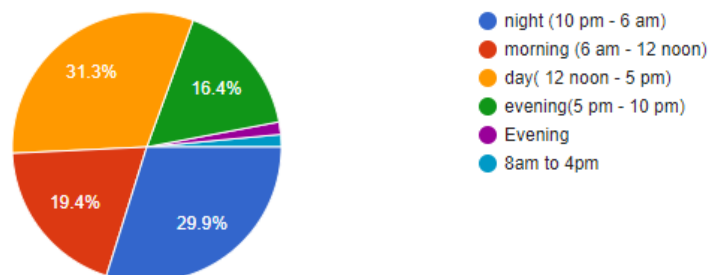
2. What type of problems you face at normal parking system

61 responses



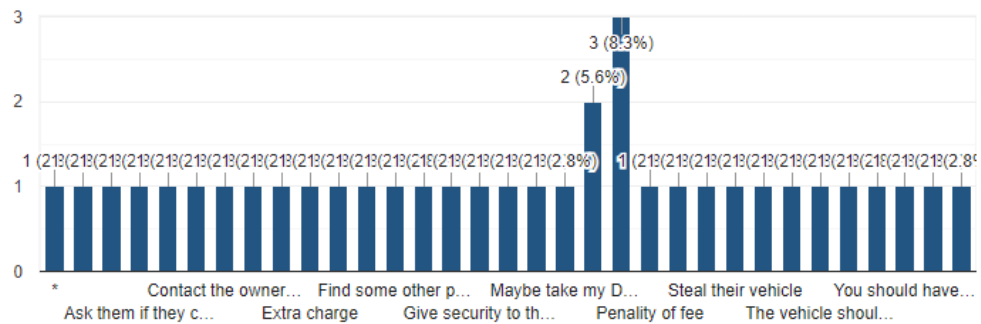
3. Your preferable time for parking

67 responses



4. Suggestion if someone is unable to take the vehicle after the given time slot (what action should we take with your vehicle?)

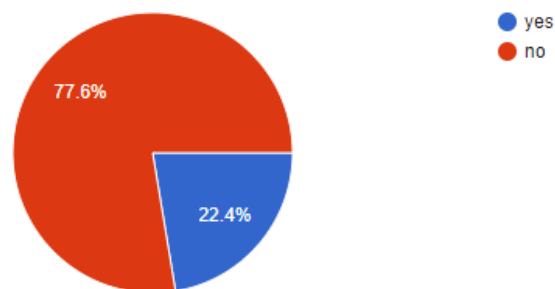
36 responses



5. Are you willing to give your vehicle details and your contact number?

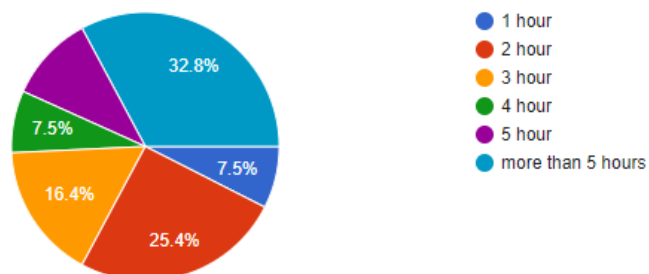
5. Are you willing to give your vehicle details and your contact number?

67 responses



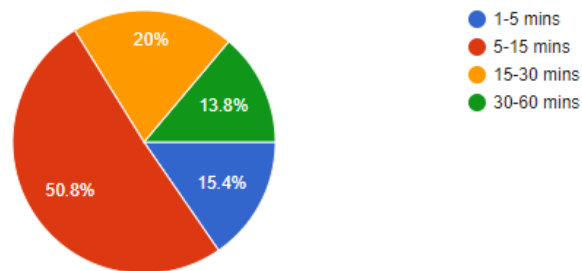
6. Duration for which you want to hire the parking slot

67 responses



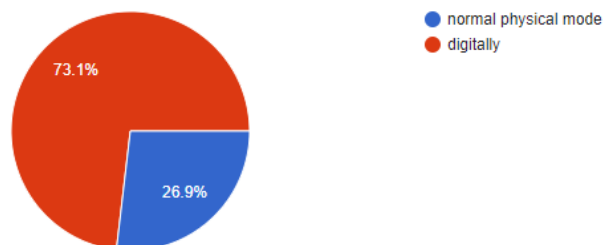
7. How much time you spend looking for a parking spot in normal parking system

65 responses



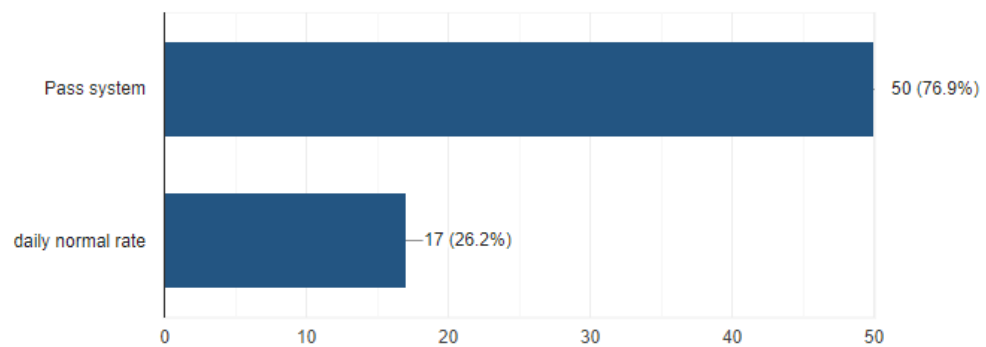
8. What type of token system you prefer.

67 responses



9. If you park your vehicle daily, So what would you prefer pass system or daily normal rate..

65 responses



CH-2

Introduction:

The SRS is produced at the culmination of the analysis task. The function and performance allocated to software as part of the system engineering and refined by establishing a complete information description, a detailed functional description, a representation of system behaviour, indication of performance requirements and design constraints, appropriate validation criteria and the other information related to requirements. The SRS is a technical specification of requirements of the Online Parking System. This specification describes what the proposed system should do without describing how it will do it. It also describes complete external behaviour of the proposed system.

2.1.1 Purpose:

The main purpose of our system is to make the parking facility easier and more convenient and to develop an app which automates the parking system. This document serves as the guide for the users of this app.

2.1.2 Document Conventions

The SRS document is designed as per the IEEE 830-1998 standard SRS template.

2.1.3 Intended Audience and Reading suggestion:

The intended audience for the system is the users of the app. The users will be the general people seeking for the parking space availability in the parking area. Admin is able to block any particular slot and can manage parking charges

2.1.4 Product scope

In scope:

Users can book slots for parking.

Users can choose the time duration of parking.

Users can see the way to the parking slot.

Users can unbook the slot.

Users can see their parking History.

Out of Scope:

Management of more vehicles than the limited space.

2.1.5 References:

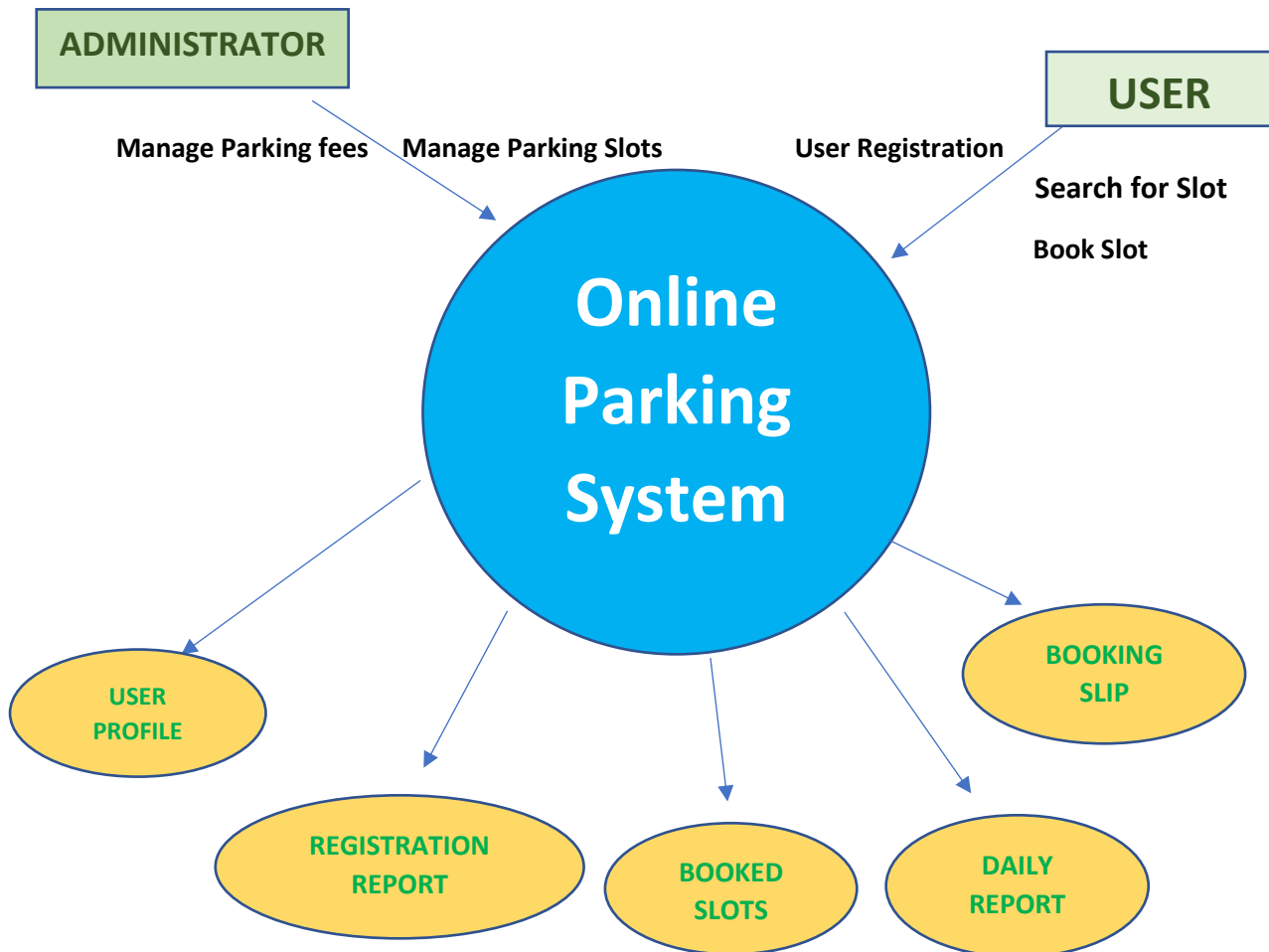
IEEE 830-1998 standard SRS template [IEEE SRS Template](#).

2.2 Overall Description

2.2.1 Product Prospective:

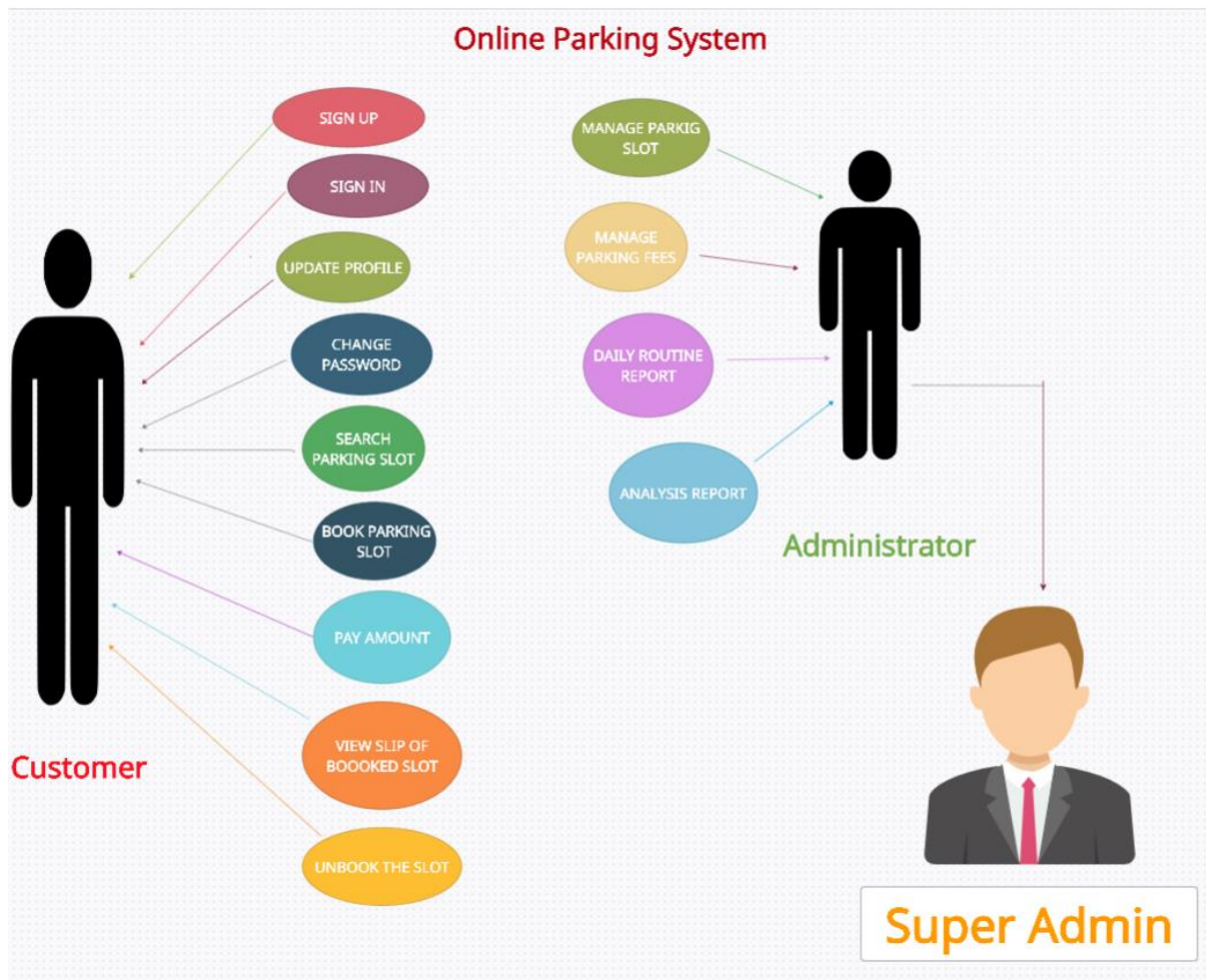
Data Flow Diagrams:

• 0-Level DFD



2.2.2 Product Functions

ParkVic APP



Use Case Diagram

User case 1: sign up

Primary actor: app user

Pre-condition: log in page is available

Success scenario:

System shows sign-up page

User fill the details like name email password etc.

User submit the details

System stores the details in database.

System shows successfully registered notification.

System provides user id

Exception scenario:

Invalid email id

User already exists.

All information not filled.

Basic flow:

Sign up->enter information->submit->successfully registered->provide user id

Alternate flow:

Sign up-> enter information->submit->user already exists

Sign up-> enter information->submit->incomplete information

User case 2: log in

Primary actor: Registered user

Pre-condition: Log in page is available

Success scenario:

System shows login page

User enter his user id and password

User submit the details

System checks the details in database.

Log in successful.

Exception scenario:

Invalid email id

Invalid password

User not registered.

Basic flow:

Log in -> enter id and password ->log in->successfully logged in

Alternate flow:

Log in-> enter user id and password->log in->invalid id and password

Log in-> enter user id and password ->log in->user not exists

User case 3: Forgot password page

Primary actor: Registered user

Pre-condition: Should be registered before

Have a valid and registered email id

Success scenario:

User clicks on forgot password button

System show forgot password page

User enter the email

System checks the details in database.

Provide create new password option

User creates new password

System displays Log in

Exception scenario:

Attempts to change password exceeds

Basic flow:

Forgot password -> enter email id-> system check in database->create new password->log in page

Alternate flow:

Forgot password-> enter email id->attempts exceeds

Use case 4: Booking of parking slots

Primary actor: Registered user

Pre-condition: Login page is available

success scenario: System show parking slots

User choose slot information (slot time and date)

System check for chosen parking slot

System confirms the parking slot

System directed to payment page

Slot booked confirmation message

Exception scenario: Slots not available

All information not filled-> show error

Basic flow:

Show parking slot->fill information-> chose slot->system confirm the slot-> payment page->confirmation message

Alternate flow:

Show parking slot->fill information-> chose slot-> slot unavailable

Show parking slot->fill information-> chose slot-> incomplete info.

Use case 5: Payment page

Primary actor: Registered user

Pre-condition: Slot should be booked

Main success scenario: System show payment page

User choose the payment option

User pay the amount

Slot booked confirmation message

Exception scenario: Payment not done

Server error

Basic flow:

Show payment page-> choose payment option->pay the amount>confirmation message

Alternate flow:

Show payment page-> choose payment option->pay the amount-> payment not done

Show payment page-> choose payment option->pay the amount-> server error

Use case 6: Unbook the slot

Primary actor: Registered user

Pre-condition: Should booked a slot before

Main success scenario: System show the unbook page

User unbook the slot

System returns the payment after
deducting fine

Redirected to booking page

Exception scenario: unable to unbook

Payment not returned

Basic flow:

unbook page->unbook the slot->returns payment

Alternate flow:

unbook page->unbook the slot->payment not returned

Use case 7: Parking history

Primary actor: Registered user

Pre-condition: Vehicle should be parked before

Main success scenario: System show history page

User see his history

Exception scenario: no history available

Basic flow:

History page-> show history

Alternate flow:

History page-> no history

Use case 8: Daily Analysis

Primary actor: Admin

Pre-condition: Should be logged in

Main success scenario: System show the analysis page
System gives the report to admin

Exception scenario: No report generated

Basic flow:

Analysis page-> show report

Alternate flow:

analysis page-> no report generation

2.2.3 User Classes and Characteristics:

1. General User:

- a. Users will be able to use the app to book Parking slots.
- b. Users can manage their timing and can view their parked History.

2. Admin

- a. Manage parking slots
- b. Manage parking fees
- c. Able to keep track of booked slots

3. Super Admin

- a. Manage all important information and view records

2.2.4 Operating Environment

The app will operate on mobile devices for both android and ios. An internet connection is a must.

2.2.5 Design and Implementation Constraints:

1. Only English language is supported.
2. Users have Android or iOS based devices.
3. User must have correct password and user-id to login
4. Booking of slots will depend upon the availability of slots and

timing.

2.2.6 Assumptions and Dependencies:

- 1.All the data entered will be correct and up to date.
- 2.The user is connected to the internet and is willing to share personal Information online.

2.3 External Interface Requirements.

2.3.1 User Interfaces:

GUI (Graphical User Interface)

2.3.2 Hardware Interfaces:

- 1) OS - Android/iOS
Android version 7.0 (Nougat) and up
IOS version 8.1.3 and up
- 2) Storage - 200 MB
- 3) RAM – 312 MB

2.3.3 Software Interfaces:

HTML, CSS, JAVASCRIPT, PHP

OS - Android Studio

Database- SQL

2.3.4 Communications Interfaces:

- 1 . Android
- 2 . SMS Service protocol
- 3 . POP protocol
- 4 . TCP/IP

2.4 System Features

2.4.1 Sign up

2.4.1.1 Description

This functionality is available for all the users of the app. Users can sign up using correct information.

2.4.1.2 Stimulus/Response Sequence

1. Start of App
- 2 . Entering Valid information
like Email, name, password.
3. signed up

2.4.1.3 Functional Requirements

Internet Connectivity and Data usage Permission is Necessary

2.4.2 Login

2.4.2.1 Description

This functionality is available for all the users of the app. Users can login using identical login information.

2.4.2.2 Stimulus/Response Sequence

1. Start of App
- 2 . Entering Valid Login
Credentials
3. Logged In

2.4.2.3 Functional Requirements

Internet Connectivity and Data usage Permission is Necessary

Users should be signed Up for the App.

2.4.3 Booking of Parking Slot

2.4.3.1 Description

This functionality is only available for the registered user. The user will be able to book a parking slot for an unoccupied space.

2.4.3.2 Sequence

1. User has to login to the app.
2. Choose the unoccupied slot
3. Enter vehicle details like
vehicle no.
4. choose preferable time
5. Enter Identity Verification
details like (Aadhar card no.)
6. pay the booking charge
7. Slot is Booked.

2.4.3.3 Functional Requirements

1. Internet connectivity is a must.
2. Users must verify their identity using a suitable verification or they won't be able to use this function.

2.4.4 Payment of Booked Parking Slot

2.4.4.1 Description

This functionality is only available for the registered user who already booked a slot.

User will be able to pay the booking charge for the parked slot.

2.4.4.2 Sequence

1. User has to book slot.
2. Choose the payment method
3. Enter payment details like (UPI).
4. pay the booking charge.
5. Payment is successful.

2.4.4.3 Functional Requirements

1. Internet connection is must.
2. Users must have online payment system.

2.4.5 Unbooking of Parking Slot

2.4.5.1 Description

This functionality is only available for the registered user who already booked a slot.

The user will be able to unbook a parking slot which he already booked

2.4.5.2 Sequence

1. User has to select the booked slot.
2. Choose the unbook option
3. pay the unbooking charge.
4. Slot is unbooked.

2.4.5.3 Functional Requirements

1. Internet connectivity is must.
2. Users must book a slot before otherwise they won't be able to use this function.

2.5 Other Non-functional Requirements

2.5.1 Performance Requirements

Response time-The system will give responses within 1 second after checking the user information and other information.

Capacity-The system must support 1000 people at a time

User interface- User interface screen will be very responsive.

2.5.2 Safety Requirements

If any failure happens in the database, such as a disk crash, the recovery method can be used to restore a past copy of the database that was backed up in storage .

2.5.3 Security Requirements

All the admins have unique logins so the system can only be accessed by those with login. Only system Admins can change records and valuable data. No intruders allowed.

2.5.4 Availability:

App can be downloaded from Google play store or Apple store

2.5.5 Software Quality Attributes

App will be available 24*7. It can be accessed on any device that meets the minimum requirement.

Modularity: App is modular.

Maintainability: Source code of software will be easy to maintain.

Usability: software can be used again and again without distortion.

Accessibility: Administrator and many other users can access the system but the access level is controlled for each user according to their work scope.

Accuracy: The reliability of the information and output. Can depend on and be sure of the outcome.

REFERENCES:

1. IEEE 830-1998 standard SRS template [IEEE SRS Template.](#)
2. Aggarwal, K . K., Software Engineering.