Software Requirements Specification

Parking Management System

Group Members Names:

1) Ramit Kumar Gupta: 70362019318

2) Siddhant Jain: 70362019370

3) Tanay Singh Chauhan: 70362019429

Submitted to: - Dr Vikas Jain

Software Engineering

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

In this SRS document, we provide a detailed overview of our software product, its parameters and goals. 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.

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

VEHICLE PARKING MANAGEMENT SYSTEM

1.1 Purpose

The **purpose** of this project is to build a vehicle parking management system which will **ease** the overall process of finding a suitable parking spot by reducing the time needed to find one and by managing the allocated parking space as **efficiently** as possible. The system will also make it easy to keep track of the entry and exit of vehicles, by maintaining a listing of vehicles within the parking lot. This will also **enhance** security measures altogether.

1.2 Scope

With the major increase of automobiles, for the time when the vehicles are not in motion and need to be parked, finding a suitable parking spot has become a problem with people parking their vehicles wherever they can find space. This not only wastes time and effort but also ends up consuming so much of the road space causing traffic and a dangerous environment for the pedestrians.

This imbalance is partially due to ineffective usage of parking spaces and miscalculations when it comes to space allocations and could be readily solved if a proper system comes into effect that efficiently uses space and allocates parking.

1.3 Intended Audience

Owners of locations that have a dedicated parking space within their establishments. These locations can include: malls, theatres, hospitals, banks, public parking spaces etc.

Any person who owns a vehicle, be it a two-wheeler or a four-wheeler and travels using that vehicle on a regular basis, especially in crowded localities where people park their vehicles randomly and vehicles occupy most of the road-space.

2.5 References

- IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- Example SRS_1
- Example SRS_2

CHAPTER-2

OVERALL DESCRIPTION

2.1 Product Perspective

The Vehicle Parking Management System is a GPS-based mobile application as well as a desktop application which helps people to find the closest parking spots based on the user's current position and other specifications like locality, vehicle type, parking duration and more. It even allows them to book a suitable spot to their liking if that parking space is supported by our system.

The application should be free to download from either a mobile phone application store or similar services. Parking owners can provide their parking space information using the web-portal. This information will act as the basis for the search results displayed to the customer. An administrator also uses the web-portal in order to administer the system and keep the information accurate. The administrator can, for instance, verify parking owners and manage user information.

Furthermore, the mobile application needs both Internet and GPS connection to fetch and display results. All system information is maintained in a database, which is located on a web-server. The software also interacts with the GPS-Navigator software which is required to be an already installed application on the user's mobile phone.

By using the GPS-Navigator, users can view desired parking places on a map and be navigated to them. The application also has the capability of representing both summary and detailed information about the parking locations such as number of vehicles currently parked, nearby landmarks, parking fees etc.

Our system extends to being a desktop application which will help the parking operators to manage their parking spaces as efficiently and safely as possible and will also allow them to keep a track of vehicles entering and exiting the parking space enhancing security measures.

2.2 Product functions

This system will consist of three parts: a mobile application, a desktop application and a web portal, each having their own respective functions which are described below:

2.2.1 Functions of Mobile Application

The mobile application will mainly be used by the customers. It will be used to find nearby parking spaces, view information about those parking space and then if needed, user can even book a parking slot in the supported parking spaces. Information about the parking spaces can include things like about how many parking slots are left, what type of vehicles are parked at what slots, parking fees, distance of the user to that parking space etc. The app will also provide a navigation feature that will enable the user to navigate over to a selected parking space.

2.2.2 Functions of Desktop Application

The desktop application will be used by the parking operators, operating a particular parking space. This application will allow the operators to maintain the parking space as efficiently as possible. Parking operators will be able to confirm the allocation of a parking slot for a customer and also deallocate it. The application will also assist the operator in the calculation of parking fees and in maintaining data of the vehicles entering and exiting the parking space.

2.2.3 Functions of Web Portal

The web portal will be used for managing the data related to usernames and passwords of the admins, parking operators and of customers, as well as data related to the parking spaces. Owners of parking spaces will be able to register their parking spaces on this web portal. Customers can also register their accounts on his web portal, they will also be able to search and navigate to parking spaces using this portal.

Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. All three, the mobile application, the desktop application and the web portal will communicate with the database, however in slightly different ways. The desktop application will only use the database to get data while the mobile application and the web portal will also add and modify data. All of the database communication will go over the internet.

2.3 User Classes and Characteristics

There are three types of users that will interact with the system: users of the parking booking website, parking operators and administrators. Each of these three types of users has different use of the system so each of them has their own requirements.

2.3.1 User Class 1 - The Users

These are the users that can only access the website to view the parking grid of a particular place and then search and book a convenient parking slot.

Characteristics:

- **Identification:**
  - They will be identified using their login IDs and passwords.

- **Accessibility:**
  - Users will be able to first search for a place registered with our company.
  - If the location that the user is searching for is indeed registered with our company, the user will then be able to view the parking grid of that location.
  - Users will also be able to book an available parking slot for a certain duration of time.
  - After a successful payment, user will get access to a dynamic map that will allow him to navigate to the booked parking slot.

2.3.2 User Class 2 - Parking Operators

Parking operators are those users that will have a direct access to the parking management software and also a certain portion of the website and will act as an interface between the users and the main software.

Characteristics:

- **Identification:**
  - They will be identified with their company's name and employee ID.
- **Accessibility:**
  - They will be able to confirm the incoming requests from the users for the bookings of parking slots.
  - They can allocate parking slots and deallocate them.
  - They can also access the website restricted to the portion of their own company and manage info such as managing or updating the parking grid, address etc.

2.3.3 User Class 3 - Administrators

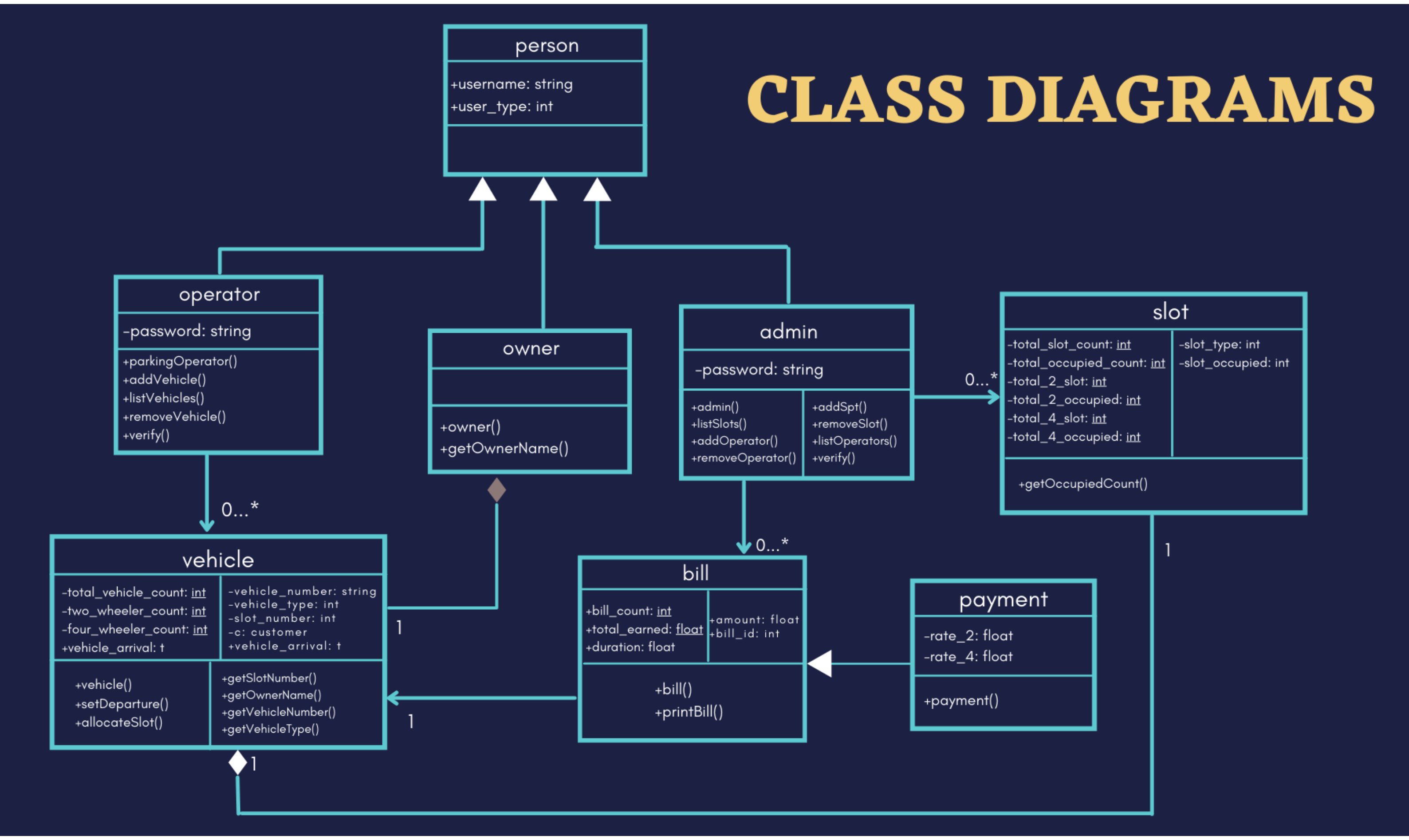
These are the users that can only interact with the web portal and are responsible for the maintainability of the entire system.

Characteristics:

- **Identification:**
  - They will be identified with their employee ID.

- **Accessibility:**
  - They will manage the overall system and its working so as to sustain a smooth working, if something goes wrong or if an instance of the system produces an error then they will handle it.
  - They will have access to everything in the system with the exception of any company's private data that is stored on the database.

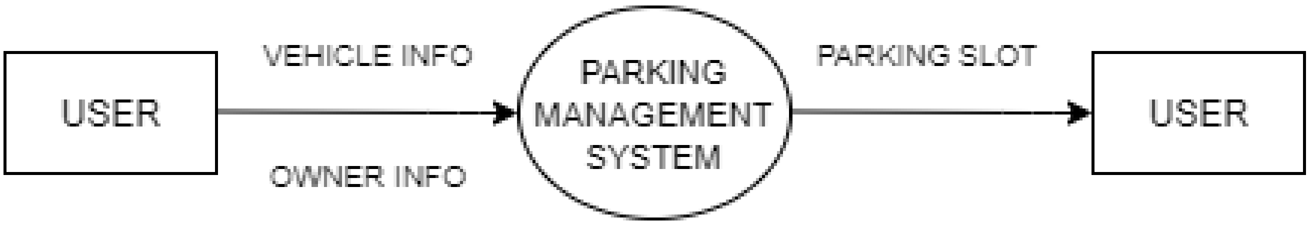
2.4 Class Diagrams



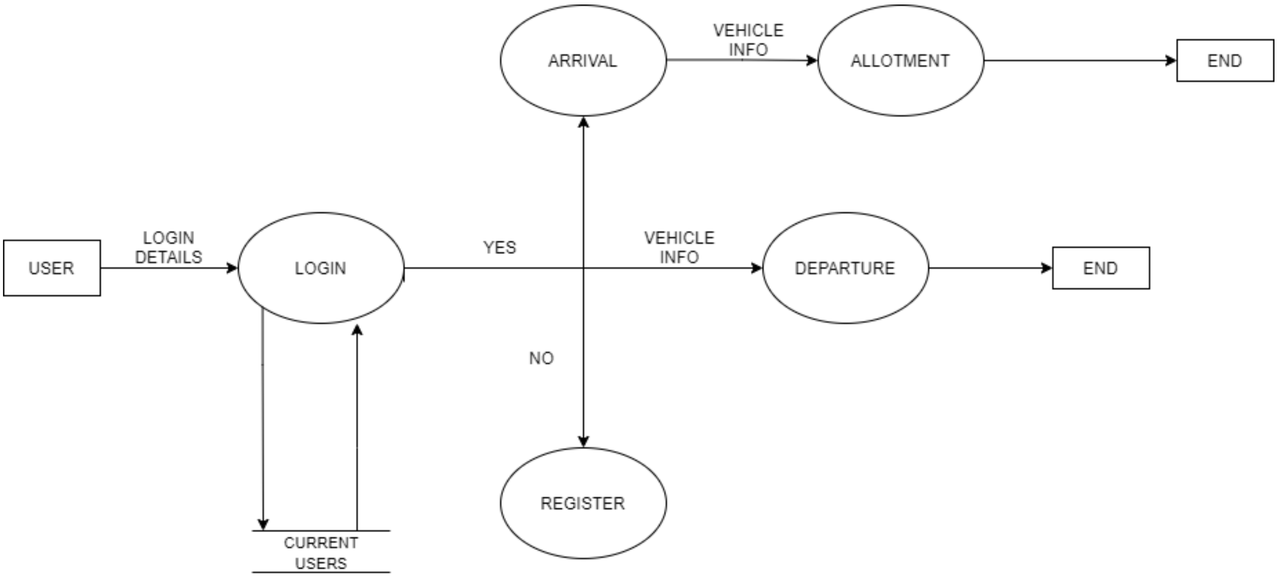
2.5 Data Flow Diagrams

A data flow diagram shows the way information flows through a process or system. It includes data input and outputs, data stores, and the various sub processes the data moves through.

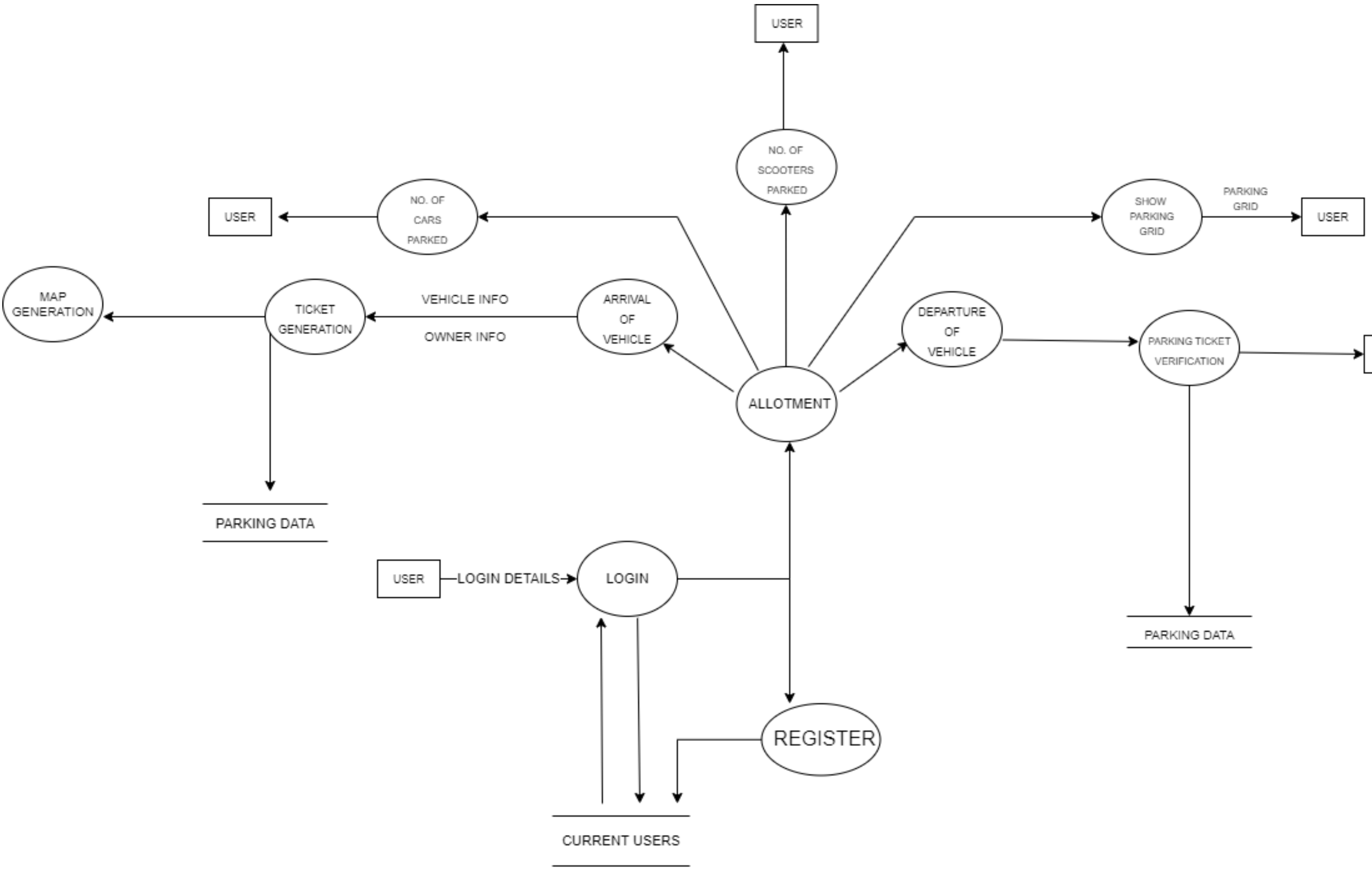
2.5.1 Level-0 DFD or Context diagram:



2.5.2 Level-1 DFD:



2.5.3 Level-2 DFD



CHAPTER-3

INTERFACES AND CONSTRAINTS

3.1 Software Interfaces

The mobile application communicates with the GPS application in order to get geographical information about where the user is located and the visual representation of it, and with the database in order to get the information about the parking spaces.

The communication between the database and desktop application, mobile application and web portal consists of both modifying and reading data.

3.2 Hardware Interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

3.3 Product Constraints

- The mobile application is constrained by the system interface to the GPS navigation system within the mobile phone. Since there are multiple systems and multiple GPS manufacturers, the interface will most likely not be the same for every one of them. Also, there may be a difference between what navigation features each of them provide.

- The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

- Both the web portal and the mobile application will be constrained by the capacity of the database. Since the database is shared between both applications it may be forced to queue incoming requests and therefore increase the time it takes to fetch data.

3.4 Product Assumptions and Dependencies

- One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all.

- Another assumption is that the GPS components in all phones work in the same way. If the phones have different interfaces to the GPS, the application needs to be specifically adjusted to each interface and that 6 would mean the integration with the GPS would have different requirements than what is stated in this specification.

CHAPTER 4

SPECIFIC REQUIREMENTS

4.1 Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.

Since our system has 3 components, each has its own set of functional requirements which we will specify one by one.

4.1.1 Functionalities of the Mobile Application

Here the user is: customer

R.1: Register

Given that a user has downloaded the mobile application, then the user should be able to register through the mobile application by entering the following info:

Input: “register option”,

Output: user prompted to enter the keywords.

R.1.1:

Input: user-name, password, email-address and phone-number(optional).

Output:

- user prompted to confirm his email-address to finish registration or
- “email-address already in use”, user gets prompted to login using that email-address or to use another email-address for registration.

R.2: Login

Once the user has registered, he/she can then login.

Input: “login option”,

Output: user prompted to enter the keywords.

R.2.1:

Input: user-name and password.

Output:

- “login successful” and the user gets sent to the homepage of the application.
- user informed about a bad password or username.

Processing: Password validation.

R.3: Forgot password

This functionality allows the user to recover his account in case he has forgotten his password.

Input: “forgot password option”

Output: user gets prompted to enter his registered email-address.

R.3.1:

Input: registered email-address.

Output:

- “a password reset link has been sent to your email-address” or,
- “this email-address is not registered with us”.

Processing: email-address validation, reset-password link generation and sending the link to the user's email-address.

R.3: Search

Once the user has logged in, he can then search for a nearby parking space on the basis of certain filters.

R.3.1:

Input: “search option”

Output: user prompted to enter the search keywords

R.3.2:

Input: location, vehicle-type, min-parking-rate and max-parking-rate.

Output: The user will get search results for parking spaces.

R.4: No match found

This is the functionality for when a user tries to search for a parking space but no matches are found.

Input: none

Output: “no results found” a new search page is opened up for the user below the “no results found”.

R.5: Book

Once the user has selected a parking space after searching, he will be given the option to book a parking slot given that the slots are not full and the parking space is not closed.

R.5.1

Input: “book option”

Output: user prompted to enter booking keywords.

R.5.2

Input: His vehicle type (2-wheeler or a 4-wheeler) and date-duration for the parking.

Output: user will be prompted to pay for the duration stated earlier

R.5.3

Input: Paytm number and amount

Output: if the payment is successful then “slot booked” otherwise “payment unsuccessful” and the user will be sent his bill on the email-address.

R.6: Add more duration

This functionality allows the user to increase the duration of his parking given that he has already booked one.

Input: “extend option”

Output: user prompted to select the parking slot and then asked for how much duration to increase.

R.7: Cancel

This functionality enables the user to cancel any booked parking slot, given that the time of parking hasn’t already started.

Input: “cancel option”

Output: user prompted to select the booked parking slot that he wishes to cancel.

R.8: Profile page

This functionality is about the user’s profile page which has all the user details.

Input: “profile page button”

Output: user’s profile page is displayed which contains information such as: name, email address and phone number.

4.1.2 Functionalities of the Desktop Application

Here the user is: Parking Operator

R.1: Manage bookings

This functionality allows the user to handle bookings which includes: add a booking (vehicle arrival), cancel a booking, verify a booking and end a booking (vehicle departure).

R.1.1: Add a booking

This is the first way for a customer to get a parking slot.

Input: customer’s details: name, contact number, customer’s vehicle number, vehicle type and duration for which the vehicle will be parked.

Output: Booking ID, Parking slot has been allocated.

Processing: System creates a new booking entry in the database and input all the booking details and generates a unique booking ID which is then given to the customer.

R.1.2: Verify a booking done through the mobile application

This is another way for the customer to get a parking slot, this one uses the mobile application for the purpose of booking a parking slot.

Input: system generated bill number (that was given to the customer after successful payment for the parking slot)

Output: Parking slot has been allocated.

Processing: System input the customer’s information using the bill number by fetching details from the database and updates the parking info.

R.1.3 Cancel a booking

This functionality allows the parking operator to cancel a booking.

Input: booking id

Output: booking has been cancelled, money refunded.

Processing: System fetches the booking details from the database by using the booking ID and updates the database by mentioning the booking as cancelled.

R.1.4 End a booking

This functionality allows the parking operator to successfully end a booking cycle, given that the booking was done through the desktop application and not through the mobile application.

Input: booking id

Output: parking duration and the customer gets prompted to pay for the duration of parking.

Processing: System fetches the booking details from the database by using the booking ID, calculates the amount to be paid by the customer.

R.1.4.1 Payment

Input: payment number and

4.1.3 Functionalities of the web portal

Here the users are **admins** and **parking space owners.**

R 1:

- **Register:**

  Input: Once the user enters the web portal he is asked to register himself.

  Output: The system should show the output as user registered or invalid credentials.

  Processing: Verify the credentials are valid or identical or not.

R 2:

- **Login:**

  Input: Once the user registers himself he is asked to login into the portal.

  Output: The system should display the message “Login Successful” or “Invalid user” if credentials are wrong.

  Processing: Verify the credentials are valid or not.

R 3:

- **Search Parking:**

  Input: Once the user selects the search parking function he is asked to enter his location and type of vehicle.

  Output: System should display the message parking location found or no parking available in the mentioned area.

  Processing: Search the parking space in the mentioned area.

Here the user is **admin**

R 4:

- **Add and Remove Parking Space:**

  Input: As the admin clicks on the add and remove parking space icon he is asked to add or remove the parking space according to the space available.

  Output: Display parking slot added or deleted or display not enough space available.

  Processing: System should be able to add or delete parking slots according to the space available.

CHAPTER 5

TESTING

What is Software Testing Technique?

Software Testing Techniques help you design better test cases. Since exhaustive testing is not possible; Manual Testing Techniques help reduce the number of test cases to be executed while increasing test coverage. They help identify test conditions that are otherwise difficult to recognize.

Different Types of Software Testing:

Given below is a list of some common types of Software Testing:

Functional Testing types include:

- Unit Testing
- Integration Testing
- System Testing
- Sanity Testing
- Smoke Testing
- Interface Testing
- Regression Testing
- Beta/Acceptance Testing
- Integration Testing

Non-functional Testing types include:

- Performance Testing
- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing
- Install Testing

- Recovery Testing
- Reliability Testing
- Usability Testing
- Compliance Testing
- Localization Testing

Testing technique suitable for our project:

We will use smoke testing for our model as **Smoke Testing** is a software testing technique performed post software build to verify that the critical functionalities of software are working fine. It is executed before any detailed functional or regression tests are executed. The main purpose of smoke testing is to reject a software application with defects so that the QA team does not waste time testing broken software applications.

Another testing model we will use is **Black Box Testing** as this will tell us that we are getting required output for the given input (Told by the customer).

White Box Testing:

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

White Box Testing is also going to be super useful for our project since it will help us to rectify most of the errors, with the help of various test case, especially because the system is very much integrated.