



College of Engineering Pune

(An Autonomous Institute of the Govt. of Maharashtra)

Software Engineering Mini Project II

Third Year Computer Engineering

Parky

Under the guidance of
Prof.Tanuja Patanshetti

Manish Arora 111803036

Riddhi Ghate 111803051

Sakshi Kalekar 111803059

Table of Contents

Table of Contents	Page number
1. Abstract	1
2. Introduction	1
3.SRS.....	1
 3.1. Introduction	2
 3.1.1 Purpose	2
 3.1.2 Document Conventions	2
 3.1.3 Intended Audience and Reading Suggestions	2
 3.1.4 Product Scope	2
 3.1.5 References	2
 3.2. Overall Description	3
 3.2.1 Product Perspective	3
 3.2.2 Product Functions	3
 3.2.3 User Classes and Characteristics	3
 3.2.4 Operating Environment	4
 3.2.5 Design and Implementation Constraints	4
 3.2.6 User Documentation	4
 3.2.7 Assumptions and Dependencies	4
 3.3. External Interface Requirements	4
 3.3.1 User Interfaces	4
 3.3.2 Hardware Interfaces	5
 3.3.3 Software Interfaces	5
 3.3.4 Communications Interfaces	5
 3.4. System Features	6
 3.4.1 Account Creation	6
 3.4.2 Renting Parking Space	6
 3.4.3 Getting the Parking Space	6
 3.4.4 Suggesting the Alternative Spots	6
 3.4.5 Cashless Transactions	6
 3.4.6 Verifying the Parking Space	7
 3.5. Other Nonfunctional Requirements	7
 3.5.1 Performance Requirements	7
 3.5.2 Safety Requirements	7
 3.5.3 Security Requirements	7
 3.5.4 Software Quality Attributes	7
 3.5.5 Business Rules	7
 3.6. Other Requirements	8
Appendix A: Glossary	8
4. Literature Review.....	9
 4.1 Tabular representation	9
 4.2 Conclusion that summarizes the literature review.....	14
5.Proposed Statement.....	14
6.Planning.....	15

7.Design.....	16
7.1 Data Flow Diagram	16
Level 0	16
Level 1	17
Level 2	18
7.2 Use Case Diagram	19
7.3 Activity Diagram	20
7.4 System Architecture	21
7.5 Class Diagram	22
7.6 Sequence Diagram	23
7.7 State Diagram	24
7.8 ER Diagram	25
8.Implementation.....	26
9.Testing.....	37
10. Github Link.....	71
11.References.....	71

1. Abstract

Due to the proliferation in the number of vehicles on the road, traffic problems are bound to exist. This is due to the fact that the current transportation infrastructure and car park facility developed are unable to cope with the influx of vehicles on the road.

To alleviate the aforementioned problems, we are developing a parking management system . Parking Management System is a web app built using MERN stack . This will provide users with a better interactive UI. The user data will be saved in the database once registered. This parking management system will help patrons to easily locate and secure a vacant parking space at any car park deemed convenient to them.If the slot is booked then patron will be redirected to some nearby parking place in the same area.Booking charges can be done prior while will help to solve the problem of cash on delivery. If patrons cancels his/her booking refund will be provided by deducting some cancelation charges. So this web app will help to reduce major traffic problems in the city.

2. Introduction

Parking Management System is a web app built using MERN stack . This will provide users with a better interactive UI. The user data will be saved in the database once registered. This parking management system will help patrons to easily locate and secure a vacant parking space at any car park deemed convenient to them.If the slot is booked then patron will be redirected to some nearby parking place in the same area.Booking charges can be done prior while will help to solve the problem of cash on delivery. If patrons cancels his/her booking refund will be provided by deducting some cancelation charges. So this web app will help to reduce major traffic problems in the city.

3. SRS

3.0 Revision History

Name	Date	Reason For Changes	Version
SRS	31-01-21	Added Section 1 & 2	1.0
SRS	15-02-21	Added Section 3, 4 & 5	1.1
SRS	27-02-21	Added user flow diagram	1.2

3.1 Introduction

3.1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Parking Management System” software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

3.1.2 Document Conventions

1. Italicized all external links and references.
2. Bold to highlight key points.
3. Font: Raleway

3.1.3 Intended Audience and Reading Suggestions

This document is intended for

Developers

In order to be sure they are developing the right project that fulfills requirements provided in this document.

Testers

In order to have an exact list of the features and functions that have to respond according to requirements and provided diagrams.

Users

In order to get familiar with the idea of the project and suggest other features that would make it even more functional

3.1.4 Product Scope

Parking Management System’ is a web application which helps people to book a parking space in their desired location. Its purpose is to save the user's time of going to a parking place and finding the parking space. It also solves traffic problems at parking areas. Beside this, it also maintains a database of incoming and outgoing vehicles coming in a particular place.

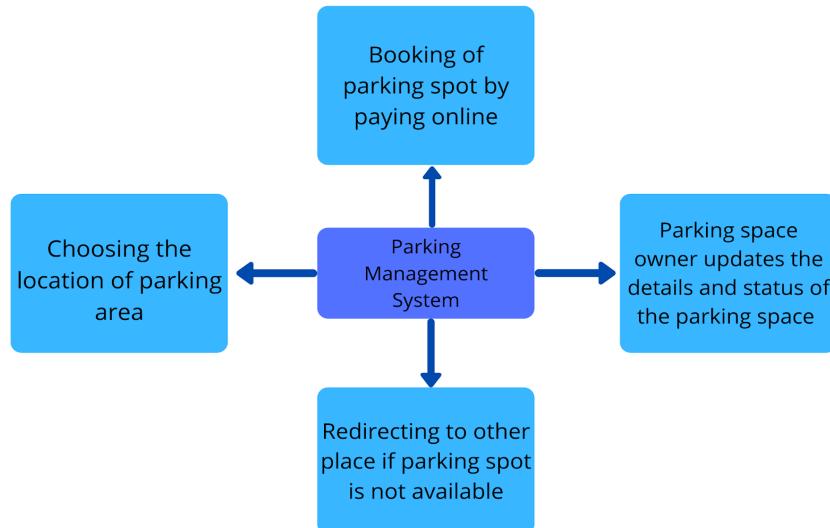
3.1.5 References

- 1.https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3163473
- 2.https://en.wikipedia.org/wiki/Software_requirements_specification

3.2. Overall Description

3.2.1 Product Perspective

This system is designed to operate initially on a local level. Later after improvements the product can be used on global level.



3.2.2 Product Functions

- The users should be able to create an account on the website, they can be an owner or rentee.
- The users should be able to change the password.
- The owners of the parking space should be able to update the availability of the space.
- Rentee should be able to enter the location and see if there are any available spots for parking and book the space.
- If a particular space is unavailable the web app should be able to suggest another space which is available nearby.
- Rentee should be able to book a space by completing an online transaction.

3.2.3 User Classes and Characteristics

There are 3 types of users in this system mainly admin, parking space owner and users of this web app.

Admin manages the rentees and parking space owner. He is responsible for verifying the parking space, for development purposes and managing the database.

Parking space owners add the parking area and keep account of availability of parking space in that particular area. He is also responsible for keeping the records of incoming and outgoing vehicles at that particular parking area.

Parky

Rentee can book the parking space in a particular area. They can also pay the owner in advance for that parking space.

3.2.4 Operating Environment

The software can operate on any of the Operating Systems like Windows 7 and higher versions, Linux or Mac which supports TCP/IP protocols. It can also operate on Android of version 8 and higher. The device should have a basic internet browser to use the system.

3.2.5 Design and Implementation Constraints

1. There are limitations to what the payment APIs allow developers to do specifically, due to privacy restrictions and hence many actions are unavailable.
2. The application is also constrained to the number of queries it can make using the Google Map API.
3. The design of the application has to be responsive and accessible as the user might open it on a small screen or a large screen.

3.2.6 User Documentation

The web portal will have a Quick Start Guide on its main page. The quick start guide will lend itself for easy access to the most fundamental functions of the Parking Management System. The guide will have full blown instructions regarding the usage of the system.

3.2.7 Assumptions and Dependencies

1. The web application is assumed to be used on a device with a web browser and stable internet connectivity. It is also assumed to work on any operating system.

The dependencies are :

1. The payments done by rentees on the application depend upon the Razorpay API.
2. The parking space locations that the rentee can see and the directions to it are dependent upon the Google Maps API.

3.3. External Interface Requirements

3.3.1 User Interfaces

The Parking Management System's user interface is specifically designed with the customers in mind, giving them convenience while they try to book the space. The interface makes sure at every point, that the customer spends most of the time using the device rather than figuring out how to use it. The interface allows the user to access the application on smaller screen devices.

Parky

1.Home Screen - The home screen offers a menu to login or create a new account on the system. It also has a quick start guide for easy access. The user can select one of the options on the menu, and is taken to the respective screen.

2.Login and SignUp Page - The user can click on any one of the options either to login or to create a new account and once done he is taken to the screen of their choice.

3.Navigation Menu - In addition, each screen has a navigation bar to quickly jump to other screens. The navigation bar contains a text field to enter search terms. Rentees who wish to book a parking space can enter the location in the search bar. A list of search results pertaining to the query is displayed.

4.Rentee Screen - The rentee screen will have options to either rent a parking space or book a parking space. If a rentee chooses to book then he is asked to enter the location and map appears with locations of nearby available spots.

5. Owner Screen - The owner screen will have options to rent a parking space which will be verified by admin.

6. Admin Screen - It will have options to verify the Parking space uploaded by owner, manage users and parking spaces.

7. Payment Screen - After choosing the location rentee is redirected to the payment screen which gives different options for payment.

3.3.2 Hardware Interfaces

The external hardware interface required for accessing the Parking Management System is the user's personal computer or any device like laptop, tablet and mobile phone. The mobile phone should be touch screen.

Minimum requirements for Desktop/Laptop:

Memory : 3.8 GiB

Processor : Intel® Core™ i3-5015U CPU@2.10GHz x 4

Graphics : Intel® HD Graphics 5500(BDW GT2)

GNOME : 3.28.2

OS type : 32 or 64-bit

Minimum requirement for Mobile/Tablet:

Android Version : 8 or Higher

3.3.3 Software Interfaces

Operating System - The application can be used on any operating system eg (Windows, Linux, DOS, etc) with a web browser.

Database - The database is used to store the details of registered users and to track their previous booking/renting history and payment history. We have chosen MongoDB database (version 4.4.4) to be used.

API - The Razorpay API is used for online payment process and the Google Maps API is used for locating the parking space.

3.3.4 Communications Interfaces

The users require a web browser and a stable internet connection for connecting to the application. The external communication of the application also involves connecting to the Razorpay API for payments and Google Maps API for viewing the map.

3.4. System Features

3.4.1 Account Creation:

Users will be able to create the account on the website. He/she can register as a user or owner. The user data will be saved in the MongoDB database. Users can update the registered data if necessary and updated will be saved in the database.

This is a high priority feature because the user can not access other features of the website till he creates the account on it.

3.4.2 Renting Parking Space:

Owner will be able to register his or her space for renting it as parking space. Owner will have to specify the details of the space like space area, address of the space, no of vehicles that can fit in that space and type of vehicle. This data will be saved in the MongoDB database using the backend express framework. This is a high priority feature as other rentees will be able to book the space only if the space is available. If any owner doesn't specify any of the above mentioned details then he will be notified about the required details.

3.4.3 Getting the Parking Space:

Rentees will be able to search for the parking space at a particular location and book the parking space before visiting that place and he/she will be directed using google maps to visit that location. This is the high priority feature as this will solve the main traffic problem in parking areas. If the particular space is not available then the rentee will be asked whether he/she wants to check for nearby locations.

3.4.4 Suggesting the Alternative Spots:

If the particular space is not available, the web app will suggest the alternative spaces in the nearby location. Rentees can choose any one location from suggested ones. This is a medium priority feature and will help to save the rentee's time in searching for a new parking location.

3.4.5 Cashless Transactions:

Rentees will be able to book the place by completing online transactions.

Parky

If he/she cancels the booking then refund will be provided by applying some cancellation charges. This is a high priority feature.

3.4.6 Verifying the Parking Space:

Admin of the system will verify the parking space details uploaded by owner. After verification the parking space will be visible to rentee otherwise the space won't be Visible to rentee.

3.5. Other Nonfunctional Requirements

3.5.1 Performance Requirements

The system must be interactive and the delays involved must be less .So in every action-response of the system, there are no delays. Delays should be less while booking the parking space. The time period for sending/receiving notifications should be as low as possible.

3.5.2 Safety Requirements

- Data of location should not be altered by any unauthorized user.
- Information about the parking location must be true

3.5.3 Security Requirements

Data of all the registered users must be secure as it may contain important documents . There should not be any privacy issues.

3.5.4 Software Quality Attributes

- **Availability :**
The system must have parking space so that users(drivers) can book the space for parking
- **Usability :**
The system should satisfy maximum user requirements

3.5.5 Business Rules

1. When a user registers at the website , it will be saved in the database.
2. Parking Space information uploaded by owner will be available to all the rentees
3. Rentees will be able to see his/her booking history and payment history
4. Rentees will get the email after booking
5. Admin can see all the users ie rentees and owners registered on the website
6. Owner can view the list of rentees who booked his space

3.6. Other Requirements

Category Requirement : There are three categories of users , admin, owners and the rentees. Depending upon the category of the user the corresponding interface is provided.If the user is an owner he has the options to update the parking space status as well as book some other parking space. Admin can book, rent and verify the spaces.

Appendix A: Glossary

1. API - Application Program Interface
2. IP - Internet Protocol
3. Rentee - One who takes something on rent
4. TCP - Transmission Control Protocol

4. Literature Review

4.1. Tabular Representation

Year	Title of Paper	Journal/ Conference Details	Methodology Used	Proposed Idea	Advantages/ Achieved Objectives In Paper	Disadvantages/ Limitations
2011	A Reservation Based Smart Parking System	Computer Science And Engineering Theses, Dissertations and Student Research	RSP uses a Dynamic pricing scheme based on The real time parking Information given by the sensors	The Management system determines the dynamic parking prices based on real-time parking Information, and broadcast live Parkingprices to users. Upon receiving dynamic Parking prices, the user selects a desired parking lot and reserves a space in the parking lot.	1.Every parking lot has access to the Internet to communicate with the management system and users, and share parking information with other parking lots. 2.The sensor system deployed in parking lot is responsible for monitoring the real-time condition of parking lots and delivers the live aggregated sensing information to the management system. The sensing information is updated on demand. Based on the state of parking lots, the Management system analyzes their occupancy status and congestion level, determines the parking prices according to dynamic pricing scheme, broadcasts the prices to all users periodically, and stores the parking information and prices for further analysis	1.The system does not use a centralized system and hence there is no synchronization of the parking states.
2011	Parking Management System	21st LAAS International Science	Eclipse Based Mobile Application	Sensors determines The presence / absence of car in The lot. Sensor	1. This system successfully locate free space for the car	1. This system can be used only when client is present in the mall.

Parky

	using Mobile Application	Conference "Horizon 2020: Advances in Sciences And Technology"	Using sensors	Status is sent To database using Arduino Mega 2560 Microcontroller And W5100 Arduino Ethernet Shield. Mobile Application built Using Eclipse software interacts With the Database for Sending and receiving data From it in JSON Format. As client enters the mall Timer starts and depending on the Time spent Parking fee will Be withdrawn from clients account	2. It locates the place of parked car 3.Manages the parking fees	2. It doesn't allow pre booking the empty parking slot.
2016	Research of Parking Management System Based on Centralized Control	6th International Conference on Mechatronics, Computer and Education Informization	PMS based on browser and server Structure and Centralized control	The system used automatic license plate recognition technology.If the vehicle has a membership then the system will recognise the plate number and let the vehicle in else a temporary card will be issued to the vehicle.When the vehicle exits membership vehicles have free parking and others should pay	1.The system is easy to maintain and update as it uses the browser and server instead of software which requires constant updating on the client side. 2.When system processes data, devices of workstation collect the data send it to the server and then returned to the workstation after being processed by the server.This improves the system's stability. 3.The system automatically recognises the license plate and checks for membership thus saving the time spent on payment.	1.The idea proposed in this paper does not take care of available parking space in the lot. 2.The system does not allow pre booking the space.

2017	Intelligent Parking System Based On Video Analytics	International Journal of Advanced Research in Computer Science and Software Engineering	Design a real Time optimized Parking Management System based on video analytics in multi-camera environment	In this paper, the video data obtained from multiple cameras will be converted to picture frames and will be processed using feature Extractor and machineLearning algorithms. After processing the objects can be identified and the vacant parking space can be located In the parking lot	1.The system can recognise the vehicle and vacant parking space successfully. 2.The video analytics of real time data can also be used to get the vehicle details like number plate, type of vehicle also the entry and exit time of vehicles.	1.The system cannot detect parking space in the outdoor area or unstructured parking areas. 2.The system does not allow pre booking the vacant parking space.
2017	Parking Management System Using Image Processing And Distributed Approach	International Journal of Advanced Research in Computer and Communication Engineering	Real time data Extraction Using Image Processing And Distributed approach	The surveillance camera will enable the system to perform image processing on live video tracking of a specific parking system. The number of parking slots available is calculated and these data is stored in the database at the central server.	The proposed framework gives An alternative way to traditional parking management System, to deal with identification Of opportunities of parking spots. This approach Comprises of distinguishing empty Parking spots by means of a Surveillance camera. The framework likewise gives drivers' information assistance. This module gives a continuous refresh of parking slots to drivers through cellphone application with a specific end goal to guide them and encourage the errand of finding an accessible parking slot.	1.The system does not allow the users to prebook the vacant space.

2018	A LowCost Smart Parking Solution for Smart Cities Based on Open Software And Hardware	Information Sciences, Technologies And Architecture Research Center	IOT based Parking Management System	Android App will help users to see the availability of parking spot. Car park uses WSN (wireless sensor network) which monitors the parking lots. parking gateway will be used for communicating with the backend	1. It uses real time visualization of parking space as it uses sensors. 2. The system uses both software and hardware based components which will enhance the quality of system	1. Online payment is not available. 2. If the parking space is not available then the system does not redirect the user to another place.
2018	Development Of an Android Application For Smart Parking System	Department of Computer Engineering, Government Engineering College Gandhinagar, Gujarat, India	PMS based on Smart Reservation And Management System using Android Application	Latitude and Longitude of the slot is stored in the database. after booking the slot, user has to reach within 2 meters radius of that location. if he is unable to reach within the 30 min booking would be cancelled.	1. Pre Booking facility of the parking slots. 2. User can view the path from current location to parking location. 3. User can view his booking details and cancel the booking.	1. Online payment is not available. 2. If the parking space is not available then the system does not redirect the user to another place.
2019	A Novel Parking Management For Smart Cities, to Save Fuel, Time and Money	The 9th Annual Computing and Communication Workshop and Conference (CCWC 2019) At Las Vegas, NV	Raspberry PI Device and Google cloud Vision API Based Parking Management System	Hardware and Software is Continuously Working. Distance sensors sense the incoming car. The 16 Raspberry Pi device in front lane take photos of car by camera and sent to google cloud vision api which Checks the License number of the car. Each car is given a Device which gives pager to driver and direction to location. Accordingly Nearest parkingSpot is allocated and data is Stored in	1. Cars are properly Parked as Instruction are given to car using pager 2. If any car is parked at wrong spot , car will be informed about the mistake. So problem of mistakes usually Made by drivers are solved . 3. If 2 empty spots are assigned to two different cars. If one of the driver parks the in place assigned to other.Then other driver will be automatically assigned new empty spot.	1. System does not allow pre booking the vacant place. 2. Online Transactions are not available

Parky

				Database and updated on display board		
2019	Smart Car Parking Management System	United Internati-Onal University, Dhaka, Bangladesh	RFID based Smart Parking System	RFID card is scanned at entry booth. vehicle's image is captured through camera. server stores image and rfid, The barrier gate is open. then the vehicle enter parking space	<ul style="list-style-type: none"> 1. Response Time between entry and exit booths have decreased. 2. As less response time, the traffic in the parking area has decreased but not beyond a certain extent. 3. It's a automated system which will open barrier automatically after scanning RFID. 	<ul style="list-style-type: none"> 1. As there is no pre booking option, it might happen that parking space is not available. 2. The cost to implement this idea is higher than web base solution as this idea requires many hardware components.
2020	The Smart Parking Management System	Internation al Journal of Computer Science and Informatio n Technolog y (IJCSIT) Vol 12	IOT based system that sends data to free and busy parking places via mobile application/net	The system works in 3 Stages. <ol style="list-style-type: none"> 1. Arduino Sensors will detect no. of free slots, booked slots. 2. WIFI Module is used for communication between sensors and mobile app. 3. Mobile app provide interface between clients and systems. 	<ul style="list-style-type: none"> 1. System detects parking space using IOT 2. Calculates the time of entry and exit 3. Allows payment by calculating the expected cost 	<ul style="list-style-type: none"> 1. System doesn't allow Pre Booking of empty parking places.

4.2. Conclusion

The research papers mentioned above show the successful implementation of Parking Management System in different domains. Various techniques like IOT, video analytics, machine learning, web applications are implemented. Many of the techniques lack the centralized approach and costly hence we have proposed to implement a web application which uses the browser-server architecture. The system needs to be user friendly and easy to understand for both users and developers and MERN stack proves to be the easiest one to implement.

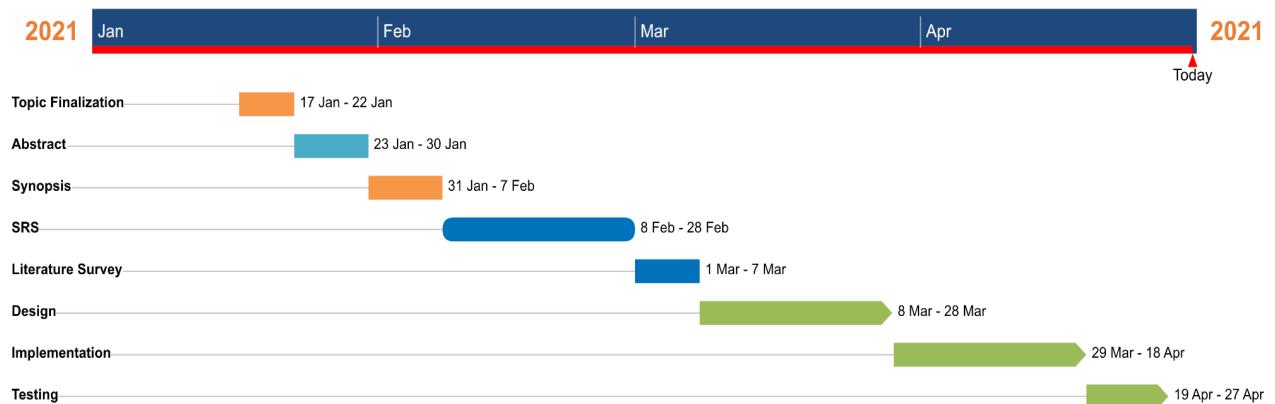
Though the research papers discussed above are able to detect the parking space availability on its own they do not provide the facility to book the parking space if it is available. Since the space is detected by the system itself it is limited to structured parking areas only and the owner of the space has less control over it. In our proposed system the owner of the space can update the status of the space thus giving the flexibility to rent space only in time when he wants, it also allows any user to prebook the space if it is seen available.

5. Proposed Statement

To design a system, where a user can register as an owner to rent parking space or register as a rentee to book parking space prior to the required time and pay for it online.

6. Planning

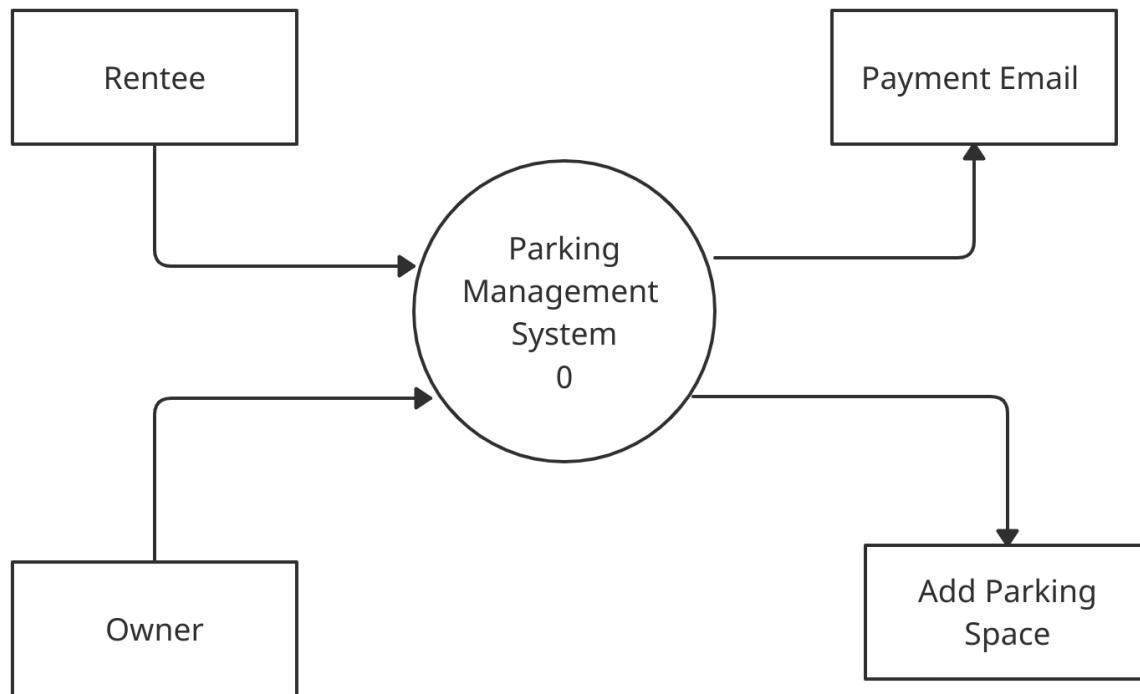
Task	Task Name	Duration	Start Date	Finish Date
Task 1	Topic Finalization	5 days	17/01/2021	22/01/2021
Task 2	Abstract	7 days	23/01/2021	30/01/2021
Task 3	Synopsis	7 days	31/01/2021	07/02/2021
Task 4	SRS	20 days	08/02/2021	28/02/2021
Task 5	Literature Survey	7 days	1/03/2021	07/03/2021
Task 6	Design	20 days	08/03/2021	28/03/2021
Task 7	Implementation	20 days	29/03/2021	18/04/2021
Task 8	Testing	8 days	19/04/2021	27/04/2021



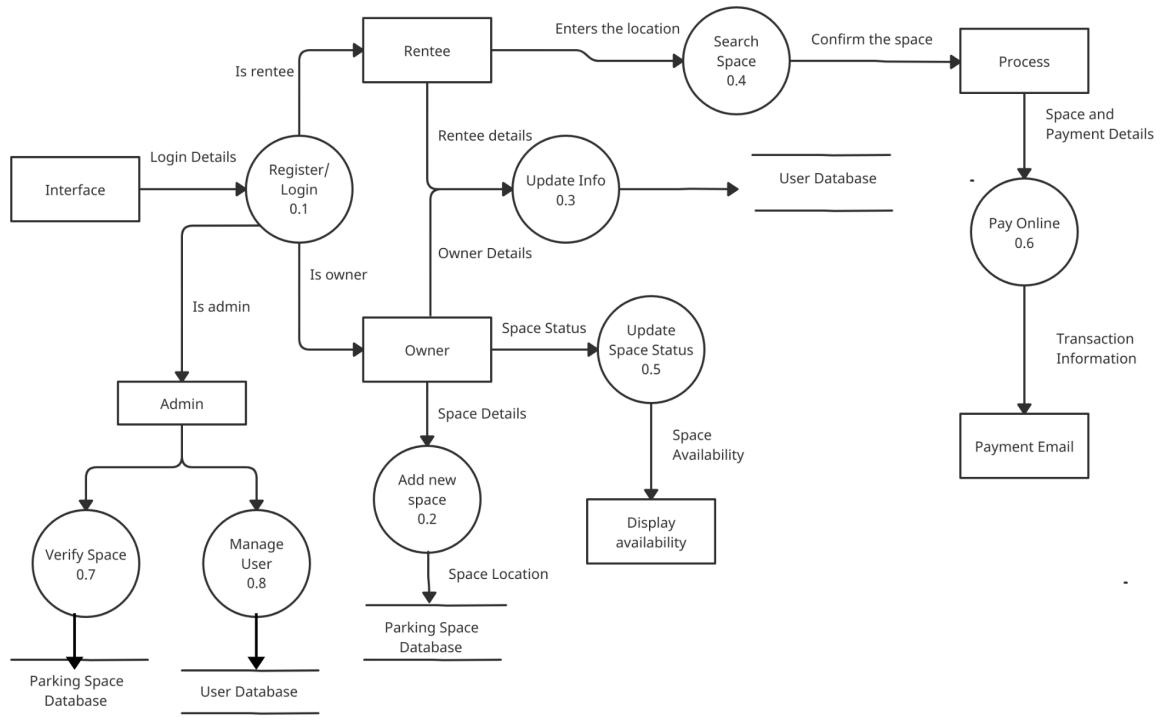
7. Design

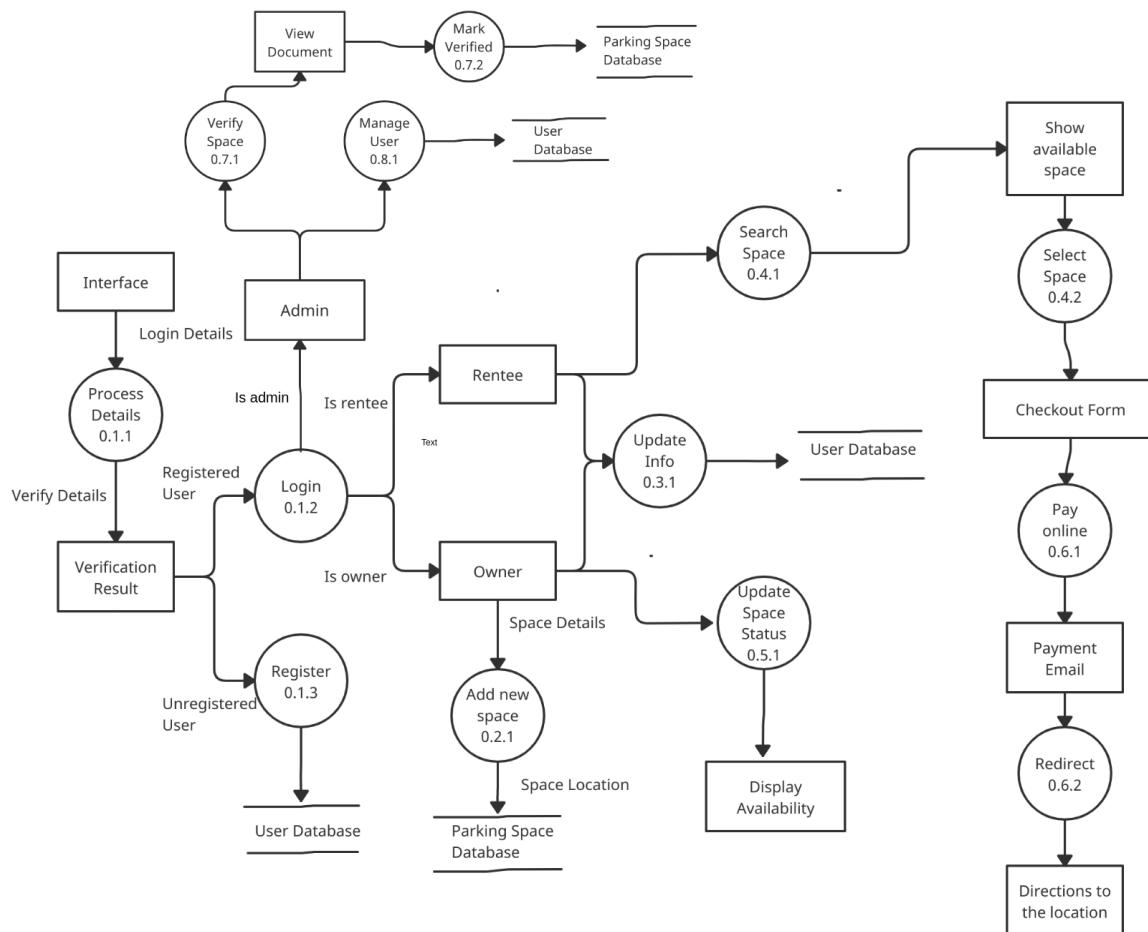
7.1 Data Flow Diagram

Level 0

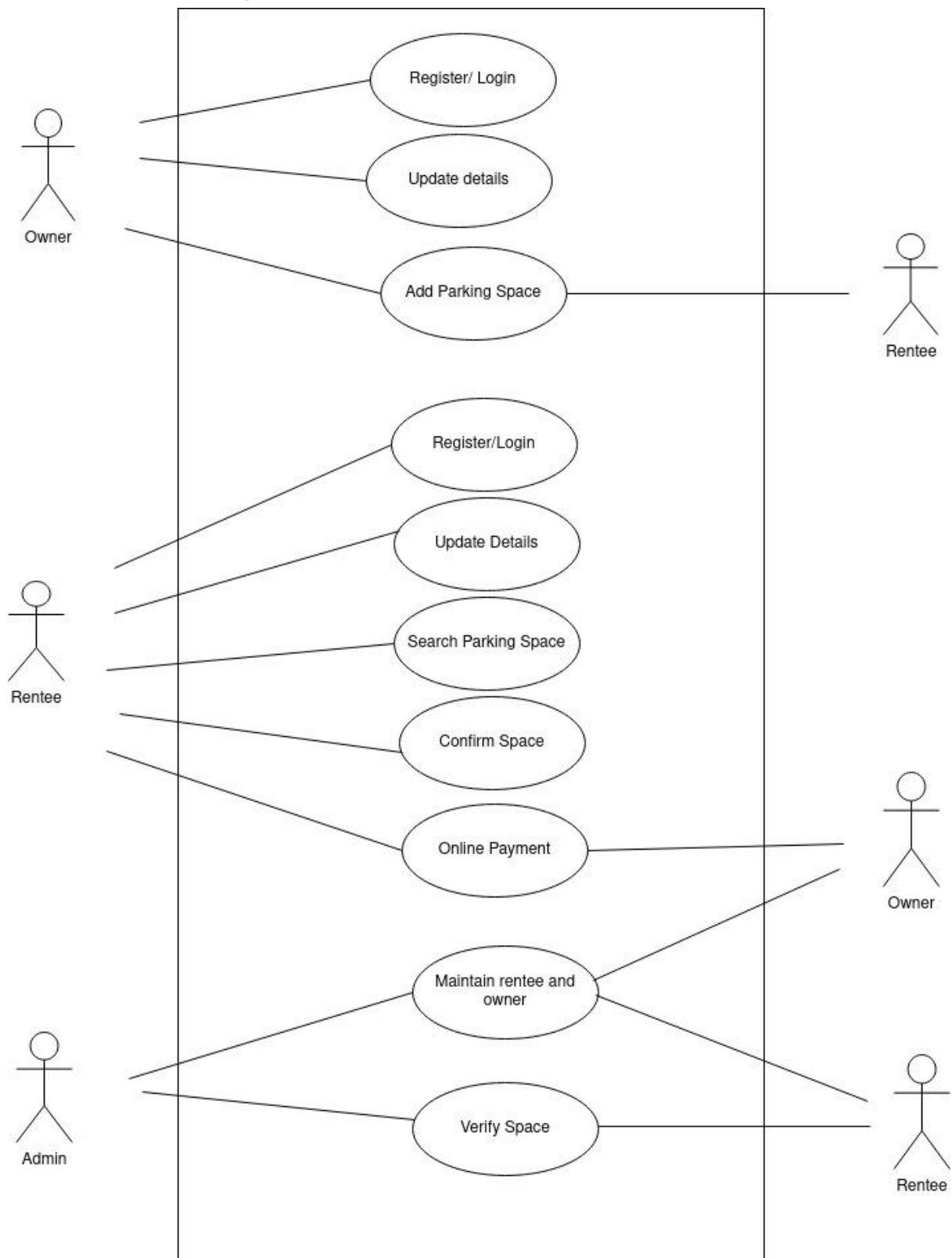


Level 1

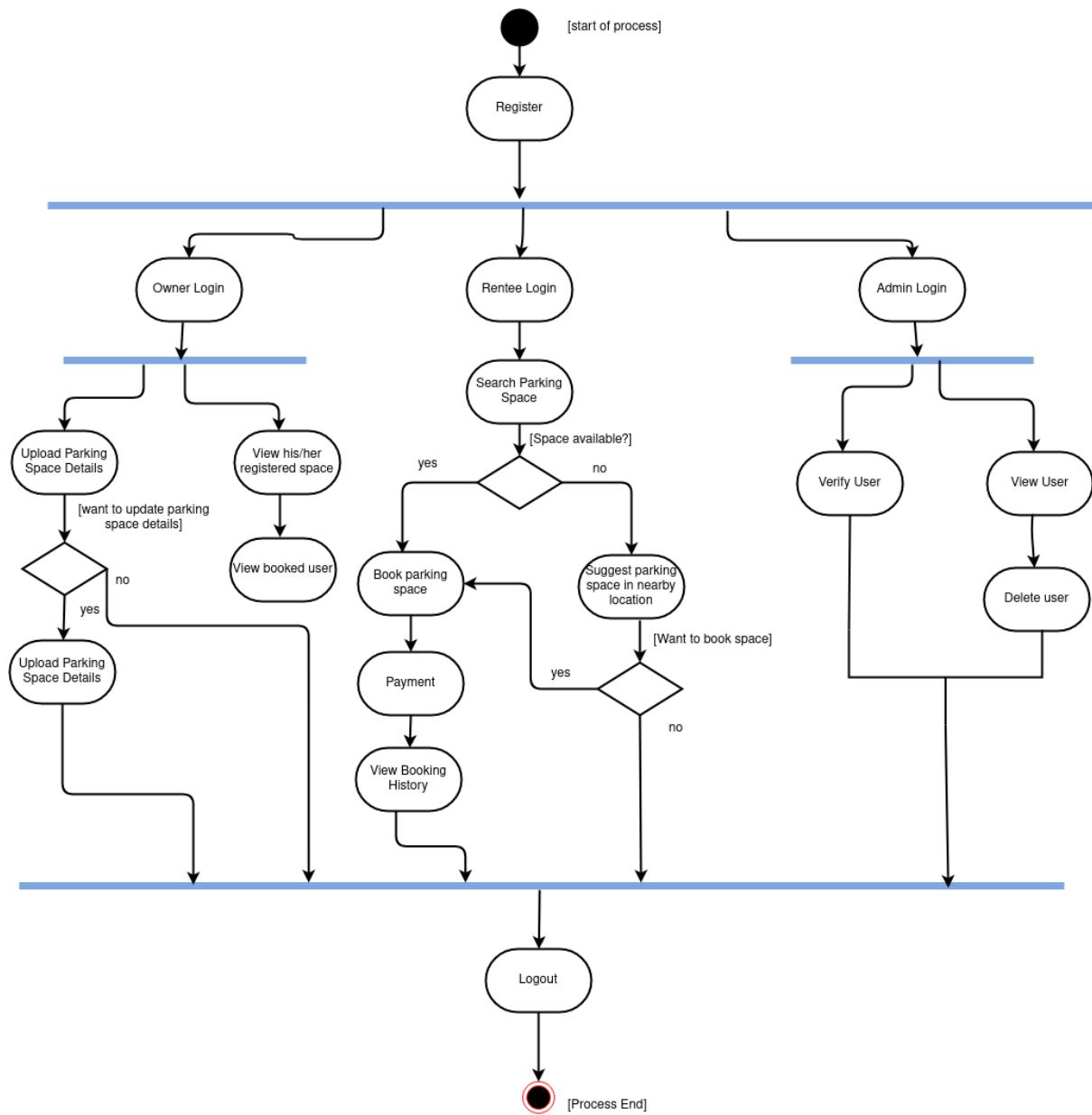


Level 2

7.2 Use Case Diagram

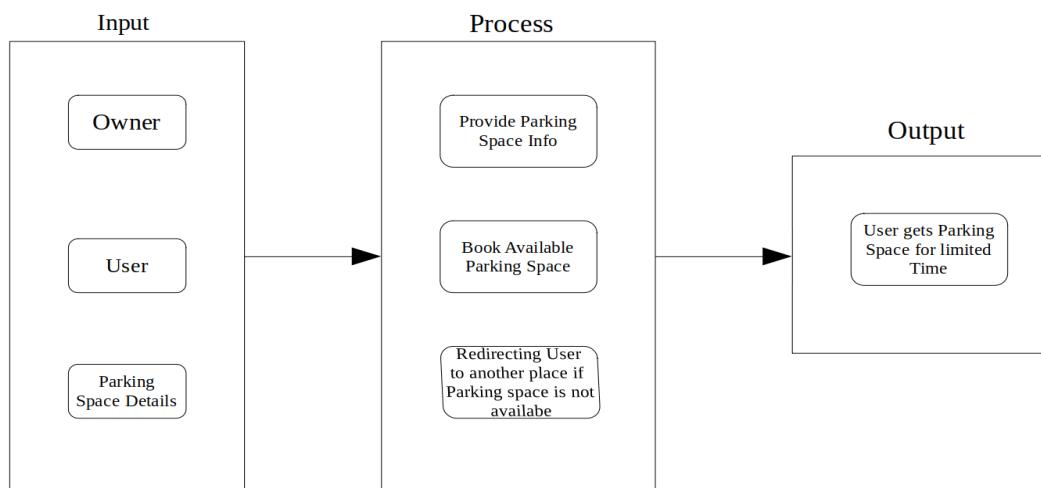


7.3 Activity Diagram

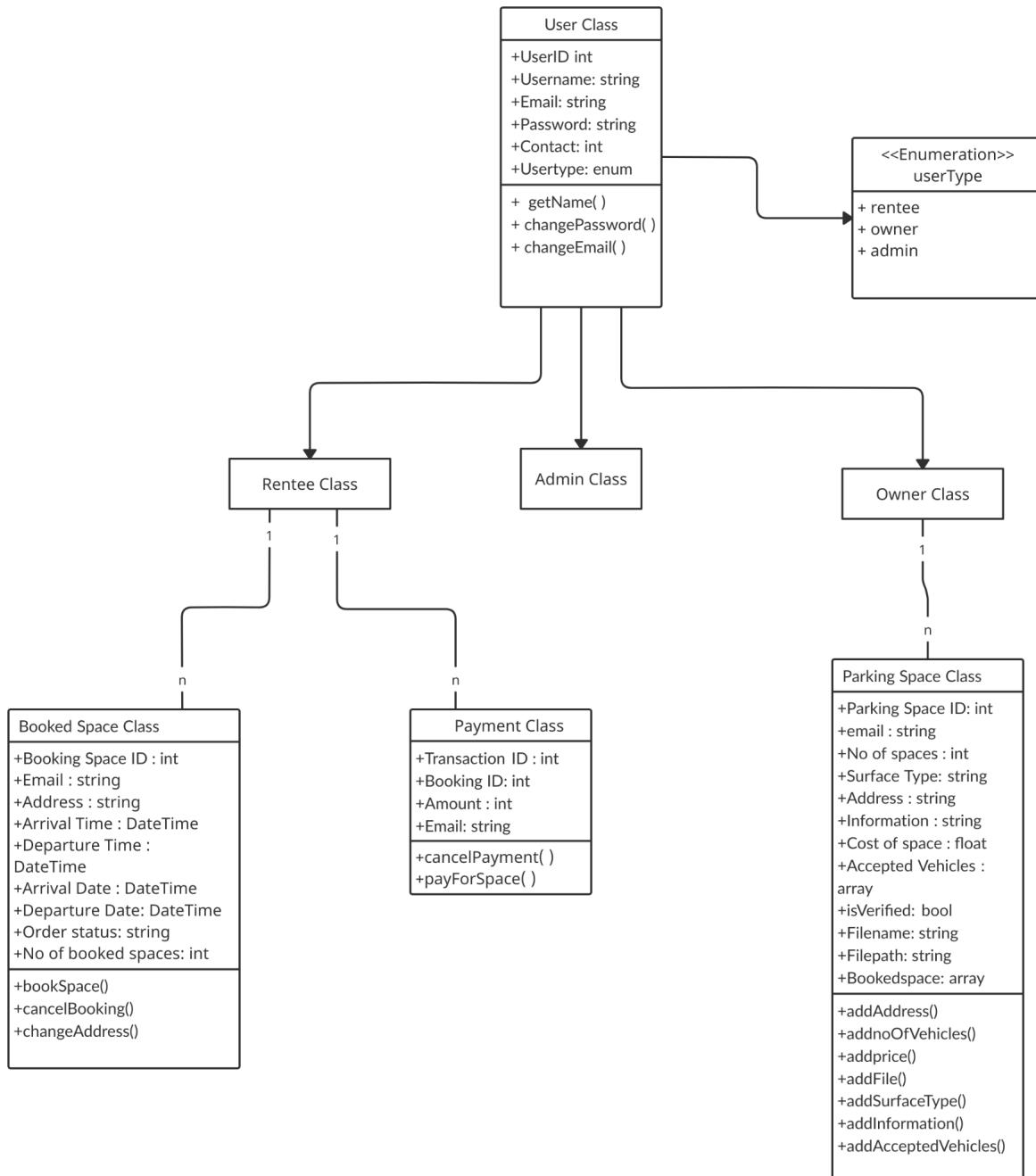


7.3 System Architecture

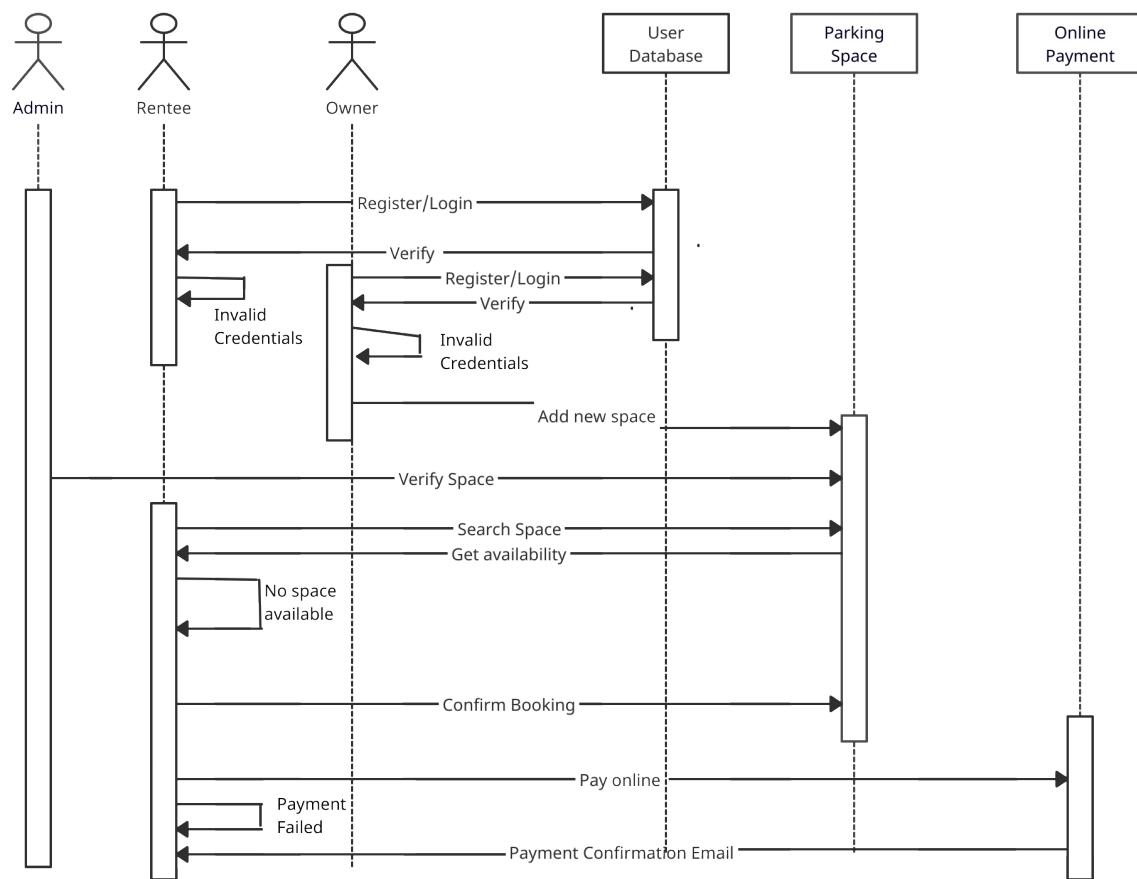
Architecture Diagram



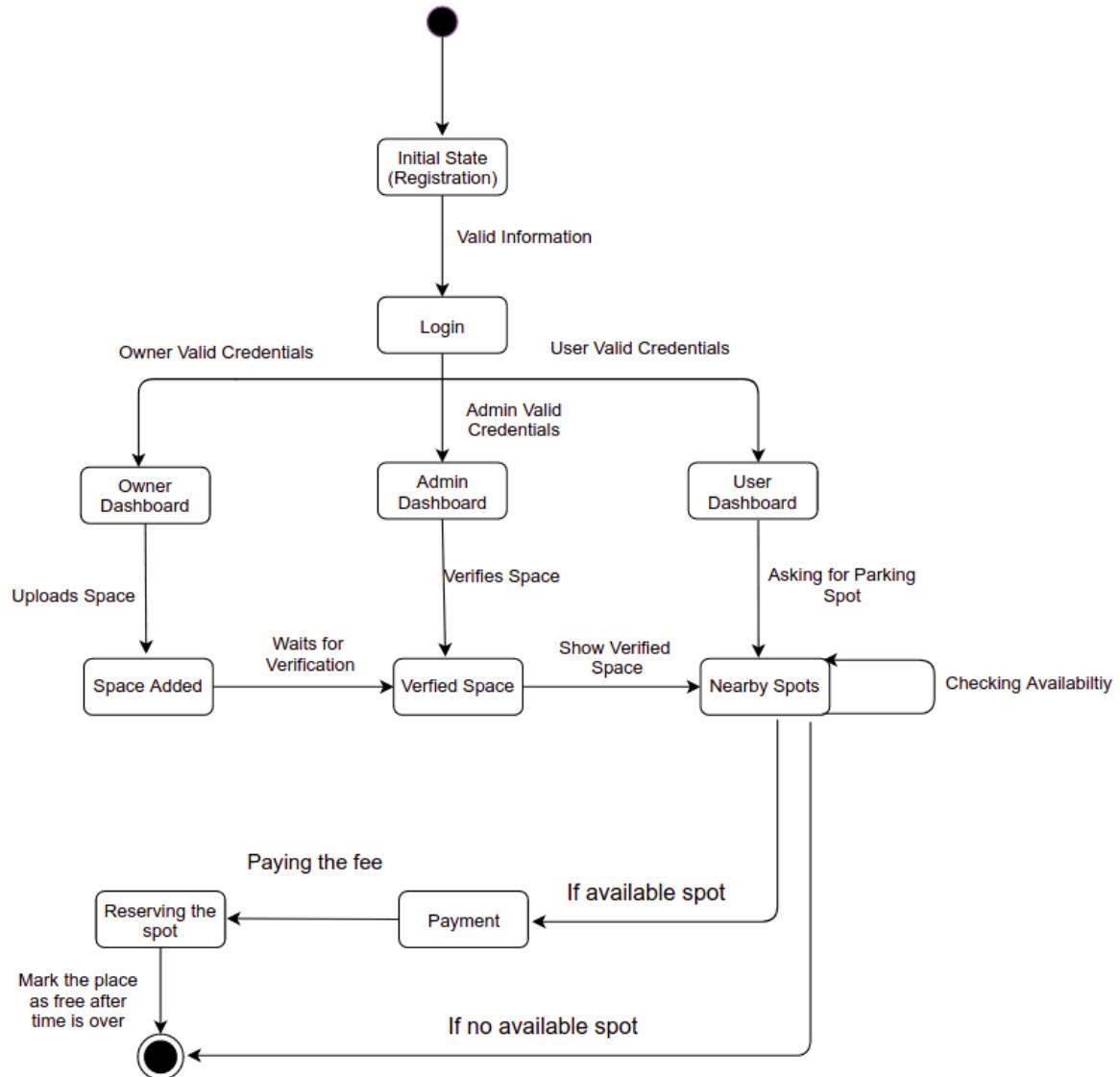
7.4 Class Diagram



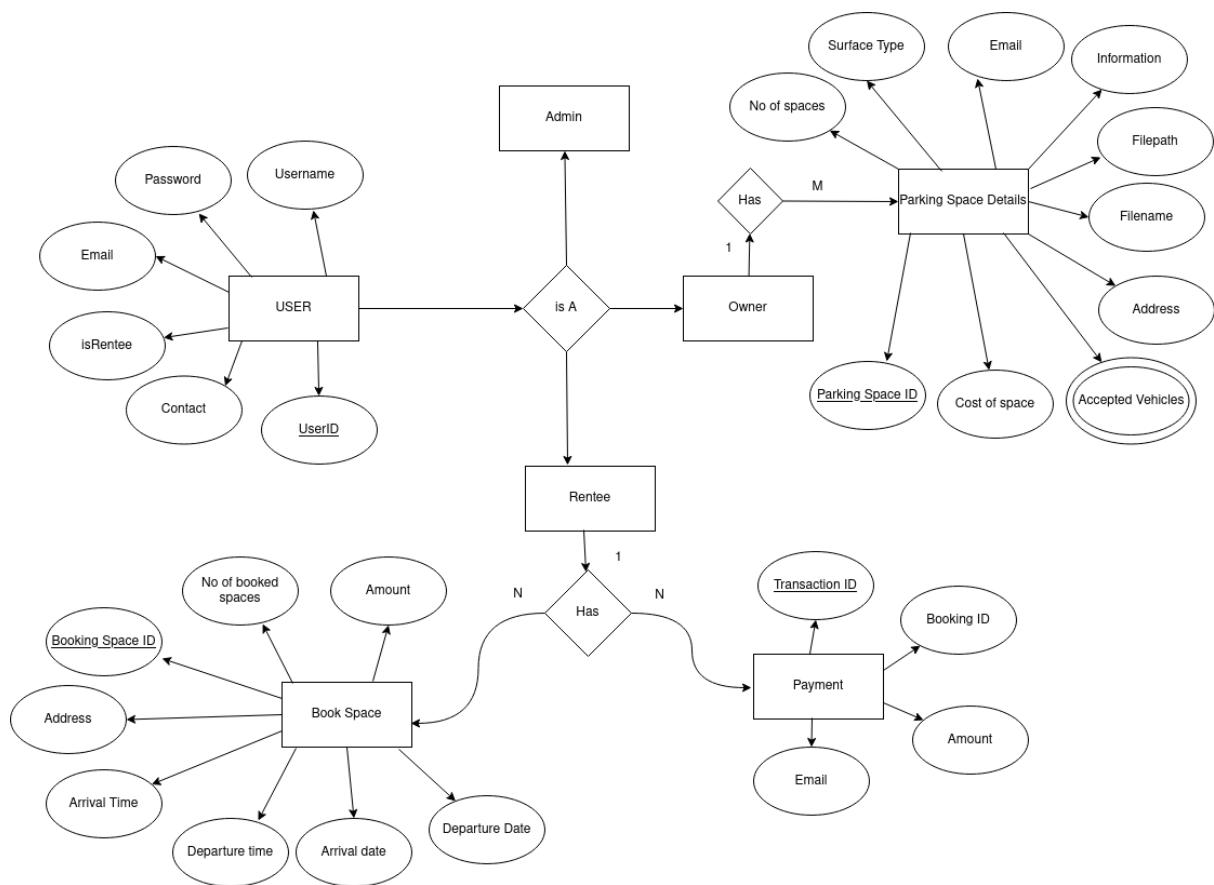
7.5 Sequence Diagram



7.6 State Diagram



7.7 ER Diagram



8.Implementation

The image consists of two screenshots of a web application named "Parky".

The top screenshot shows the "Renter Registration" page. The URL in the browser is "localhost:3000/userregister". The page has a light blue header with the title "Renter Registration". It contains input fields for "Name" (Riddhi Prakash Ghate), "Email" (Riddhitestuser@gmail.com), "Phone Number" (9604237506), "Password" (*****), and "Confirm Password" (*****). Below the password fields, there is a green text "matching". At the bottom is a black "Register" button. A link "Already have an Account? Sign In" is at the bottom right.

The bottom screenshot shows the homepage. The URL in the browser is "localhost:3000/searchspace". The header includes the "Parky" logo, the user's name "Riddhi Prakash Ghate", and a "Logout" link. The main content features a purple banner with the text "PARKING JUST GOT A LOT SIMPLER" and "Book the Best Spaces and Save up to 20% off". Below the banner is a search bar with the placeholder "Where do You want to Park?" and a dropdown menu showing "Shivajinagar, Pune, Maharashtra, India". A blue button labeled "Search Your Parking Space" is below the search bar. At the bottom, there is a section titled "How Parky Works?" with a sub-section titled "LOOK" featuring an icon of a smartphone displaying a parking app interface.

Parky

The image shows two screenshots of the Parky application interface. The top screenshot displays a map of Pune city center, specifically the KASBA PETH area, with various landmarks like Shivaji Nagar, SPHURTI SOCIETY, and RAIJGANDHI NAGAR SLUM. A red marker indicates a parking location. The bottom screenshot shows a detailed view of the Kothrud area, listing several parking spots such as Jangali Maharaj Rd, Sutar Hospital, and Deeyoni Multi Speciality Hospital. Each listing includes details like surface type, number of spaces, accepted vehicles, price, and additional info, along with a 'Book Space' button.

Jangali Maharaj Rd, Shivajinagar, Pune, Maharashtra, India
Surface Type: Partially Covered
Number of Spaces: 3
Accepted Vehicles: Car
Price: 30
Additional Info: vdvd
[Book Space](#)

Kothrud Bus Stand Rd, Rahul Nagar, Kothrud, Pune, Maharashtra 411038, India
Surface Type: Underground
Number of Spaces: 4
Accepted Vehicles: CarMotorCycle
Price: 40
Additional Info:
[Book Space](#)

Parky

The screenshot shows the Parky Checkout page. At the top, it displays the address: "Kothrud Bus Stand Rd, Rahul Nagar, Kothrud, Pune, Maharashtra 411038, India". It indicates "Total No of Spaces: 4", "Surface Type: Underground", "Accepted Vehicles: CarMotorCycle", and "Price per hour: 40". Below this, there are dropdown menus for "Select Arrival Date" (set to 04/30/2021), "Select Departure Date" (set to 04/30/2021), and "Select Arrival Time" (set to 15:14). There are also dropdown menus for "Select Departure Time" (set to 17:18) and "No of Spaces required" (set to 1). A "Proceed to Pay" button is at the bottom.

The screenshot shows the Parky Checkout page with a payment modal overlay. The modal title is "Parky Booking" and the amount is ₹ 81. The language is set to English. The modal lists several payment methods: "Card" (Visa, MasterCard, RuPay, and Maestro), "UPI / QR" (Instant payment using UPI App), "Netbanking" (All Indian banks), and "Wallet" (MobiKwik & More). The background of the page shows the same basic details as the first screenshot. A red ribbon banner in the top right corner of the page says "Test Mode".

The screenshot shows a web browser window titled "Parky" with the URL "localhost:3000/bookinghistory". The page has a dark header with the title "Parky" and a search bar. Below the header is a table titled "Booking Details" showing four completed bookings. Each booking row contains the Order No., Address, Order Amount, Booked Date and Time, Order Status, Directions, and Options buttons.

Order No	Address	Order Amount	Booked Date and Time	Order Status	Directions	Options
1	Kothrud Bus Stand Rd, Rahul Nagar, Kothrud, Pune, Maharashtra 411038, India	81	2021-04-30 15:14 - 2021-04-30 17:16	Completed	Get Directions	Get Receipt
2	Jangali Maharaj Rd, Shivajinagar, Pune, Maharashtra, India	67	2021-04-27 14:20 - 2021-04-27 16:35	Completed	Get Directions	Get Receipt
3	Opp COEP Hostel, Wellesley Road, Shivajinagar, Shivajinagar, Pune, Maharashtra 411005, India	34	2021-04-27 15:05 - 2021-04-27 16:05	Completed	Get Directions	Get Receipt
4	Fergusson College Rd, Shivajinagar, Pune, Maharashtra 411004, India	40	2021-04-26 14:00 - 2021-04-26 16:00	Completed	Get Directions	Get Receipt

Add Parking Space

Address: S.No, 120/A-2, Alandi Rd, Ram Society, Yerawada, Pune, Maharashtra 411006, India

Surface Type: Covered

Accepted Vehicles:

- Car
- MotorCycle
- Bus
- Truck
- MiniBus/ Van

Frontpage+Content Page (1).docx

Submit

#	Address	Price per Hour	Number of Parking Space	Surface Type	Status	Document	Booking Details	Edit Space Info	Delete Space
1	Pratap Nagar, Nagpur, Maharashtra, India	20 Rs	2		Verified	Pratap Nagar, Nagpur, Maharashtra, India.pdf	View Booking Details	Edit Space	Delete Space
2	Shivaji Nagar, Bengaluru, Karnataka, India	30 Rs	3	Covered	Not Verified		View Booking Details	Edit Space	Delete Space

Manage User Details

Manage the details of rentees and owners here.

VERIFY

Verify Spaces

You can verify the space details uploaded by the owner.

Manage Users

Verify Space

#	Name	Email	Contact	User Type	Edit	Delete
1	Tushar Ghate	tushar@gmail.com	9834556233	owner	Edit Details	Delete User
2	Manish	pms.parking24by7@gmail.com	7249335682	admin	Edit Details	Delete User
3	Riddhi Prakash Ghate	newuser@gmail.com	9604237506	user	Edit Details	Delete User
4	Andy	andyjonas@gmail.com	9999999999	user	Edit Details	Delete User
5	Riddhi Prakash Ghate	newowner@gmail.com	9604237506	owner	Edit Details	Delete User
6	Saksh	saksh@gmail.com	9999999999	owner	Edit Details	Delete User
7	Sakshi Kalekar	kalekarsakshi2000@gmail.com	9999999999	user	Edit Details	Delete User
8	Saksh	kalekarss18.comp@coep.ac.in	9561303330	owner	Edit Details	Delete User

The screenshot shows a web browser window titled "Parky" with the URL "localhost:3000/showspaces". The page displays a table titled "Spaces" containing four rows of data. Each row represents a parking space with columns for ID, Email, Price, Address, Surface Type, Number of Spaces, Accepted Vehicles, Verification Status, Document, Verify button, and Delete Space button.

#	Email	Price	Address	Surface Type	Number of Spaces	Accepted Vehicles	Verification Status	Document	Verify	Delete Space
1	tushar@gmail.com	30	Shivaji Nagar, Bengaluru, Karnataka, India	Covered	3		Not Verified		<button>Verify Space</button>	<button>Delete Space</button>
2	tushar@gmail.com	20	Pratap Nagar, Nagpur, Maharashtra, India		2	Car	Verified	Pratap Nagar, Nagpur, Maharashtra, India.pdf	<button>Verify Space</button>	<button>Delete Space</button>
3	newowner@gmail.com	30	Jangali Maharaj Rd, Shivajinagar, Pune, Maharashtra, India	Partially Covered	3	Car	Verified	Jangali Maharaj Rd, Shivajinagar, Pune, Maharashtra, India.pdf	<button>Verify Space</button>	<button>Delete Space</button>
4	newowner@gmail.com	40	Fergusson College Rd,	Covered	0	Car	Verified	Pratap Nagar, Nagpur,	<button>Verify Space</button>	<button>Delete Space</button>

Parky

The image shows two instances of Visual Studio Code side-by-side, both displaying code for a project named "Parky".

AddParkingSpace.js - Parky - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
OPEN EDITORS
  x JS AddParkingSpace.js front...
  x JS AddParkingSpace.js front...
PARKY
  > backend
  > frontend
    > node_modules
    > public
    > src
      > Components
        > AddParkingSpace.js
        > AdminEditProfile.js
        > AdminPage.js
        > BookingHistory.js
        > BookSpace.js
        > EditParkingSpace.js
        > EditProfile.js
        > Footer.js
        > GoogleMap.js
        > Homepage.js
        > Login.js
        > MySpace.js
        > Navigationbar.js
        > NotFoundPage.js
        > OwnerRegister.js
        > Payment.js
        > Profile.js
        > ResetPassword.js
        > SearchSpace.js
        > ShowSpace.js
        > ShowUsers.js
        > UserBookingDetails.js
        > UserRegister.js
        > VerifySpace.js
      > CSS
      > App.js
      > App.test.js
      > FormValidation.js
      > Index.js
      > OUTLINE
      > TIMELINE
      100
      101
      102
      103
      104
      105
      106
      107
      108
      109
      110
      111
      112
      113
      114
      115
      116
      117
      118
      119
      120
      121
      122
      123
      124
      125
      126
      127
      128
      129
      130
      131
      132
      133
      134
      135
      136
      137
      138
      139
      140
      141
      142
      143
      144
      145
      146
      147
      148
      149
      150
      151
      152
      153
      154
      155
      156
      157
      158
      159
      160
      161
      162
      163
      164
      165
      166
      167
      168
      169
      170
      171
      172
      173
      174
      175
      176
      177
      178
      179
      180
      181
      182
      183
      184
      185
      186
      187
      188
      189
      190
      191
      192
      193
      194
      195
      196
      197
      198
      199
      200
      201
      202
      203
      204
      205
      206
      207
      208
      209
      210
      211
      212
      213
      214
      215
      216
      217
      218
      219
      220
      221
      222
      223
      224
      225
      226
      227
      228
      229
      230
      231
      232
      233
      234
      235
      236
      237
      238
      239
      240
      241
      242
      243
      244
      245
      246
      247
      248
      249
      250
      251
      252
      253
      254
      255
      256
      257
      258
      259
      260
      261
      262
      263
      264
      265
      266
      267
      268
      269
      270
      271
      272
      273
      274
      275
      276
      277
      278
      279
      280
      281
      282
      283
      284
      285
      286
      287
      288
      289
      290
      291
      292
      293
      294
      295
      296
      297
      298
      299
      300
      301
      302
      303
      304
      305
      306
      307
      308
      309
      310
      311
      312
      313
      314
      315
      316
      317
      318
      319
      320
      321
      322
      323
      324
      325
      326
      327
      328
      329
      330
      331
      332
      333
      334
      335
      336
      337
      338
      339
      340
      341
      342
      343
      344
      345
      346
      347
      348
      349
      350
      351
      352
      353
      354
      355
      356
      357
      358
      359
      360
      361
      362
      363
      364
      365
      366
      367
      368
      369
      370
      371
      372
      373
      374
      375
      376
      377
      378
      379
      380
      381
      382
      383
      384
      385
      386
      387
      388
      389
      390
      391
      392
      393
      394
      395
      396
      397
      398
      399
      400
      401
      402
      403
      404
      405
      406
      407
      408
      409
      410
      411
      412
      413
      414
      415
      416
      417
      418
      419
      420
      421
      422
      423
      424
      425
      426
      427
      428
      429
      430
      431
      432
      433
      434
      435
      436
      437
      438
      439
      440
      441
      442
      443
      444
      445
      446
      447
      448
      449
      450
      451
      452
      453
      454
      455
      456
      457
      458
      459
      460
      461
      462
      463
      464
      465
      466
      467
      468
      469
      470
      471
      472
      473
      474
      475
      476
      477
      478
      479
      480
      481
      482
      483
      484
      485
      486
      487
      488
      489
      490
      491
      492
      493
      494
      495
      496
      497
      498
      499
      500
      501
      502
      503
      504
      505
      506
      507
      508
      509
      510
      511
      512
      513
      514
      515
      516
      517
      518
      519
      520
      521
      522
      523
      524
      525
      526
      527
      528
      529
      530
      531
      532
      533
      534
      535
      536
      537
      538
      539
      540
      541
      542
      543
      544
      545
      546
      547
      548
      549
      550
      551
      552
      553
      554
      555
      556
      557
      558
      559
      560
      561
      562
      563
      564
      565
      566
      567
      568
      569
      570
      571
      572
      573
      574
      575
      576
      577
      578
      579
      580
      581
      582
      583
      584
      585
      586
      587
      588
      589
      590
      591
      592
      593
      594
      595
      596
      597
      598
      599
      600
      601
      602
      603
      604
      605
      606
      607
      608
      609
      610
      611
      612
      613
      614
      615
      616
      617
      618
      619
      620
      621
      622
      623
      624
      625
      626
      627
      628
      629
      630
      631
      632
      633
      634
      635
      636
      637
      638
      639
      640
      641
      642
      643
      644
      645
      646
      647
      648
      649
      650
      651
      652
      653
      654
      655
      656
      657
      658
      659
      660
      661
      662
      663
      664
      665
      666
      667
      668
      669
      670
      671
      672
      673
      674
      675
      676
      677
      678
      679
      680
      681
      682
      683
      684
      685
      686
      687
      688
      689
      690
      691
      692
      693
      694
      695
      696
      697
      698
      699
      700
      701
      702
      703
      704
      705
      706
      707
      708
      709
      710
      711
      712
      713
      714
      715
      716
      717
      718
      719
      720
      721
      722
      723
      724
      725
      726
      727
      728
      729
      730
      731
      732
      733
      734
      735
      736
      737
      738
      739
      740
      741
      742
      743
      744
      745
      746
      747
      748
      749
      750
      751
      752
      753
      754
      755
      756
      757
      758
      759
      759
      760
      761
      762
      763
      764
      765
      766
      767
      768
      769
      770
      771
      772
      773
      774
      775
      776
      777
      778
      779
      779
      780
      781
      782
      783
      784
      785
      786
      787
      788
      789
      789
      790
      791
      792
      793
      794
      795
      796
      797
      798
      799
      799
      800
      801
      802
      803
      804
      805
      806
      807
      808
      809
      809
      810
      811
      812
      813
      814
      815
      816
      817
      818
      819
      819
      820
      821
      822
      823
      824
      825
      826
      827
      828
      829
      829
      830
      831
      832
      833
      834
      835
      836
      837
      838
      839
      839
      840
      841
      842
      843
      844
      845
      846
      847
      848
      849
      849
      850
      851
      852
      853
      854
      855
      856
      857
      858
      859
      859
      860
      861
      862
      863
      864
      865
      866
      867
      868
      869
      869
      870
      871
      872
      873
      874
      875
      876
      877
      878
      879
      879
      880
      881
      882
      883
      884
      885
      886
      887
      888
      889
      889
      890
      891
      892
      893
      894
      895
      896
      897
      898
      899
      899
      900
      901
      902
      903
      904
      905
      906
      907
      908
      909
      909
      910
      911
      912
      913
      914
      915
      916
      917
      918
      919
      919
      920
      921
      922
      923
      924
      925
      926
      927
      928
      929
      929
      930
      931
      932
      933
      934
      935
      936
      937
      938
      939
      939
      940
      941
      942
      943
      944
      945
      946
      947
      948
      949
      949
      950
      951
      952
      953
      954
      955
      956
      957
      958
      959
      959
      960
      961
      962
      963
      964
      965
      966
      967
      968
      969
      969
      970
      971
      972
      973
      974
      975
      976
      977
      978
      979
      979
      980
      981
      982
      983
      984
      985
      986
      987
      988
      989
      989
      990
      991
      992
      993
      994
      995
      996
      997
      998
      999
      999
      1000
      1001
      1002
      1003
      1004
      1005
      1006
      1007
      1008
      1009
      1009
      1010
      1011
      1012
      1013
      1014
      1015
      1016
      1017
      1018
      1019
      1019
      1020
      1021
      1022
      1023
      1024
      1025
      1026
      1027
      1028
      1029
      1029
      1030
      1031
      1032
      1033
      1034
      1035
      1036
      1037
      1038
      1039
      1039
      1040
      1041
      1042
      1043
      1044
      1045
      1046
      1047
      1048
      1049
      1049
      1050
      1051
      1052
      1053
      1054
      1055
      1056
      1057
      1058
      1059
      1059
      1060
      1061
      1062
      1063
      1064
      1065
      1066
      1067
      1068
      1069
      1069
      1070
      1071
      1072
      1073
      1074
      1075
      1076
      1077
      1078
      1078
      1079
      1080
      1081
      1082
      1083
      1084
      1085
      1086
      1087
      1088
      1088
      1089
      1090
      1091
      1092
      1093
      1094
      1095
      1096
      1097
      1098
      1098
      1099
      1099
      1100
      1101
      1102
      1103
      1104
      1105
      1106
      1107
      1108
      1109
      1109
      1110
      1111
      1112
      1113
      1114
      1115
      1116
      1117
      1118
      1119
      1119
      1120
      1121
      1122
      1123
      1124
      1125
      1126
      1127
      1128
      1129
      1129
      1130
      1131
      1132
      1133
      1134
      1135
      1136
      1137
      1138
      1138
      1139
      1140
      1141
      1142
      1143
      1144
      1145
      1146
      1147
      1148
      1148
      1149
      1150
      1151
      1152
      1153
      1154
      1155
      1156
      1157
      1158
      1158
      1159
      1160
      1161
      1162
      1163
      1164
      1165
      1166
      1167
      1168
      1169
      1169
      1170
      1171
      1172
      1173
      1174
      1175
      1176
      1177
      1178
      1178
      1179
      1180
      1181
      1182
      1183
      1184
      1185
      1186
      1187
      1187
      1188
      1189
      1190
      1191
      1192
      1193
      1194
      1195
      1196
      1197
      1197
      1198
      1199
      1199
      1200
      1201
      1202
      1203
      1204
      1205
      1206
      1207
      1208
      1208
      1209
      1210
      1211
      1212
      1213
      1214
      1215
      1216
      1217
      1218
      1218
      1219
      1220
      1221
      1222
      1223
      1224
      1225
      1226
      1227
      1228
      1228
      1229
      1230
      1231
      1232
      1233
      1234
      1235
      1236
      1237
      1237
      1238
      1239
      1240
      1241
      1242
      1243
      1244
      1245
      1246
      1247
      1247
      1248
      1249
      1250
      1251
      1252
      1253
      1254
      1255
      1256
      1257
      1258
      1258
      1259
      1260
      1261
      1262
      1263
      1264
      1265
      1266
      1267
      1268
      1268
      1269
      1270
      1271
      1272
      1273
      1274
      1275
      1276
      1277
      1277
      1278
      1279
      1280
      1281
      1282
      1283
      1284
      1285
      1286
      1287
      1287
      1288
      1289
      1290
      1291
      1292
      1293
      1294
      1295
      1296
      1297
      1297
      1298
      1299
      1299
      1300
      1301
      1302
      1303
      1304
      1305
      1306
      1307
      1308
      1308
      1309
      1310
      1311
      1312
      1313
      1314
      1315
      1316
      1317
      1317
      1318
      1319
      1320
      1321
      1322
      1323
      1324
      1325
      1326
      1327
      1327
      1328
      1329
      1330
      1331
      1332
      1333
      1334
      1335
      1336
      1337
      1337
      1338
      1339
      1340
      1341
      1342
      1343
      1344
      1345
      1346
      1347
      1347
      1348
      1349
      1350
      1351
      1352
      1353
      1354
      1355
      1356
      1357
      1358
      1358
      1359
      1360
      1361
      1362
      1363
      1364
      1365
      1366
      1367
      1367
      1368
      1369
      1370
      1371
      1372
      1373
      1374
      1375
      1376
      1377
      1377
      1378
      1379
      1380
      1381
      1382
      1383
      1384
      1385
      1386
      1387
      1387
      1388
      1389
      1390
      1391
      1392
      1393
      1394
      1395
      1396
      1397
      1397
      1398
      1399
      1399
      1400
      1401
      1402
      1403
      1404
      1405
      1406
      1407
      1408
      1408
      1409
      1410
      1411
      1412
      1413
      1414
      1415
      1416
      1417
      1417
      1418
      1419
      1420
      1421
      1422
      1423
      1424
      1425
      1426
      1427
      1427
      1428
      1429
      1430
      1431
      1432
      1433
      1434
      1435
      1436
      1437
      1437
      1438
      1439
      1440
      1441
      1442
      1443
      1444
      1445
      1446
      1447
      1447
      1448
      1449
      1450
      1451
      1452
      1453
      1454
      1455
      1456
      1457
      1458
      1458
      1459
      1460
      1461
      1462
      1463
      1464
      1465
      1466
      1467
      1468
      1468
      1469
      1470
      1471
      1472
      1473
      1474
      1475
      1476
      1477
      1477
      1478
      1479
      1480
      1481
      1482
      1483
      1484
      1485
      1486
      1487
      1487
      1488
      1489
      1490
      1491
      1492
      1493
      1494
      1495
      1496
      1497
      1497
      1498
      1499
      1499
      1500
      1501
      1502
      1503
      1504
      1505
      1506
      1507
      1508
      1508
      1509
      1510
      1511
      1512
      1513
      1514
      1515
      1516
      1517
      1517
      1518
      1519
      1520
      1521
      1522
      1523
      1524
      1525
      1526
      1527
      1527
      1528
      1529
      1530
      1531
      1532
      1533
      1534
      1535
      1536
      1537
      1537
      1538
      1539
      1540
      1541
      1542
      1543
      1544
      1545
      1546
      1547
      1547
      1548
      1549
      1550
      1551
      1552
      1553
      1554
      1555
      1556
      1557
      1558
      1558
      1559
      1560
      1561
      1562
      1563
      1564
      1565
      1566
      1567
      1567
      1568
      1569
      1570
      1571
      1572
      1573
      1574
      1575
      1576
      1577
      1577
      1578
      1579
      1580
      1581
      1582
      1583
      1584
      1585
      1586
      1587
      1587
      1588
      1589
      1590
      1591
      1592
      1593
      1594
      1595
      1596
      1597
      1598
      1598
      1599
      1599
      1600
      1601
      1602
      1603
      1604
      1605
      1606
      1607
      1608
      1608
      1609
      1610
      1611
      1612
      1613
      1614
      1615
      1616
      1617
      1617
      1618
      1619
      1620
      1621
      1622
      1623
      1624
      1625
      1626
      1627
      1627
      1628
      1629
      1630
      1631
      1632
      1633
      1634
      1635
      1636
      1637
      1637
      1638
      1639
      1640
      1641
      1642
      1643
      1644
      1645
      1646
      1647
      1647
      1648
      1649
      1650
      1651
      1652
      1653
      1654
      1655
      1656
      1657
      1658
      1658
      1659
      1660
      1661
      1662
      1663
      1664
      1665
      1666
      1667
      1668
     
```


Parky

The image shows two instances of Visual Studio Code side-by-side, both displaying code for a project named "Parky".

GoogleMap.js - Parky - Visual Studio Code

```
function createMarker(place) {
  if (!place.geometry || !place.geometry.location) return;
  const marker = new google.maps.Marker({
    map,
    position: place.geometry.location,
  });
  addInfoWindow(marker, place.name);
}

function addInfoWindow(marker, content) {
  var infoWindow = new google.maps.InfoWindow({
    content: content
  });
  google.maps.event.addListener(marker, 'click', function () {
    infoWindow.open(map, marker);
  });
}

function customMarker(place, details) {
  const icons = {
    parking: {
      icon: 'parking_lot.png',
    },
  };
  const content = '<p>Address: ' + details.address + '</p>' +
    '<p>Accepted Vehicles: ' + details.accepted_vehicles + '</p>' +
    '<p>Number of spaces: ' + details.spacesnumber + '</p>' +
    '<p>Price per hour: ' + details.price + '</p>' +
    '<p>Site Type: ' + details.sitetype + '</p>' +
    '<p>Additional Info: ' + details.info + '</p>';
  const marker = new google.maps.Marker({
    position: place.geometry.location,
    icon: {
      size: new google.maps.Size(80, 80),
      scaledSize: new google.maps.Size(80, 80),
      url: icons['parking'].icon
    },
    map: map,
  });
  addInfoWindow(marker, content);
}
```

Payment.js - Parky - Visual Studio Code

```
import React from 'react';
import Bookspacecss from './CSS/BookSpace.module.css';
import axios from 'axios';
import { toast, ToastContainer } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';

function loadScript(src) {
  return new Promise((resolve) => {
    const script = document.createElement('script');
    script.src = src;
    script.onload = () => {
      resolve(true)
    }
    script.onerror = () => {
      resolve(false)
    }
    document.body.appendChild(script)
  })
}

function savePaymentDetails(data) {
  let payment = {
    transaction_id: '',
    amount: '',
    email: '',
    booking_id: ''
  }
  payment_email = window.sessionStorage.getItem("useremail")
  payment.amount = data.amount
  payment.transaction_id = data.id
  payment.booking_id = window.sessionStorage.getItem("booking_id")
  axios.post(process.env.REACT_APP_BACKEND + '/savepaymentdetails', payment)
  .then(
    res => [
      console.log("saved")
    ]
  )
  .catch(
    err => [
      console.log(err)
    ]
  )
}

function Payment(props) {
```

Parky

The image shows two instances of Visual Studio Code side-by-side, both displaying code files for a project named "Parky".

Payment.js - Parky - Visual Studio Code

File Edit Selection View Go Run Terminal Help

OPEN EDITORS

frontend > src > Components > Payment.js > savePaymentDetails > then() callback

```
function Payment(props) {  
    const displayRazorpay = () =>  
        const res = await loadScript('https://checkout.razorpay.com/v1/checkout.js')  
        if (!res) {  
            alert('Razorpay SDK failed to load. Are you online?')  
            return  
        }  
        const data = await fetch(process.env.REACT_APP_BACKEND + '/razorpay',  
            { method: 'POST',  
                headers: { 'Content-Type': 'application/json' },  
                body: JSON.stringify(t.json())  
        )  
        const arrival_time = props.parkinginfo.arrival_date + 'T' + props.parkinginfo.arrival_time  
        const departure_time = props.parkinginfo.departure_date + 'T' + props.parkinginfo.departure_time  
        const date1 = new Date(arrival_time)  
        const date2 = new Date(departure_time)  
        const minutesDifference = (date2 - date1) / (1000 * 60)  
        var Price = Math.floor((parseInt(props.parkinginfo.price) * parseInt(props.parkinginfo.no_of_booked_spaces)) * parseInt(minutesDifference)) / 60  
        data.amount = Price  
        savePaymentDetails(data)  
        const options = {  
            key: 'rzp_test_4z2vw67s30xv3b',  
            currency: data.currency,  
            amount: Price*100,  
            order_id: data.amount.id,  
            name: 'Parky Booking',  
            description: '',  
            handler: function(response) {  
                props.parkinginfo.price = Price  
                axios.post(process.env.REACT_APP_BACKEND + '/bookspace', props.parkinginfo)  
                .then(res => {  
                    if (res.data.error) {  
                        toast.error(res.data.error)  
                    }  
                    else {  
                        toast.success(res.data.msg)  
                        props.history.push('/bookinghistory')  
                    }  
                })  
            }  
        }  
        handler(response);  
    }  
}
```

Ln 34, Col 13 Tab Size 4 UTF-8 LF JavaScript ⚡

SearchSpace.js - Parky - Visual Studio Code

File Edit Selection View Go Run Terminal Help

OPEN EDITORS

frontend > src > Components > SearchSpace.js > loadScript > onreadystatechange

```
let autoComplete;  
const loadScript = (url, callback) => {  
    const script = document.createElement("script");  
    script.type = "text/javascript";  
    if (script.readyState) {  
        script.onreadystatechange = function () {  
            if (script.readyState === "loaded" || script.readyState === "complete") {  
                callback();  
            }  
        };  
    } else {  
        script.onload = () => callback();  
    }  
    script.src = url;  
    document.getElementsByTagName("head")[0].appendChild(script);  
};  
function handleScriptLoad(updateQuery, autoCompleteRef) {  
    autoComplete = new window.google.maps.places.Autocomplete(  
        autoCompleteRef.current,  
        {  
            fields: ["formatted_address", "geometry", "name"],  
            componentRestrictions: { country: "in" }  
        },  
        {  
            autoComplete.setFields(["address_components", "formatted_address"]);  
            autoComplete.addListener("place_changed", () =>  
                handlePlaceSelect(updateQuery)  
            );  
        }  
    );  
}  
async function handlePlaceSelect(updateQuery) {  
    const addressObject = autoComplete.getPlace();  
    const query = addressObject.formatted_address;  
    updateQuery(query);  
}  
function BookaSlot(props) {  
    const [query, setQuery] = useState("");  
    const autoCompleteRef = useRef(null);  
    const handleSubmit = () => {  
        // Handle form submission  
    };  
}
```

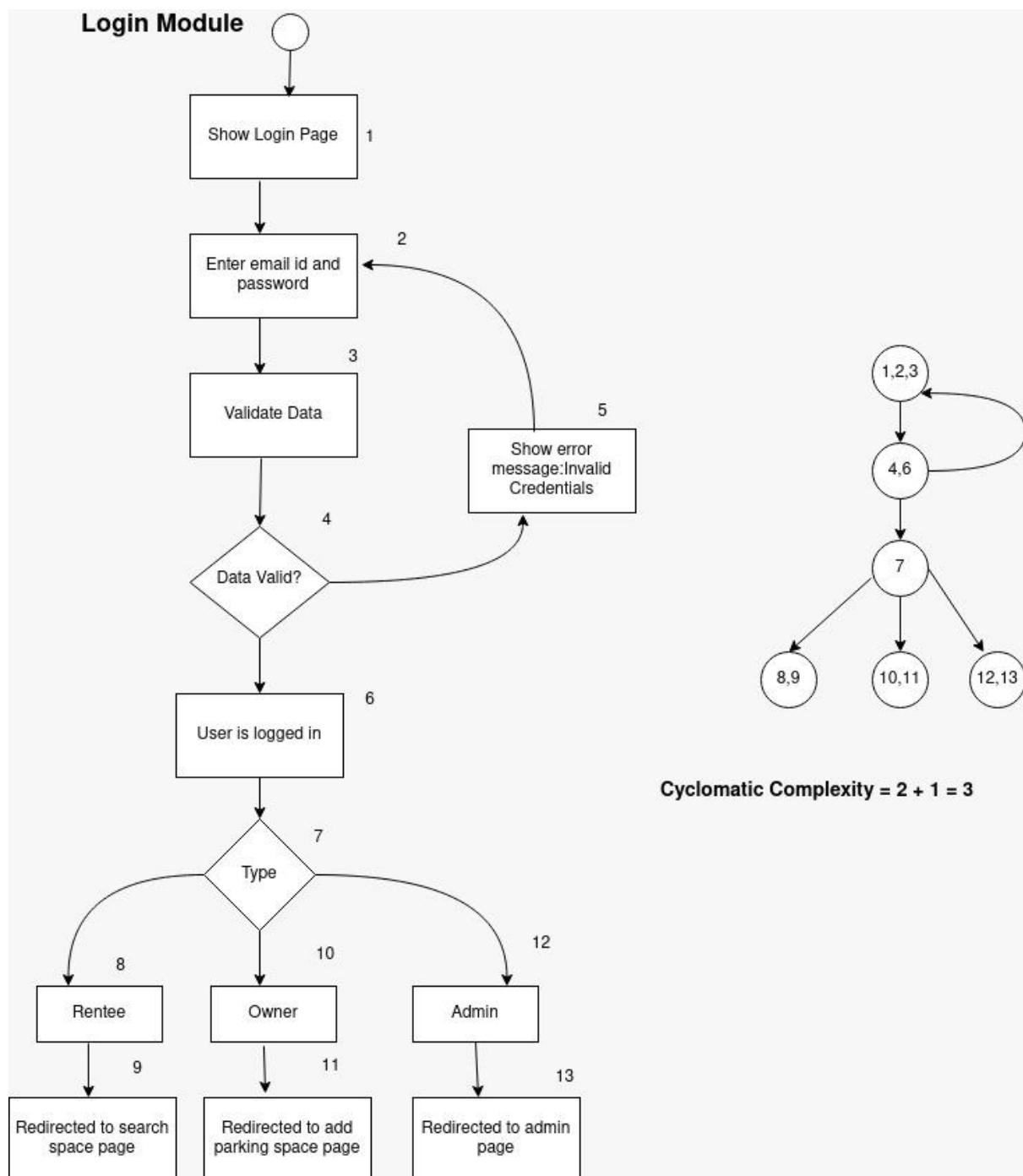
Ln 24, Col 8 Spaces: 2 UTF-8 LF JavaScript ⚡

9. Testing

1] 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.

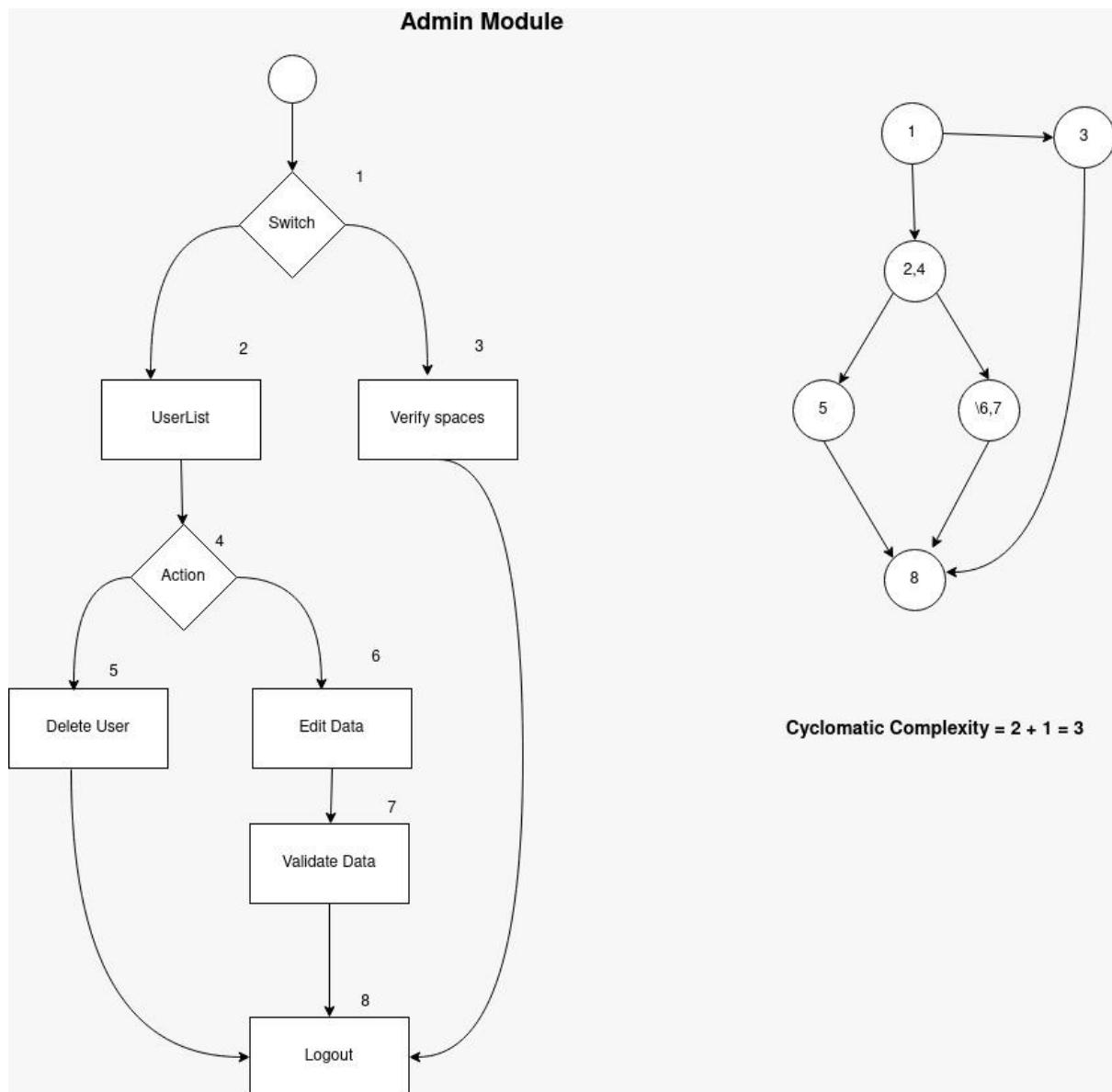
1] Login Module



Parky

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status
TS_PARKY_001	Verify User Login Functionality	TC_PARKY_UserLogin_001	Enter valid email, password	1. Enter valid email 2. Enter valid password 3. Click on login button	Valid URL test data	email: andyjonas@gmail.com password: AndyJonas@21	User should see search space page. Successfull login popup	Successfull login Redirected to search space page	Successfull login Redirected to search space page	PASS
TS_PARKY_001	Verify User Login Functionality	TC_PARKY_UserLogin_002	Enter valid email, password	1. Enter valid email 2. Enter valid password 3. Click on login button	Valid URL test data	email: andyjonas password: AndyJonas34	Error Message: Invalid credentials	Pop up appears showing error message: Invalid Credentials	Login button disabled	PASS
TS_PARKY_001	Verify Owner Login Functionality	TC_PARKY_OwnerLogin_003	Enter valid email, password	1. Enter valid email 2. Enter valid password 3. Click on login button	Valid URL test data	email: anna@gmail.com password: Anna@333	Owner should is redirected to AddParkingSpace page. Successfull login popup appears	Successfull login Redirected to add parking space page	Successfull login Redirected to add parking space page	PASS
TS_PARKY_001	Verify Owner Login Functionality	TC_PARKY_OwnerLogin_004	Enter valid email, password	1. Enter valid email 2. Enter valid password 3. Click on login button	Valid URL test data	email: annaa21 password: Anna@3	Error Message: Invalid credentials	Pop up appears showing error message: Invalid Credentials	Login button disabled	PASS
TS_PARKY_001	Verify Admin Login Functionality	TC_PARKY_OwnerLogin_005	Enter valid email, password	1. Enter valid email 2. Enter valid password 3. Click on login button	Valid URL test data	email: pms.parking24by7@gmail.com password: Pass@1234	Admin redirected to Admin page. Successfull login popup appears	Successfull login Redirected to admin page	Successful login page Redirected to Admin page	PASS
TS_PARKY_001	Verify Admin Login Functionality	TC_PARKY_OwnerLogin_006	Enter valid email, password	1. Enter valid email 2. Enter valid password 3. Click on login button	Valid URL test data	email: pms.parking24by7@gmail.com password: Pass@12	Error Message: Invalid credentials	Pop up appears showing error message: Invalid Credentials	Login button disabled	PASS

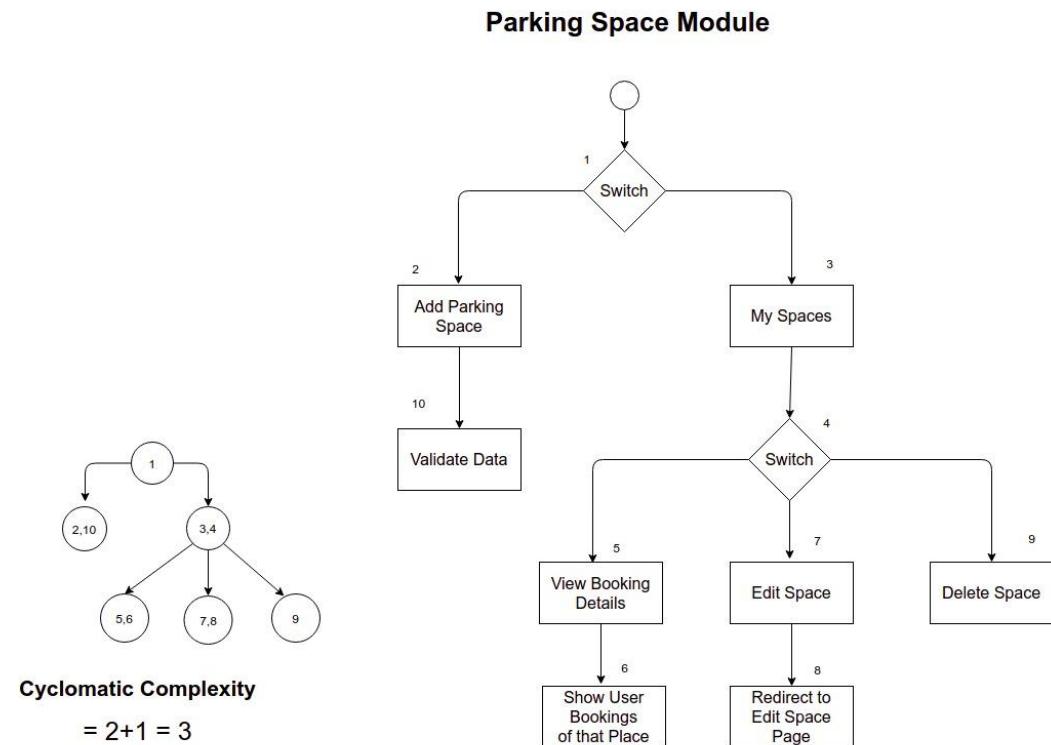
2] Admin Module:



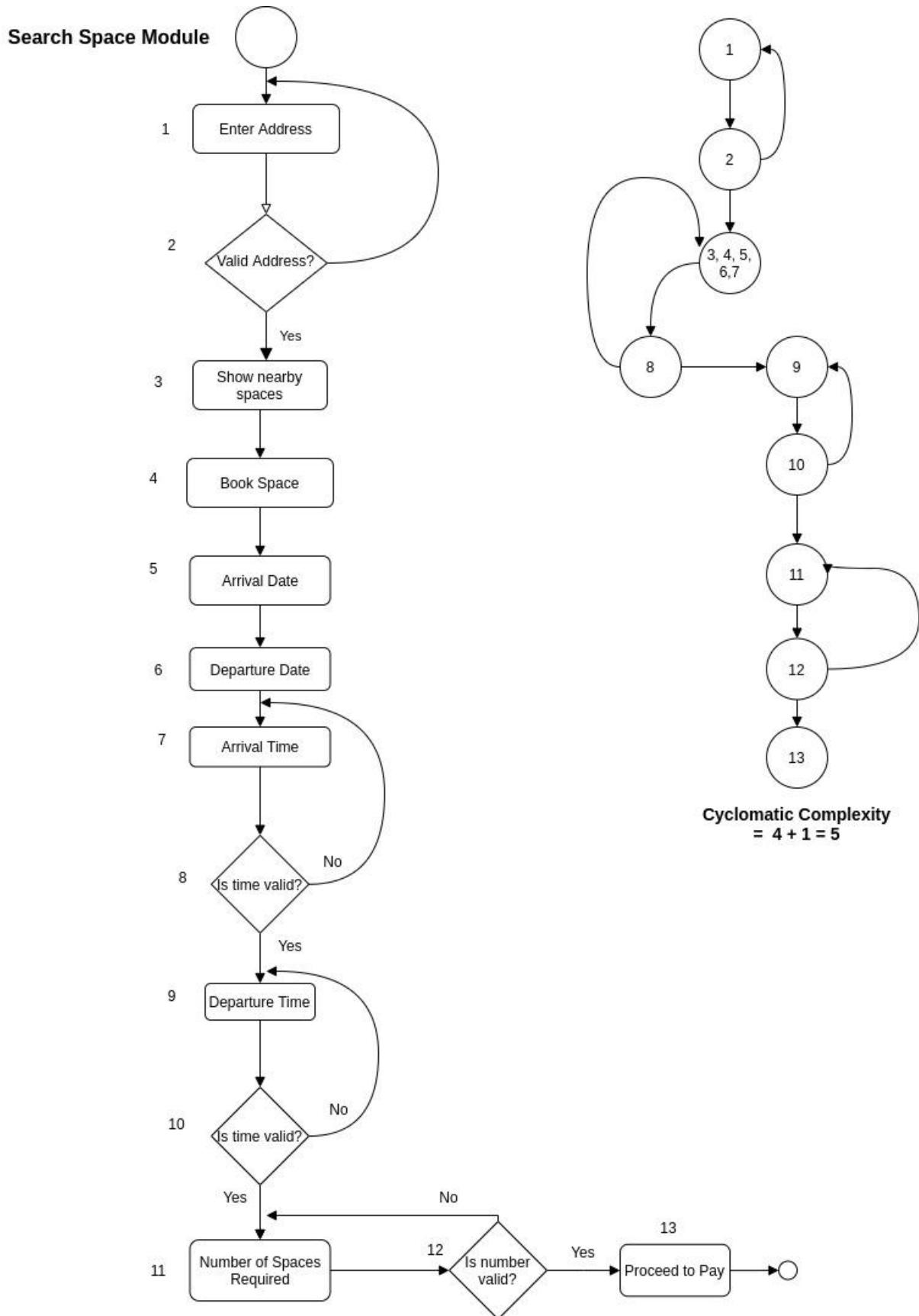
Parky

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status
TS_PARKY_002	Verify Admin Module Functionalities	TC_PARKY_AdminModule_001	Click on Button Verify Spaces. If space is not verified then click on verify space button	1. Click on Verify Spaces Button 2. If space is not verified then click on verify space button.	Valid URL test data		Verify space page is reloaded	Space status changes to verified and page is reloaded	Space status changes to verified and page is reloaded	PASS
TS_PARKY_002	Verify Admin Module Functionalities	TC_PARKY_AdminModule_002	Click on Manage User Button If admin wants delete user , clicks on Delete User Button	1. Click on Manage User Button 2. Click on Delete User Button 3. Confirm whether to delete or cancel	Valid URL test data		User is deleted from the database and page gets reloaded	User gets deleted from database and showuser page is reloaded	User gets deleted from database and showuser page is reloaded	PASS
TS_PARKY_002	Verify Admin Module Functionalities	TC_PARKY_AdminModule_003	Click on Manage User Button If admin wants to edit the user details clicks on the Edit User button. Enter the details to be updated Click on edit button	1. Click on Manage User Button 2. Click on Edit User button 3. Enter the details to be updated. 4. Click on edit button	Valid URL test data	Name : Andy Contact no: 9999999888	Updated values are stored in database.	Updated values are stored in database and Admin is redirected to Showuser page	Admin not redirected to showuser page	FAIL

3] Parking Space Module



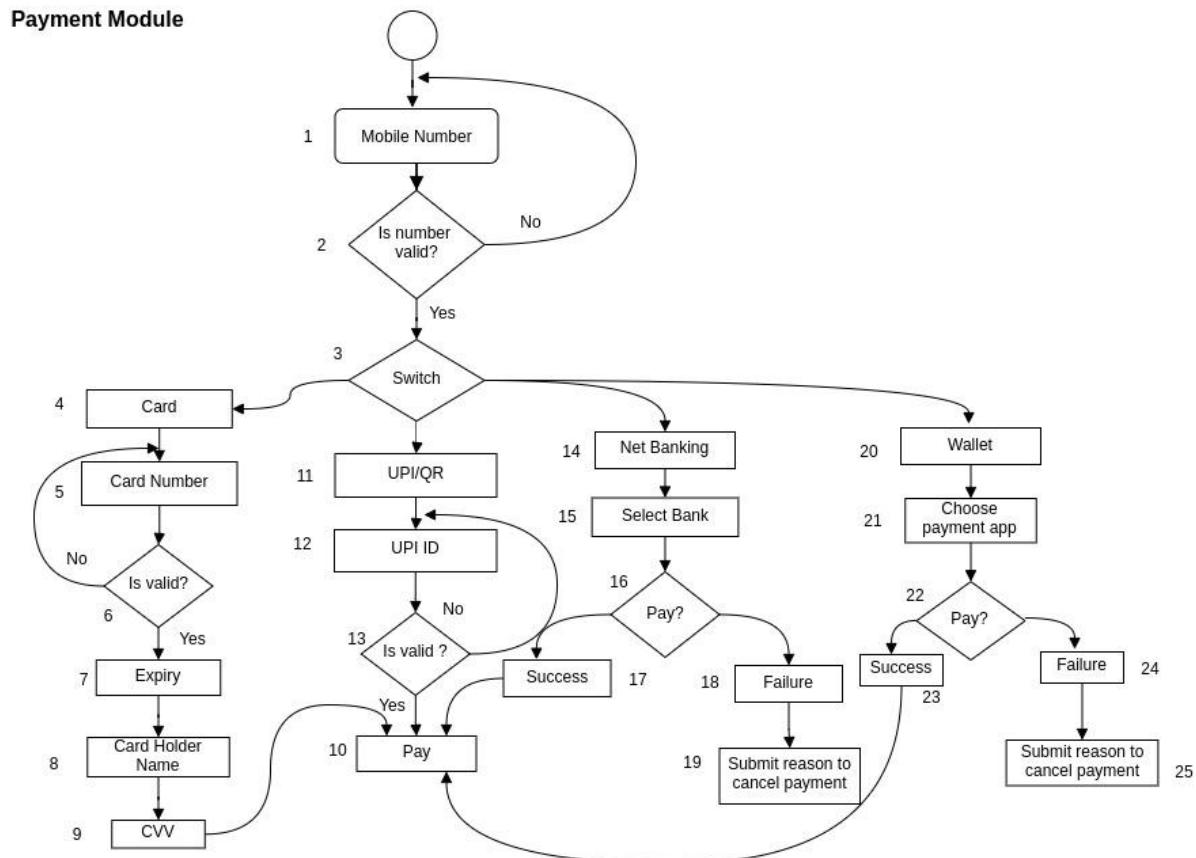
4] Search Space Module



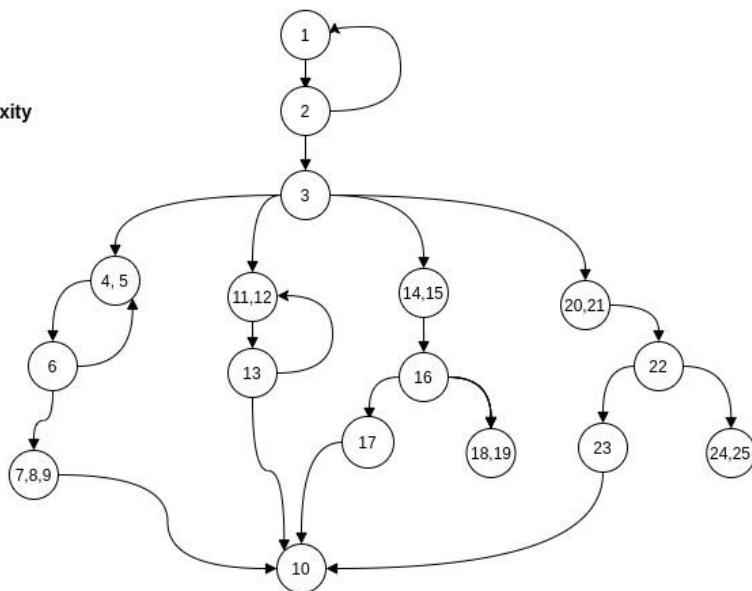
Parky

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status
TS_PARKY_010	Verify Search Space Functionality	TC_PARKY_SearchSpace_001	Enter valid address	1. Enter valid address 2. Click on Search Your Parking Space button 3. Select a place to Book. 4. Enter Valid Arrival Date 5. Enter Valid Departure Date 6. Enter Valid Arrival Time 7. Enter Valid Departure Time 8. Enter Valid Number of spaces 9. Click Proceed Button	Valid URL	Address: COEP Main Building, RB Motilal Kennedy Rd, Shivajinagar, Pune, Maharashtra 411005, India Book Space : Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Arrival date: 04/26/2021 Departure date: 04/26/2021 Arrival time: 11:00 Departure time: 13:00 No of Spaces: 2	Entered address will be cleared and user gets redirected to BookSlot Page	Address is searched by google maps and nearby spaces are displayed if available. After selecting the place and filling the checkout form, user redirected to booking history page	Address is searched by google maps and nearby spaces are displayed if available. After selecting the place and filling the checkout form, user redirected to booking history page	PASS
TS_PARKY_010	Verify Search Space Functionality	TC_PARKY_SearchSpace_002	Keep the address field blank	1. Keep the address field blank. 2. Click on Search Your Parking Space button	Valid URL	Address: Blank	User is redirected to the search space page	Invalid address pop up message is displayed to the user and the user is redirected to the search space page.	Invalid address pop up message is displayed to the user and the user is redirected to the search space page.	PASS
TS_PARKY_011	Verify Arrival Time Functionality	TC_PARKY_ArrivalTime_001	Enter invalid arrival time	1. Enter valid address. 2. Click on Search Your Parking Space button 3. Enter valid address 4. Select a place to Book. 5. Enter Arrival Date 6. Enter Departure Date 7. Enter Invalid Arrival Time	Valid URL	Address: COEP Main Building, RB Motilal Kennedy Rd, Shivajinagar, Pune, Maharashtra 411005, India Book Space : Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Arrival date: 04/26/2021 Departure date: 04/26/2021 Arrival Time: 11:00 (less than current time)	User sees the warning to enter valid arrival time	User sees the warning to enter valid arrival time and button to proceed is hidden.	User sees the warning to enter valid arrival time and button to proceed is hidden.	PASS

5] Payment Module



Cyclomatic Complexity
 $= 6 + 1 = 7$



Parky

TS_PARKY_011	Verify Valid Mobile Number Functionality	TC_PARKY_MobileNumber_001	Enter invalid mobile number	1.Enter invalid mobile number.	Valid URL	Mobile Number : 97756634568	User sees an alert message as warning	User sees a pop up message to enter a valid contact number and the proceed button is disabled	User sees a pop up message to enter a valid contact number and the proceed button is disabled	PASS
TS_PARKY_011	Verify Card Functionality	TC_PARKY_Card_001	Enter mobile number,card number, expiry date, Card Holder Name CVV	1.Enter valid mobile number 2.Choose Card Payment Method 3.Enter valid card number. 4.Enter valid expiry date. 5.Enter Card Holder Name 6.Enter valid CVV 7.Click pay	Valid URL	Mobile Number: 9604237506 Card Number: 1111 1111 1111 Expiry Date: 12/23 Card Hold Name : Riddhi Ghate CVV: 621	User sees a razorpay window to select success or failure	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	PASS
TS_PARKY_011	Verify Card Functionality	TC_PARKY_Card_002	Enter mobile number, choose card payment method,invalid card number	1.Enter valid mobile number 2.Choose Card Payment method. 3.Enter invalid card number.	Valid URL	Mobile Number: 9604237506 Card Number: 1111 1111 1111	User sees a pop up message	User sees a pop up message to enter a valid card number and the proceed button is disabled.	User sees a pop up message to enter a valid card number and the proceed button is disabled.	PASS
TS_PARKY_011	Verify UPI/QR Functionality	TC_PARKY_UPI_001	Enter mobile number, choose UPI/Qr payment method	1.Enter valid mobile number 2.Choose UPI/QR Method 3.Enter valid UPI id. 4.Click pay	Valid URL	Mobile Number: 9604237506 UPI ID: 9604237506@ybl	User sees the razorpay window to select success or failure	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	PASS
TS_PARKY_011	Verify UPI/QR Functionality	TC_PARKY_UPI_002	Enter mobile number, choose UPI/QR method , invalid UPI ID	1.Enter valid mobile number 2.Enter invalid card number.	Valid URL	Mobile Number: 9604237506 UPI ID: acbd	User sees a pop up message	User sees a pop up message to enter a valid VPA of the from username@bank and the proceed button is disabled	User sees a pop up message to enter a valid VPA of the from username@bank and the proceed button is disabled	PASS
TS_PARKY_011	Verify NetBanking Functionality	TC_PARKY_UPI_001	Enter mobile number, choose Netbanking payment method, select a bank	1.Enter valid mobile number 2.Choose NetBanking Method 3.Select Bank 4.Click pay 5.Choose Success	Valid URL	Mobile Number: 9604237506 Bank: SBI	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS
TS_PARKY_011	Verify NetBanking Functionality	TC_PARKY_UPI_002	Enter mobile number, choose Netbanking payment method, select a bank	1.Enter valid mobile number 2.Choose NetBanking Method 3.Select Bank 4.Click pay 5.Choose Failure	Valid URL	Mobile Number: 9604237506 Bank: SBI	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS
TS_PARKY_011	Verify Wallet Functionality	TC_PARKY_Wallet_001	Enter mobile number, choose Wallet payment method, select a bank	1.Enter valid mobile number 2.Choose Wallet Method 3.Select Bank 4.Click pay 5.Choose Success	Valid URL	Mobile Number: 9604237506 Wallet: MobiKwik	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS
TS_PARKY_011	Verify Wallet Functionality	TC_PARKY_Wallet_002	Enter mobile number, choose Wallet payment method, select a bank	1.Enter valid mobile number 2.Choose Wallet Method 3.Select Bank 4.Click pay 5.Choose Failure	Valid URL	Mobile Number: 9604237506 Wallet: Mobikwik	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS

2] Black Box Testing:**1] Register Module:****Equivalence Classes = 12**

1]Register Name	
Invalid Class	Less than 3 characters
Valid Class	3-20 Characters
Invalid Class	Greater than 20 characters
2]Register Email	
Invalid Class	Less than 3 characters
Valid Class	3-320 Characters
Invalid Class	Greater than 320 characters
3]Register Contact	
Invalid Class	Less than 10 characters
Valid Class	10 Characters
Invalid Class	Greater than 10 characters
4]Register Password	
Invalid Class	Less than 8 characters
Valid Class	8-15 Characters with at least one uppercase, one lowercase and one special character and onfirm password same as Password
Invalid Class	Greater than 15 characters

Parky

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data
1]Register Name				BVA Classes		
Invalid Class	Less than 3 characters			Invalid Class	2	
Valid Class	3-20 Characters			Valid Class	3-20	
Invalid Class	Greater than 20 characters			Invalid Class	21	
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John Andy John Andy email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21
BVA						
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John Andy John Andy email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21

Parky

TS_PARKY_011	Verify Valid Mobile Number Functionality	TC_PARKY_MobileNumber_001	Enter invalid mobile number	1.Enter invalid mobile number.	Valid URL	Mobile Number : 97756634568	User sees an alert message as warning	User sees a pop up message to enter a valid contact number and the proceed button is disabled	User sees a pop up message to enter a valid contact number and the proceed button is disabled	PASS
TS_PARKY_011	Verify Card Functionality	TC_PARKY_Card_001	Enter mobile number,card number, expiry date, Card Holder Name CVV	1.Enter valid mobile number 2.Choose Card Payment Method 3.Enter valid card number. 4.Enter valid expiry date. 5.Enter Card Holder Name 6.Enter valid CVV 7.Click pay	Valid URL	Mobile Number: 9604237506 Card Number: 1111 1111 1111 Expiry Date: 12/23 Card Hold Name : Riddhi Ghate CVV: 621	User sees a razorpay window to select success or failure	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	PASS
TS_PARKY_011	Verify Card Functionality	TC_PARKY_Card_002	Enter mobile number, choose card payment method,invalid card number	1.Enter valid mobile number 2.Choose Card Payment method. 3.Enter invalid card number.	Valid URL	Mobile Number: 9604237506 Card Number: 1111 1111 1111	User sees a pop up message	User sees a pop up message to enter a valid card number and the proceed button is disabled.	User sees a pop up message to enter a valid card number and the proceed button is disabled.	PASS
TS_PARKY_011	Verify UPI/QR Functionality	TC_PARKY_UPI_001	Enter mobile number, choose UPI/Qr payment method	1.Enter valid mobile number 2.Choose UPI/QR Method 3.Enter valid UPI id. 4.Click pay	Valid URL	Mobile Number: 9604237506 UPI ID: 9604237506@ybl	User sees the razorpay window to select success or failure	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	User sees a razorpay window to select success or failure and the payment will be successful on selecting success	PASS
TS_PARKY_011	Verify UPI/QR Functionality	TC_PARKY_UPI_002	Enter mobile number, choose UPI/QR method , invalid UPI ID	1.Enter valid mobile number 2.Enter invalid card number.	Valid URL	Mobile Number: 9604237506 UPI ID: acbd	User sees a pop up message	User sees a pop up message to enter a valid VPA of the from username@bank and the proceed button is disabled	User sees a pop up message to enter a valid VPA of the from username@bank and the proceed button is disabled	PASS
TS_PARKY_011	Verify NetBanking Functionality	TC_PARKY_UPI_001	Enter mobile number, choose Netbanking payment method, select a bank	1.Enter valid mobile number 2.Choose NetBanking Method 3.Select Bank 4.Click pay 5.Choose Success	Valid URL	Mobile Number: 9604237506 Bank: SBI	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS
TS_PARKY_011	Verify NetBanking Functionality	TC_PARKY_UPI_002	Enter mobile number, choose Netbanking payment method, select a bank	1.Enter valid mobile number 2.Choose NetBanking Method 3.Select Bank 4.Click pay 5.Choose Failure	Valid URL	Mobile Number: 9604237506 Bank: SBI	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS
TS_PARKY_011	Verify Wallet Functionality	TC_PARKY_Wallet_001	Enter mobile number, choose Wallet payment method, select a bank	1.Enter valid mobile number 2.Choose Wallet Method 3.Select Bank 4.Click pay 5.Choose Success	Valid URL	Mobile Number: 9604237506 Wallet: MobiKwik	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS
TS_PARKY_011	Verify Wallet Functionality	TC_PARKY_Wallet_002	Enter mobile number, choose Wallet payment method, select a bank	1.Enter valid mobile number 2.Choose Wallet Method 3.Select Bank 4.Click pay 5.Choose Failure	Valid URL	Mobile Number: 9604237506 Wallet: Mobikwik	User sees the razorpay window to select success or failure	User is redirected to the booking history page.	User is redirected to the booking history page.	PASS

Parky

TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John Andy John AndyJ email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21	Error Message: Register Button gets c Name should contain	Register Button gets PASS
2) Register Email								
Invalid Class	Less than 3 characters			Invalid Class	2			
Valid Class	3-320 Characters			Valid Class	3-320			
Invalid Class	Greater than 320 characters			Invalid Class	321			
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy email : an contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21	Error Message: Ema	Register Button gets c Register Button gets PASS
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John email : [greater than 320 char] contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21	Go to login Page	Successful Register
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John Andy John AndyJ email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21	Error Message: Ema	Register Button gets c Register Button gets PASS
BVA								
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy email : an contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21	Error Message: Ema	Register Button gets c Register Button gets PASS
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John email : (= 320 char) contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21	Go to login Page	Successful Register
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John email : e@g contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21	Go to login Page	Successful Register

Parky

4) Register Password					
Invalid Class	Less than 8 characters				Invalid Class
Valid Class	8-15 Characters with at least one uppercase, one lowercase and one special character and confirm password same as Password				Valid Class
Invalid Class	Greater than 15 characters				Invalid Class
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data Name: Andy email : an contact no: 9999999999 password: Andy confirm password: Andy
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data Name: Andy John email : {greater than 320 char} contact no: 9999999999 password: AndyJonas@21 confirm password: AndyJonas@21
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data Name: Andy John Andy email : andyjona@gmail.com contact no: 9999999999 password: AndyJonas@21@Andy confirm password: Andy.Jonas@21@Andy@Johns
BVA					
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data Name: Andy email : an contact no: 9999999999 password: Andyabc confirm password: Andy
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data Name: Andy John email : {greater than 320 char} contact no: 9999999999 password: AndyJ@21 confirm password: AndyJonas@21
TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data Name: Andy John email : {greater than 320 char} contact no: 9999999999 password: AndyJ@21 confirm password: AndyJonas@21

Parky

TS_PARKY_001	Verify Register functionality	TC_PARKY_Reg	Enter valid name, email, contact number, password, confirm password	1. Enter name 2. Enter email 3. Enter contact number 4. Enter password 5. Confirm password entered 6. Click on register button	Valid URL test data	Name: Andy John Andy email : andyjona@gmail.com contact no: 99999999 password: Andy.Jonas@21@And confirm password: Andy.Jonas@21@Andy@Johns	Error Password shoi	Register Button gets <	Register Button gets PASS
--------------	-------------------------------	--------------	---	---	---------------------	---	---------------------	------------------------	---------------------------

2] Book Space Module

Equivalence Classes = 10

1] Arrival Date:

Valid class: greater than current date

Invalid class : less than current date

2] Departure Date:

Valid class: greater than equal to arrival date

Invalid class: less than arrival date

3] Arrival Time :

Valid class: greater than equal to current time

Invalid class : less than current time

4] Departure Time:

Valid class: greater than arrival time

Invalid class : less than or equal to arrival time

5] No of spaces required:

Valid class: less than or equal to total no of spaces

Invalid class : gretaer than total no of spaces

Parky

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data
TS_PARKY_012	Verify Book Space Functionality	TC_PARKY_BookSpace_001	Enter valid arrival date, departure date, arrival time, departure time, no of spaces	1. Enter valid arrival date 2. Enter valid departure date 3. Enter valid arrival time 4. Enter valid departure time 5. Enter no of spaces		Arrival date: 04/26/2021 Departure date: 04/26/2021 Arrival time: 11:00 Departure time: 13:00 No of Spaces: 2
TS_PARKY_012	Verify Book Space Functionality	TC_PARKY_BookSpace_002	Enter valid arrival date, departure date, arrival time, departure time, no of spaces	1. Enter valid arrival date 2. Enter valid departure date 3. Enter valid arrival time 4. Enter valid departure time 5. Enter no of spaces	Valid URL	Arrival date: 04/26/2021 Departure date: 04/26/2021 Arrival time: (less than current time) Departure time: 13:00 No of Spaces: 2
TS_PARKY_012	Verify Book Space Functionality	TC_PARKY_BookSpace_003	Enter valid arrival date, departure date, arrival time, departure time, no of spaces	1. Enter valid arrival date 2. Enter valid departure date 3. Enter valid arrival time 4. Enter valid departure time 5. Enter no of spaces	Valid URL	Arrival date: 04/26/2021 Departure date: 04/26/2021 Arrival time: 11:00 Departure time: 10:00 (less than arrival time) No of Spaces: 2
TS_PARKY_012	Verify Book Space Functionality	TC_PARKY_BookSpace_004	Enter valid arrival date, departure date, arrival time, departure time, no of spaces	1. Enter valid arrival date 2. Enter valid departure date 3. Enter valid arrival time 4. Enter valid departure time 5. Enter no of spaces	Valid URL	Arrival date: 04/26/2021 Departure date: 04/26/2021 Arrival time: 11:00 Departure time: 14:00 No of Spaces: (greater than total no of spaces)

3] Payment Module:

Equivalence Classes: 3

Equivalence Classes: 3	BVA Classes:
Contact No: Valid class: no of digits = 10 Invalid class: no of digits < 10 Invalid class: no of digits > 10	Contact No: Valid class: no of digits = 10 Invalid class : no of digits = 9 Invalid class: no of digits = 11

Parky

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expected Result	Actual Result	Status
TS_PARKY_013	Verify Payment Functionality	TC_PARKY_Payment_001	Enter valid mobile number,valid email	1. Enter valid mobile number 2. Enter valid email 3.Click Proceed	Valid URL	Mobile Number: 9604237506 Email: riddhiguate.07@gmail.com	The user is redirected to choose a payment method	The user is redirected to choose a payment method	The user is redirected to choose a payment method	PASS
TS_PARKY_013	Verify Payment Functionality	TC_PARKY_Payment_002	Enter invalid mobile number,valid email	1. Enter invalid mobile number 2. Enter valid email 3.Click Proceed	Valid URL	Mobile Number: 960423750656 Email: riddhiguate.07@gmail.com	The user stays on the same page	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	PASS
TS_PARKY_013	Verify Payment Functionality	TC_PARKY_Payment_002	Enter invalid mobile number,valid email	1. Enter invalid mobile number 2. Enter valid email 3.Click Proceed	Valid URL	Mobile Number: 960423750656 Email: riddhiguate.07@gmail.com	The user stays on the same page	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	PASS
BVA Classes										
Contact No: Valid class: no of digits = 10 Invalid class : no of digits = 9 Invalid class: no of digits = 11										
TS_PARKY_013	Verify Payment Functionality	TC_PARKY_Payment_001	Enter valid mobile number,valid email	1. Enter valid mobile number 2. Enter valid email 3.Click Proceed	Valid URL	Mobile Number: 9604237506 Email: riddhiguate.07@gmail.com	The user is redirected to choose a payment method	The user is redirected to choose a payment method	The user is redirected to choose a payment method	PASS
TS_PARKY_013	Verify Payment Functionality	TC_PARKY_Payment_002	Enter invalid mobile number,valid email	1. Enter invalid mobile number 2. Enter valid email 3.Click Proceed	Valid URL	Mobile Number: 960423750 Email: riddhiguate.07@gmail.com	The user stays on the same page	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	PASS
TS_PARKY_013	Verify Payment Functionality	TC_PARKY_Payment_002	Enter invalid mobile number,valid email	1. Enter invalid mobile number 2. Enter valid email 3.Click Proceed	Valid URL	Mobile Number: 960423750655 Email: riddhiguate.07@gmail.com	The user stays on the same page	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	The user is given a warning to enter valid 10 digit numbe and cannot proceed further	PASS

4] Edit Profile:

<p>Equivalence Classes: 6</p> <p>1]Number of digits in contact number: Invalid Class: Less than 10 digits Valid Class: 10 digits Invalid Class: Greater than 10 digits</p> <p>2]Name: Invalid Class: Less than 3 characters Valid Class: 3 - 20 characters Invalid Class: Greater than 20 characters</p>	<p>BVA Classes: 6</p> <p>1]Number of digits in contact number: Invalid Class: 9 digits Valid Class: 10 digits Invalid Class: 11 digits</p> <p>2]Name: Invalid Class: 2 characters Valid Class: 3 -, 20 characters Invalid Class: 21 characters</p>
---	---

Parky

Equivalence Class		Number of Digits in Contact Number	Invalid	Valid	Invalid			
			< 10	10	> 10			
Test Scenario ID		Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Priority
TS_PARKY_014	Verify Edit Contact Number Functionality	TC_PARKY_EditContactNumber_002	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: Andy Contact no: 999999998	Error Contact number should be	Medium
TS_PARKY_014	Verify Edit Contact Number Functionality	TC_PARKY_EditContactNumber_001	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: Andy Contact no: 999999998	User red pri	Medium
TS_PARKY_014	Verify Edit Contact Number Functionality	TC_PARKY_EditContactNumber_002	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: Andy Contact no: 99999999866	Error Contact number should be	Medium
Equivalence Class		Number of Characters in Name	Invalid	Valid	Invalid			
			< 3	3 - 20	> 20			
TS_PARKY_014	Verify Edit Name Functionality	TC_PARKY_EditName_001	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: An Contact no: 9999999989	Error Name length -20	Medium
TS_PARKY_014	Verify Edit Name Functionality	TC_PARKY_EditName_002	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: Andy Contact no: 999999998	User red pri	Medium
TS_PARKY_014	Verify Edit Name Functionality	TC_PARKY_EditName_003	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: DrRajeshMKoothrapali Contact no: 999999998	Error Name length -20	Medium
Equivalence Class		Number of Digits in Contact Number	Invalid	Valid	Invalid			
			9 - 10	11				
TS_PARKY_014	Verify Edit Contact Number Functionality	TC_PARKY_EditContactNumber_002	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: Andy Contact no: 999999998	Error Contact number should be	Medium
TS_PARKY_014	Verify Edit Contact Number Functionality	TC_PARKY_EditContactNumber_001	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: Andy Contact no: 999999998	User red pri	Medium
TS_PARKY_014	Verify Edit Contact Number Functionality	TC_PARKY_EditContactNumber_002	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: Andy Contact no: 99999999866	Error Contact number should be	Medium

Equivalence Class		Number of Characters in Name	Invalid	Valid	Invalid			
			2 - 3 - 20		21			
Test Scenario ID		Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Priority
TS_PARKY_014	Verify Edit Name Functionality	TC_PARKY_EditName_001	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: An Contact no: 9999999989	Error Name length -20	Medium
TS_PARKY_014	Verify Edit Name Functionality	TC_PARKY_EditName_002	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: And Contact no: 999999998	User red pri	Medium
TS_PARKY_014	Verify Edit Name Functionality	TC_PARKY_EditName_003	Edit name and contact no	1.Edit username 2. Edit contact no 3. Click Update button	Valid URL	Name: DrRajeshMKoothrapali Contact no: 999999998	Error Name length -20	Medium

5] Add Parking Space Module:

Equivalence Classes : 8	BVA Classes : 8
1]Cost of Spaces: Valid Class: Cost ≥ 30 Invalid Class: Cost < 30	1]Cost of Spaces: Valid Class: Cost = 29.99 Invalid Class: Cost = 30.01
2]Address Field Length: Valid Class: Length > 0 Invalid Class: Length ≤ 0	2]Address Field Length: Valid Class: Length = 1 Invalid Class: Length = 0
3]Number of Spaces: Valid Class: Number of spaces > 0 Invalid Class: Number of spaces ≤ 0	3]Number of Spaces: Valid Class: Number of spaces = 1 Invalid Class: Number of spaces = 0
4]File Size: Valid Class: File Size $\leq 5\text{MB}$ Invalid Class: File Size $> 5\text{ MB}$	4]File Size: Valid Class: File Size = 4.99 MB Invalid Class: File Size = 5.01 MB

Parky

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data
Equivalence Class	Cost of Space	Invalid < 30	Valid > 30			
TS_PARKY_005	Verify Valid Cost Functionality	TC_PARKY_ValidCost_001	Enter valid cost of space, address, info, surface type number of spaces, accepted vehicles, document	1.Enter valid cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL and test data	Cost : 40 Address: Pratap Nagar, Nagpur, Maharashtra, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf
TS_PARKY_005	Verify Valid Cost Functionality	TC_PARKY_ValidCost_002	Enter cost of space < 30, address, info, surface type number of spaces, accepted vehicles, document	1.Enter cost < 30 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 20 Address: Pratap Nagar, Nagpur, Maharashtra, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf
Equivalence Class	Addresss field length	Invalid <= 0	Valid > 0			
TS_PARKY_006	Verify Address Field LengthFunctionality	TC_PARKY_Address_001	Enter valid cost of space , address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf

Parky

TS_PARKY_006	Verify Address Field Length Functionality	TC_PARKY_Address_002	Enter valid cost of space , blank address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf	The address field will show the warning to enter the address	The address field gives a warning and the form is not submitted	The address field gives a warning and the form is not submitted	PASS
Equivalence Class										
	Number of Spaces		Invalid	Valid						
			< 1	>= 1						
TS_PARKY_007	Verify Number of Space Functionality	TC_PARKY_Number_001	Enter valid cost of space , address, info, surface type, valid number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf	The entered form details will get cleared and the owner can add another space	Space is added successfully	Space is added successfully	PASS
TS_PARKY_007	Verify Number of Space Functionality	TC_PARKY_Number_002	Enter valid cost of space , address, info, surface type, 0 number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter 0 number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 0 Accepted Vehicles: Car Document: pdf	The number of spaces field will show the warning to enter spaces greater than 0	The number of spaces field gives a warning and the form is not submitted.	The number of spaces field gives a warning and the form is not submitted.	PASS
Equivalence Class										
	File Size		Invalid	Valid						
			> 5 MB	<= 5 MB						

Parky

TS_PARKY_006	Verify File Size Functionality	TC_PARKY_FileSize_001	Enter valid cost of space , address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: 1 MB pdf	The entered form details will get cleared and the owner can add another space	Space is added successfully	Space is added successfully	PASS
TS_PARKY_006	Verify File Size Functionality	TC_PARKY_FileSize_002	Enter valid cost of space , address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: 6 MB pdf	The entered form details will get cleared and the owner can add another space	The document upload field gives a error message and submit button is disabled	The form is submitted but the space is not saved	FAIL
BVA Class										
	Cost of Space	Invalid	Valid							
				29.99	30.01					
TS_PARKY_005	Verify Cost of Space Functionality	TC_PARKY_Cost_001	Enter valid cost of space, address, info, surface type number of spaces, accepted vehicles, document	1.Enter valid cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL and test data	Cost : 30.01 Address: Pratap Nagar, Nagpur, Maharashtra, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf	The entered form details will get cleared and the owner can add another space	Space is added successfully	Space is added successfully	PASS

Parky

TS_PARKY_005	Verify Cost of Space Functionality	TC_PARKY_Cost_002	Enter cost of space < 30, address, info, surface type number of spaces, accepted vehicles, document	1.Enter cost < 30 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 29.99 Address: Pratap Nagar, Nagpur, Maharashtra, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf	The cost field will show a warning that price cannot be less than 30	The cost field shows a warning and the submit button is disabled.	The cost field shows a warning and the submit button is disabled.	PASS
BVA Class	Address field length	Invalid	Valid	0	1					
TS_PARKY_006	Verify Address Field Length Functionality	TC_PARKY_Address_001	Enter valid cost of space , address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: A Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf	The entered form details will get cleared and the owner can add another space	Space is added successfully	Space is added successfully	PASS
TS_PARKY_006	Verify Address Field Length Functionality	TC_PARKY_Address_002	Enter valid cost of space , blank address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: pdf	The address field will show the warning to enter the address	The address field gives a warning and the form is not submitted	The address field gives a warning and the form is not submitted	PASS
BVA Class	Number of Spaces	Invalid	Valid	0	1					

Parky

TS_PARKY_007	Verify Number of Space Functionality	TC_PARKY_Number_001	Enter valid cost of space , address, info, surface type, valid number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 1 Accepted Vehicles: Car Document: pdf	The entered form details will get cleared and the owner can add another space	Space is added successfully	Space is added successfully	PASS
TS_PARKY_007	Verify Number of Space Functionality	TC_PARKY_Number_002	Enter valid cost of space , address, info, surface type, 0 number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter 0 number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 0 Accepted Vehicles: Car Document: pdf	The number of spaces field will show the warning to enter spaces greater than 0	The number of spaces field gives a warning and the form is not submitted.	The number of spaces field gives a warning and the form is not submitted.	PASS
BVA Class										
TS_PARKY_006	Verify File Size Functionality	TC_PARKY_FileSize_001	Enter valid cost of space , address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: 4.99 MB pdf	The entered form details will get cleared and the owner can add another space	Space is added successfully	Space is added successfully	PASS

Parky

TS_PARKY_006	Verify File Size Functionality	TC_PARKY_FileSize_002	Enter valid cost of space , address, info, surface type number of spaces, accepted vehicles,document	1.Enter cost 2.Enter address 3.Enter information 4.Choose surface type 5.Enter number of spaces 6.Select accepted vehicles 7.Upload document for verification 8.Click on Submit button	Valid URL	Cost : 40 Address: Fergusson College Rd, Shivajinagar Pune, Maharashtra 411004, India Info : Space in front of office Surface Type: Covered Number of spaces: 2 Accepted Vehicles: Car Document: 5.01 MB pdf	The entered form details will get cleared and the owner can add another space	The document upload field gives a error message and submit button is disabled	The form is submitted but the space is not saved	FAIL
--------------	--------------------------------	-----------------------	--	---	-----------	--	---	---	--	------

10. Github Link

<https://github.com/mnarora/Parky>

11. References

http://ijarcsse.com/docs/papers/Volume_7/5_May2017/SV7I5-0254.pdf

<https://www.semanticscholar.org/paper/Research-of-Parking-Management-System-Based-on-Chen-Su/a451105844d321a18d10b4d6ae4ac8cd25500ca4>

<https://scialert.net/fulltext/?doi=itj.2009.101.113>

https://www.researchgate.net/publication/330242183_A_Novel_Parking_Management_System_for_Smart_Cities_to_save_Fuel_Time_and_Money

https://www.researchgate.net/publication/275272649_Parking_Management_System_Using_Mobile_Application

<https://digitalcommons.unl.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1045&context=computerscidiss>

https://www.researchgate.net/publication/344411337_The_Smart_Parking_Management_System

http://ijarcsse.com/docs/papers/Volume_7/5_May2017/SV7I5-0254.pdf

<https://ijarcce.com/upload/2017/march-17/IJARCCE%20136.pdf>