

# ONLINE PARKING SYSTEM

## CH-1 INTRODUCTION

### 1.1. Problem Statement

**To automate the existing private parking system**

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

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

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

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

residents.



## 1.2. Limitations of Current System

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

### Existing solutions:

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

## **Díawbacks**

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

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

### 1.3. Proposed System Solution

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

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

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

## 1.4. Advantages of Proposed Solution

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

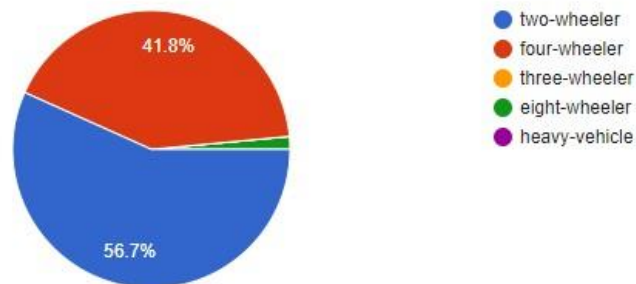
## 1.5 Google form link

<https://forms.gle/uSKfLZy5XoXdfixg9>

## Google form responses

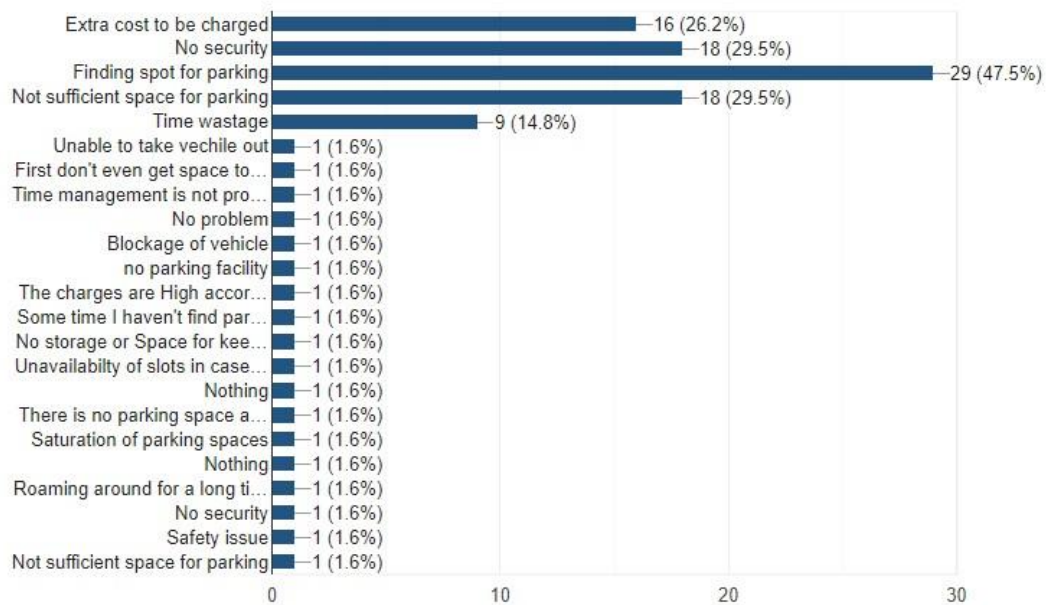
### 1. What type of vehicle you have?

67 responses



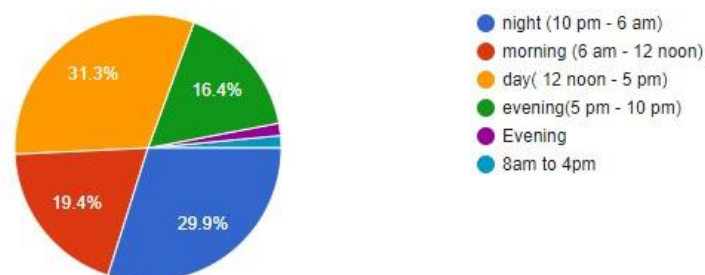
### 2. What type of problems you face at normal parking system

61 responses



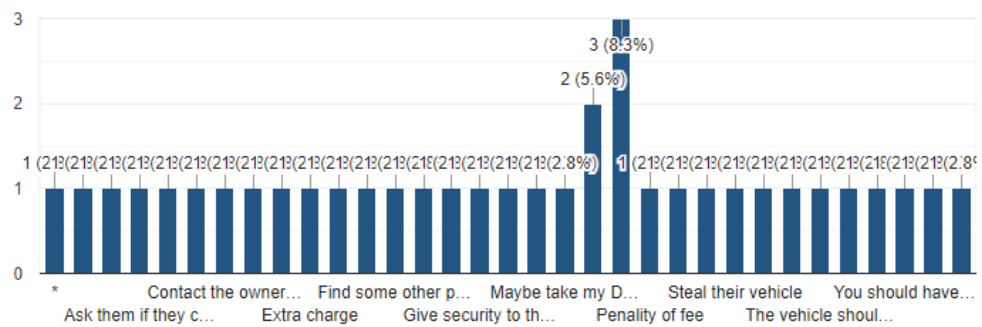
### 3. Your preferable time for parking

67 responses



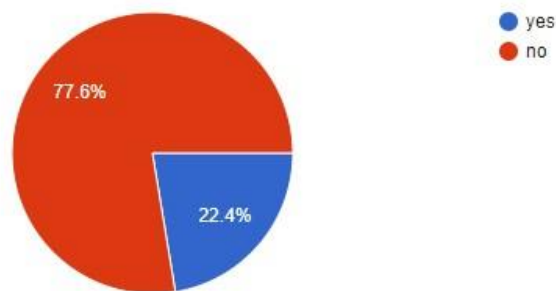
4. Suggestion if someone is unable to take the vehicle after the given time slot (what action should we take with your vehicle?)

36 responses



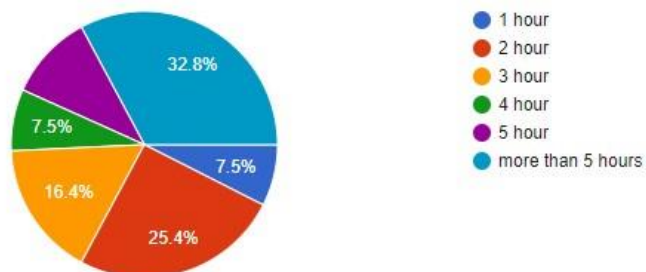
5. Are you willing to give your vehicle details and your contact number?

67 responses



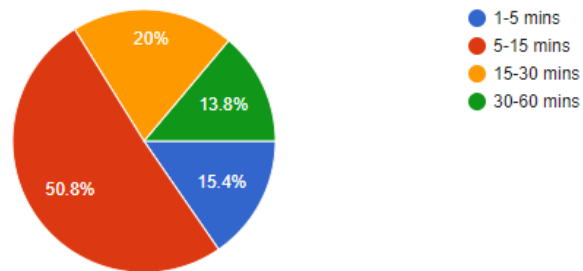
6. Duration for which you want to hire the parking slot

67 responses



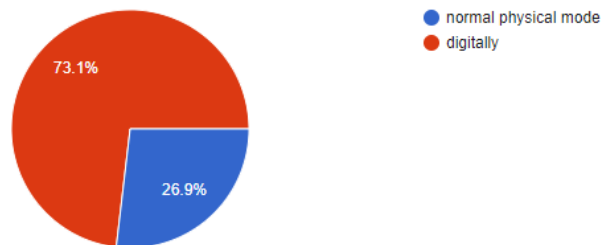
7. How much time you spend looking for a parking spot in normal parking system

65 responses



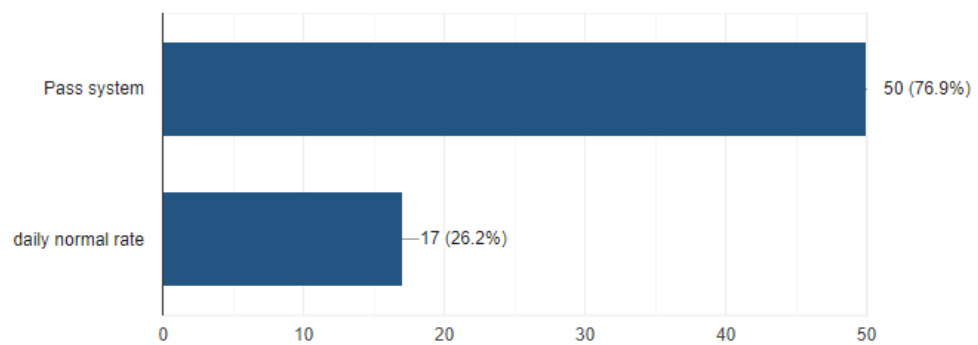
8. What type of token system you prefer.

67 responses



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

65 responses



# CH-2

## **Introduction:**

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

### 2.1.1 Purpose:

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

### 2.1.2 Document Conventions

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

### 2.1.3 Intended Audience and Reading suggestion:

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



## 2.1.4 Product scope

### In scope:

Users can book slots for parking.

Users can choose the time duration of parking.

Users can see the way to the parking slot.

Users can unbook the slot.

Users can see their parking History.

### Out of Scope:

Management of more vehicles than the limited space.

## 2.1.5 References:

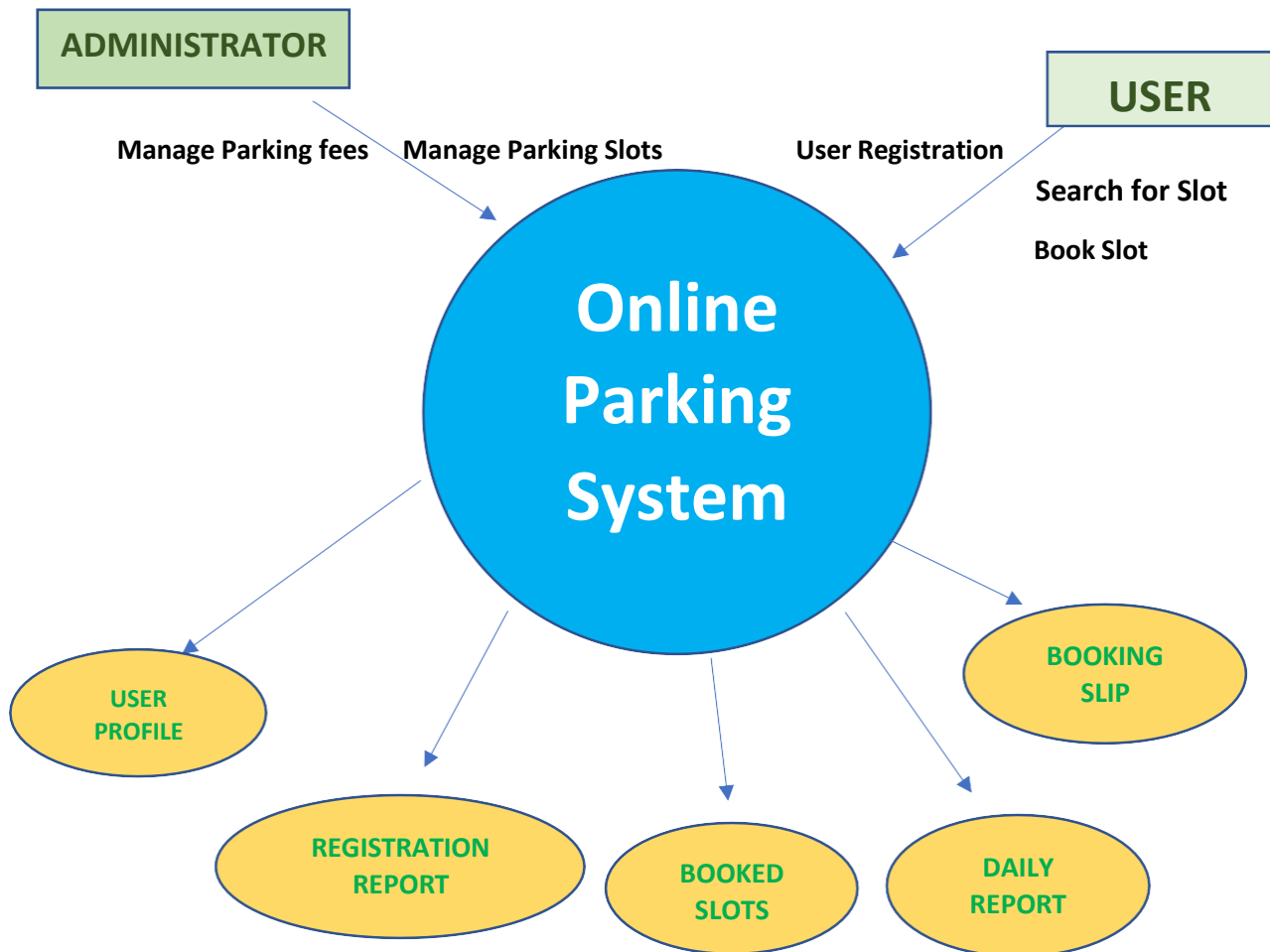
IEEE 830-1998 standard SRS template [IEEE SRS Template](#).

## 2.2 Overall Description

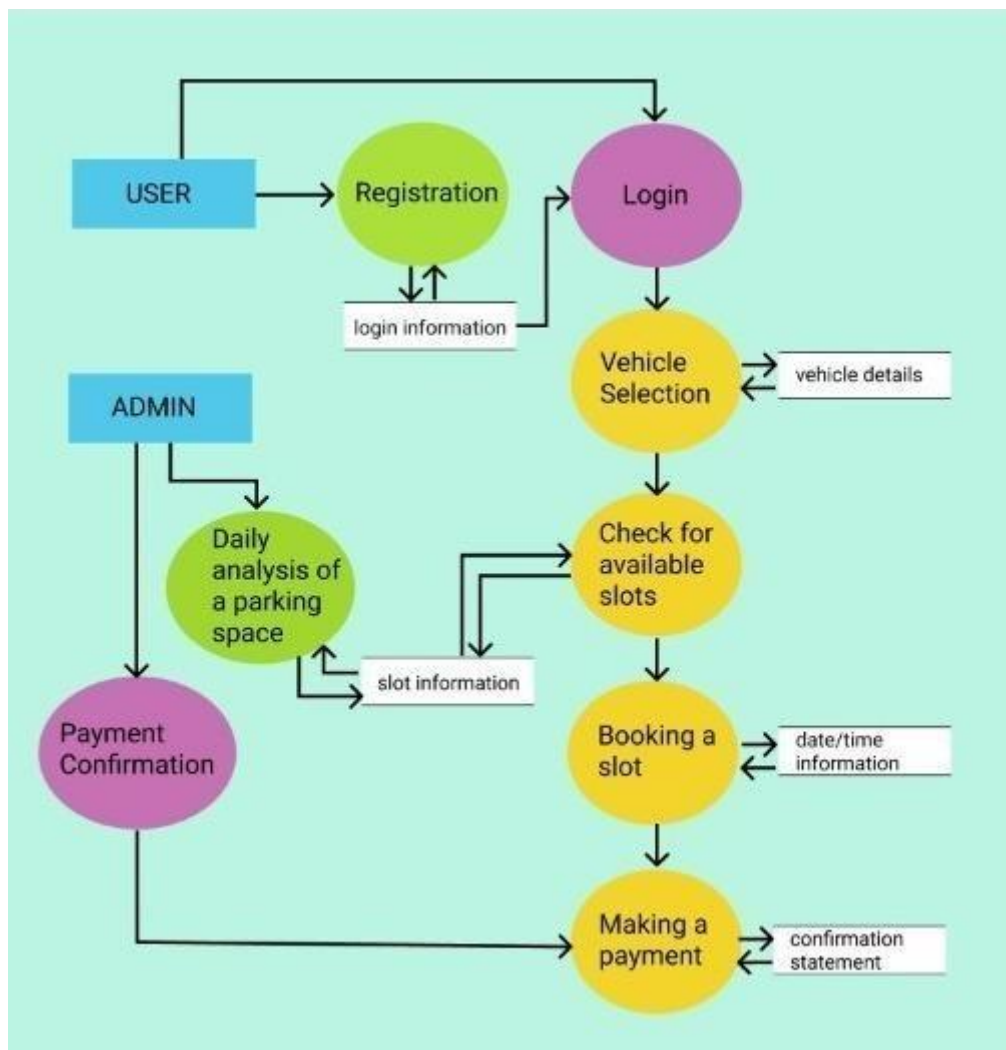
### 2.2.1 Product Prospective:

#### Data Flow Diagrams:

##### •0-Level DFD

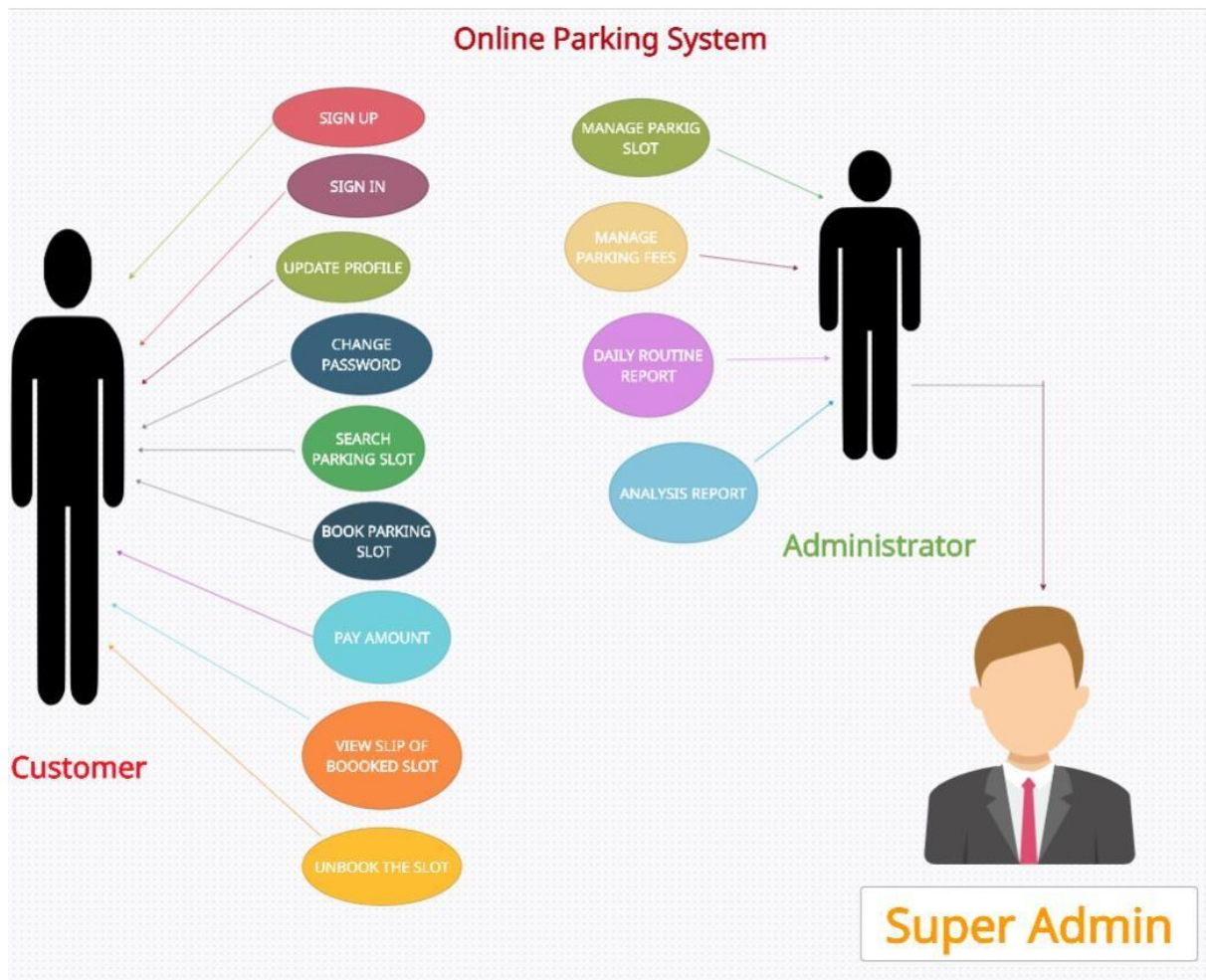


Dfd1



## 2.2.2 Product Functions

ParkVic APP



## Use Case Diagram

**User case 1:** sign up

**Primary actor:** app user

**Pre-condition:** log in page is available

### Success scenario:

System shows sign-up page

User fill the details like name email password etc.

User submit the details

System stores the details in database.

System shows successfully registered notification.

System provides user id

### Exception scenario:

Invalid email id

User already exists.

All information not filled.

### Basic flow:

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

### Alternate flow:

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

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

### User case 2: log in

**Primary actor:** Registered user

**Pre-condition:** Log in page is available

**Success scenario:**

System shows login page  
User enter his user id and password  
User submit the details  
System checks the details in database.  
Log in successful.

**Exception scenario:**

Invalid email id  
Invalid password  
User not registered.

**Basic flow:**

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

**Alternate flow:**

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

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

**User case 3:** Forgot password page

**Primary actor:** Registered user

**Pre-condition:** Should be registered before

Have a valid and registered email id

**Success scenario:**

User clicks on forgot password button

System show forgot password page

User enter the email

System checks the details in database.

Provide create new password option

User creates new password

System displays Log in

**Exception scenario:**

Attempts to change password exceeds

**Basic flow:**

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

**Alternate flow:**

Forgot password-> enter email id->attempts exceeds

**Use case 4:** Booking of parking slots

**Primary actor:** Registered user

**Pre-condition:** Login page is available

**success scenario:** System show parking slots

User choose slot information (slot time and date)

System check for chosen parking slot

System confirms the parking slot

System directed to payment page

Slot booked confirmation message

**Exception scenario:** Slots not available

All information not filled-> show error

**Basic flow:**

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

**Alternate flow:**

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

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

**Use case 5:** Payment page

**Primary actor:** Registered user



**Pre-condition:** Slot should be booked

**Main success scenario:** System show payment page

User choose the payment option

User pay the amount

Slot booked confirmation message

**Exception scenario:** Payment not done

Server error

**Basic flow:**

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

**Alternate flow:**

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

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

**Use case 6:** Unbook the slot

**Primary actor:** Registered user

**Pre-condition:** Should booked a slot before

**Main success scenario:** System show the unbook page

User unbook the slot

System returns the payment after  
deducting fine

Redirected to booking page

**Exception scenario:** unable to unbook

Payment not returned

**Basic flow:**

unbook page->unbook the slot->returns payment

**Alternate flow:**

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

**Use case 7:** Parking history

**Primary actor:** Registered user

**Pre-condition:** Vehicle should be parked before

**Main success scenario:** System show history page

User see his history

**Exception scenario:** no history available

**Basic flow:**

History page-> show history

**Alternate flow:**

History page-> no history

**Use case 8:** Daily Analysis

**Primary actor:** Admin

**Pre-condition:** Should be logged in

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

**Exception scenario:** No report generated

**Basic flow:**

**Analysis page-> show report**

**Alternate flow:**

**analysis page-> no report generation**

## 2.2.3 User Classes and Characteristics:

### 1. General User:

- a. Users will be able to use the app to book Parking slots.

- b. Users can manage their timing and can view their parked History.

## 2. Admin

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

## 3. Super Admin

- a. Manage all important information and view records

### 2.2.4 Operating Environment

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

### 2.2.5 Design and Implementation Constraints:

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

### 2.2.6 Assumptions and Dependencies:

- 1. All the data entered will be correct and up to date.

2. The user is connected to the internet and is willing to share personal Information online.

## 2.3 External Interface Requirements.

### 2.3.1 User Interfaces:

GUI (Graphical User Interface)

### 2.3.2 Hardware Interfaces:

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

### 2.3.3 Software Interfaces:

HTML, CSS, JAVASCRIPT, PHP

OS - Android Studio

Database- SQL

### 2.3.4 Communications Interfaces:

- 1 . Android
- 2 . SMS Service protocol

3 . POP protocol

4 . TCP/IP

## 2.4 System Features

### 2.4.1 Sign up

#### 2.4.1.1 Description

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

#### 2.4.1.2 Stimulus/Response Sequence

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

#### 2.4.1.3 Functional Requirements

Internet Connectivity and Data usage Permission is Necessary

### 2.4.2 Login

#### 2.4.2.1 Description

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

#### 2.4.2.2 Stimulus/Response Sequence

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

#### 2.4.2.3 Functional Requirements

Internet Connectivity and Data usage Permission is Necessary

Users should be signed Up for the App.

### 2.4.3 Booking of Parking Slot

#### 2.4.3.1 Description



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

#### 2.4.3.2 Sequence

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

#### 2.4.3.3 Functional Requirements

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

### 2.4.4 Payment of Booked Parking Slot

#### 2.4.4.1 Description

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

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

#### 2.4.4.2 Sequence

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

#### 2.4.4.3 Functional Requirements

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

### 2.4.5 Unbooking of Parking Slot

#### 2.4.5.1 Description

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

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

#### 2.4.5.2 Sequence

1. User has to select the booked slot.
2. Choose the unbook option

3. pay the unbooking charge.

4. Slot is unbooked.

#### 2.4.5.3 Functional Requirements

1 . Internet connectivity is must.

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

## 2.5 Other Non-functional Requirements

### 2.5.1 Performance Requirements

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

Capacity-The system must support 1000 people at a time

User interface- User interface screen will be very responsive.

### 2.5.2 Safety Requirements

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

### 2.5.3 Security Requirements

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

### 2.5.4 Availability:

App can be downloaded from Google play store or Apple store

### 2.5.5 Software Quality Attributes

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

Modularity: App is modular.

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

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

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

Accuracy: The reliability of the information and output. Can

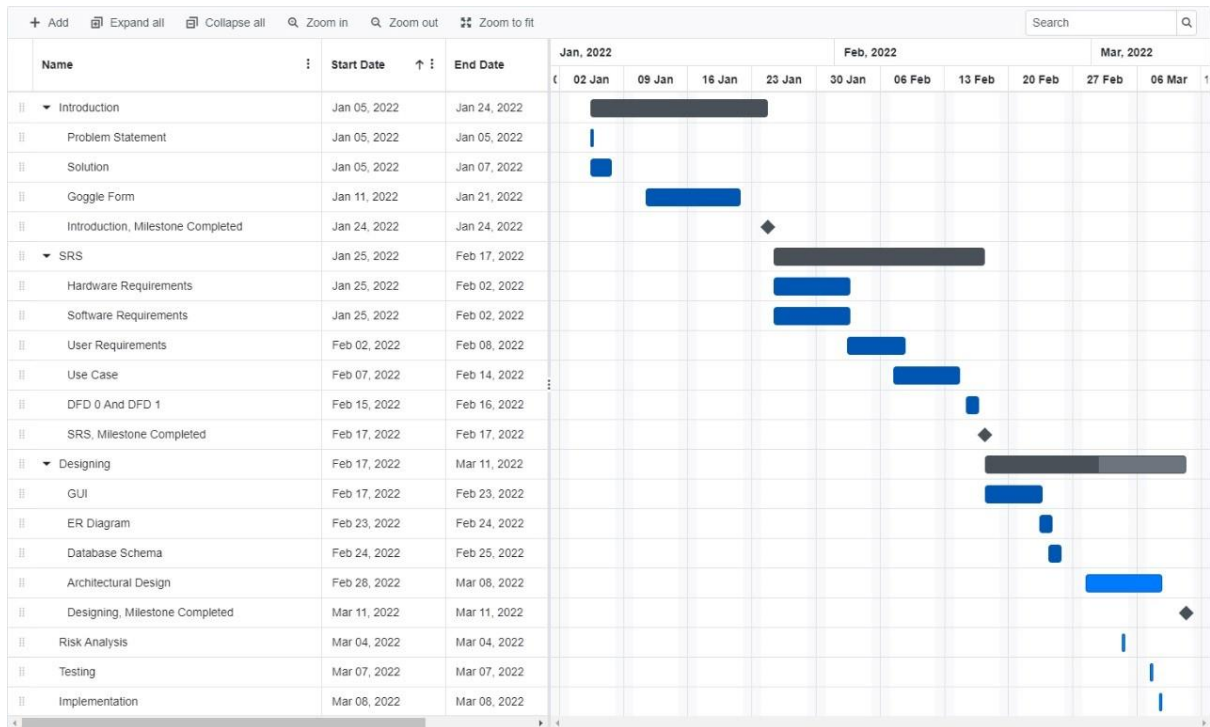
depend on and be sure of the outcome.

## REFERENCES:

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

# CH-3

## Gantt chart



# RISK ANALYSIS

Risks	Categories	Probability	Impact	RMMM
Wrong Login Info	PR	60%	Marginal	Remain on the same page a message will be display showing the Login info
Server Failure/Hacked	TR	10%	Catastrophic	Backup will be present to resolve this problem
Payment failures	TR	20%	Critical	User have to choose again the payment option
Booking cancelled but not updated	TR	30%	Critical	User will be shown a pop message after unbooking and

				unbooking and deduct the fine.
--	--	--	--	--------------------------------

--	--	--	--	--

One slot may be booked by two persons	PR	30%	Catastrophic	After one slot a confirmation message will be shown
Transaction slip not generated or slot is not boooked but money deducted	PR	20%	Critical	Refund will be provided to the user

# FUNCTIONAL POINTS



---

<b>USERS/ RESOURCE FILES</b>	<b>CASES</b>	<b>FUNCTIONAL UNITS</b>	<b>COMPL EXITY</b>
USER	LOGIN	EXTERNAL INPUT	HIGH
USER	SIGN-UP	EXTERNAL INPUT	LOW
USER	PAYMENT	EXTERNAL INPUT	AVG
USER	BOOKING SLOT	EXTERNAL INPUT	HIGH
USER	VIEW PROFILE	EXTERNAL ENQUIRY	AVG
USER	VIEW HISTORY	EXTERNAL ENQUIRY	LOW
USER	UPDATE PROFILE INFO	EXTERNAL INPUT	HIGH
USER	DELETE BOOKING	EXTERNAL INPUT	AVG
ADMIN	MAP	EXTERNAL ENQUIRY	LOW
USER	SEARCH SLOT	EXTERNAL INPUT	LOW
USER	REPORT	EXTERNAL	HIGH

USER	REPORT	EXTERNAL OUTPUT	HIGH
	PAYMENT RECEIPT	EXTERNAL ENQUIRY	HIGH

USER	VEHICLE SELECTION	EXTERNAL INPUT	AVG
	AVAILABILITY OF SLOT	INTERNAL LOGICAL FILE	LOW
ADMIN	PAYMENT CONFIRMATION	EXTERNAL ENQUIRY	LOW

ADMIN	PAYMENT CONFIRMATION	EXTERNAL ENQUIRY	LOW
ADMIN	DAILY ANALYSIS	EXTERNAL OUTPUT	LOW
USER/ADMIN	VIEW SLOTS	EXTERNAL ENQUIRY	AVG
ADMIN	BLOCK SLOT	EXTERNAL OUTPUT	AVG
ADMIN	PARKING CHARGES	EXTERNAL OUTPUT	AVG
DATA FILES	USER INFORMATION	EXTERNAL LOGICAL FILE	AVG
DATA FILES	TYPE OF VEHICLE	INTERNAL LOGICAL FILE	AVG
DATA FILES	PAYMENT	EXTERNAL LOGICAL FILE	HIGH

Information Domain Value	Count	Simple	Average	Complex	Count* Weight
External Inputs	8	3*2	4*3	8*3	42
External Output	4	4*1	5*2	0	14
External Inquiry	6	3*3	4*2	6*1	23
Internal Logical Files	2	7*1	10*1	1	17

External Logic Files	2	5	7*1	10*1	17
----------------------	---	---	-----	------	----

*Sum of Count\*Weight(count total)=113*

## Value Adjustment Factor

Ø Does the system require reliable backup and recovery? 4

Ø Are specialised data communication required to transfer information from the application? 0

Ø Are there disturbing processing functions? 2

Ø Is performance critical? 3

Ø Will the system run in an existing ,heavily utilised operational

environment? 4

Ø Does the system require online data entry? 5

Ø Does the online data entry require the input transaction to be built over multiple screens or operations? 3

Ø Are the ILFs updated online? 3

Ø Are the input, output, files or enquiry complex? 2

Ø Is the internal process complex? 4

Ø Is the code designed to be reusable? 3

Ø Are conversion and installation included in the design? 1

Ø Is the system designed for multiple installations in different organisations? 3

Ø Is the application designed to facilitate change and ease of use by the user? 3

Total =40

$FP = \text{count total} * [0.65 + 0.01 * fl]$

$FP = 113 * [0.65 + 0.01 * 40]$

$FP = 113 * 1.05$

$FP = 118.65$

# CH-4

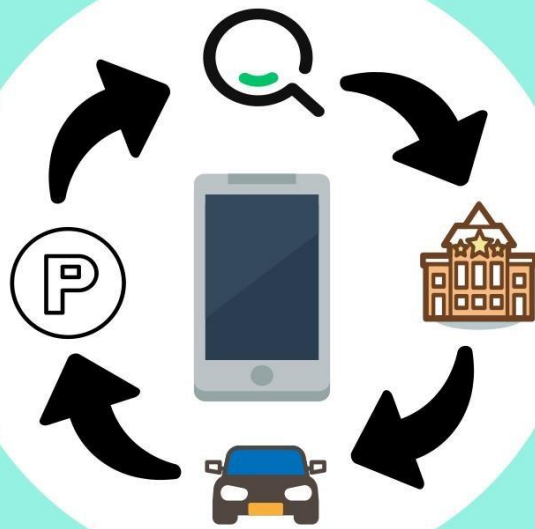
GUI



ParkVic



# ParkVic



...



## ParkVic



Trouble no more when booking venues  
for parking spot.

**Get Started**

# ParkVic



Log In

Register

# ParkVic



## Register

Show

☐ I agree with the terms and condition.

Sign Up

Forgot your password?

[Already registered](#)

# ParkVic



## Log In

Show

Log In

Forgot your password?

[Register](#)

# ParkVic



## CREATE NEW PASSWORD

New password



Confirm password



**RESET PASSWORD**



Search for a venue



BIKE



CAR

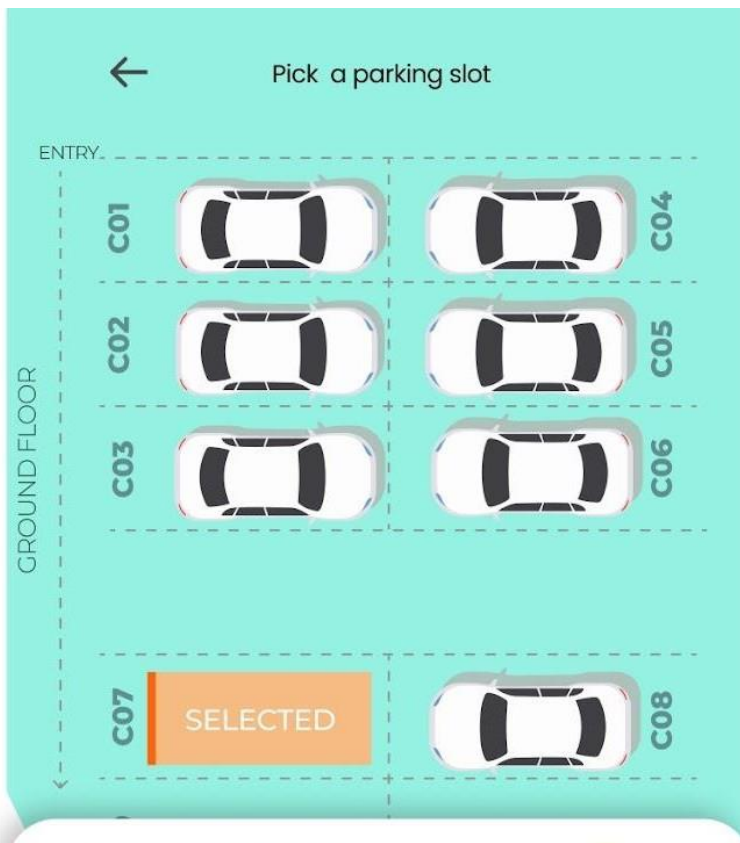


BUS



VAN

BOOK NOW



Reserve Parking at For more details

Call Directions Share

Arriving 22th JAN . 9.15PM [Change](#)

4 Spots Available  
 1.3 km away

Reserve for: RS XX  
For 24 hrs

Vehicle

TATA SUV

[Change](#)

Add Vehicle

Parking Spot

C07

[Change](#)

Payment

PAYTM

[Change](#)

Add Card



ASHISH

NEW DELHI, DELHI

NAME ASHISH

AGE 19

E-MAIL ASHI@GMAIL.COM

PHONE NO 99999xxxxx





**ASHISH**

NEW DELHI, DELHI

Your Previous Parking



1st JAN . 10:00AM



15th JAN . 11:30AM



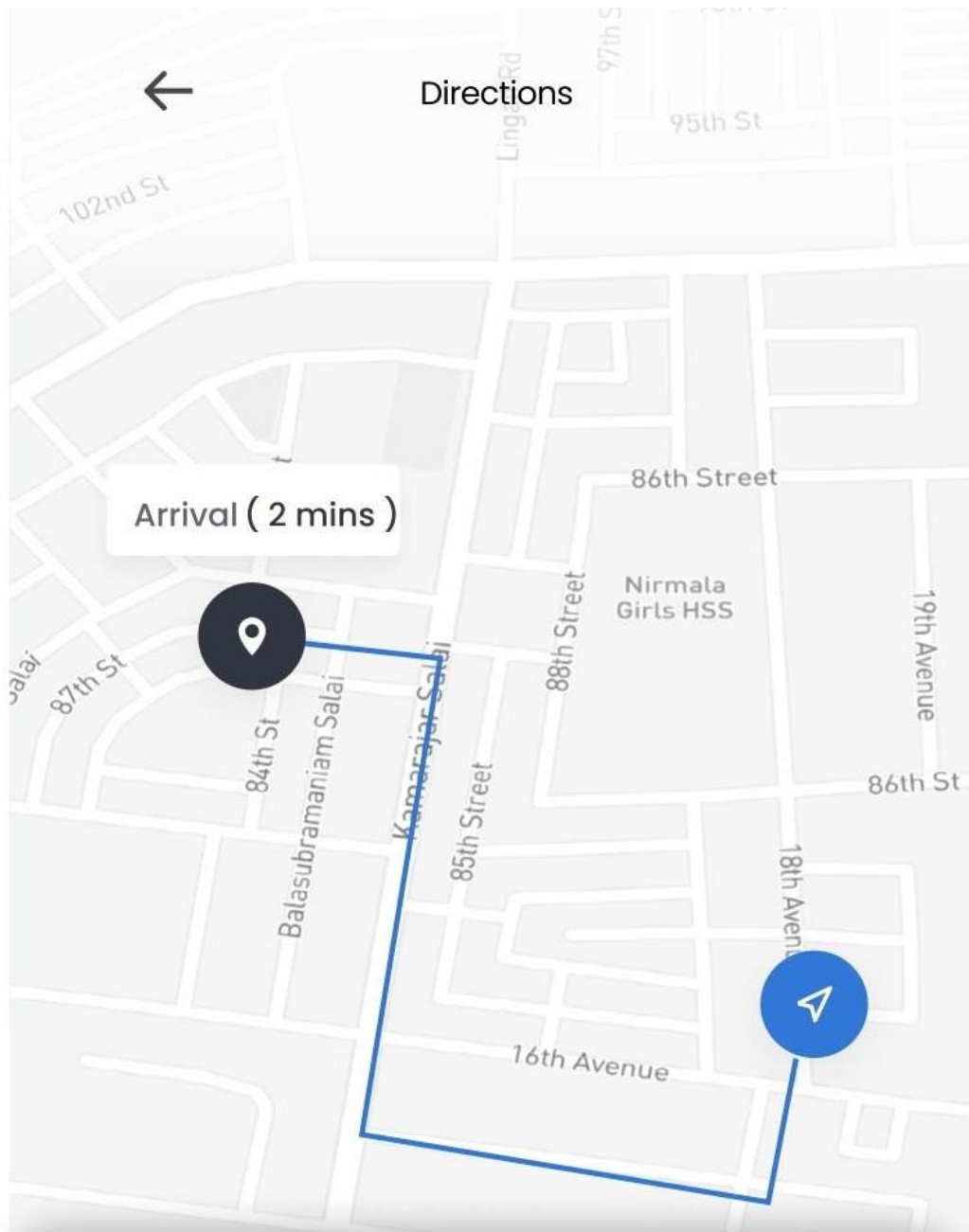
27th JAN . 1:30PM



5th FEB . 5:30AM



25th DEC . 8:30AM



 Reserve Parking at



[For more details](#)

EST: 16 min to Destination

**REROUTE DIRECTIONS**



## Reservation Summary



**Reserve Parking at**



[For more details](#)

Arriving	22th JAN . 9.15PM	✓
----------	-------------------	---

Duration	2 Hours	✓
----------	---------	---

Vehicle	TATA SUV	✓
---------	----------	---

Parking Spot	C07	✓
--------------	-----	---

Payment	PAYTM	✓
---------	-------	---

**CANCEL BOOKING**

# ParkVic



## OTP VERIFICATION

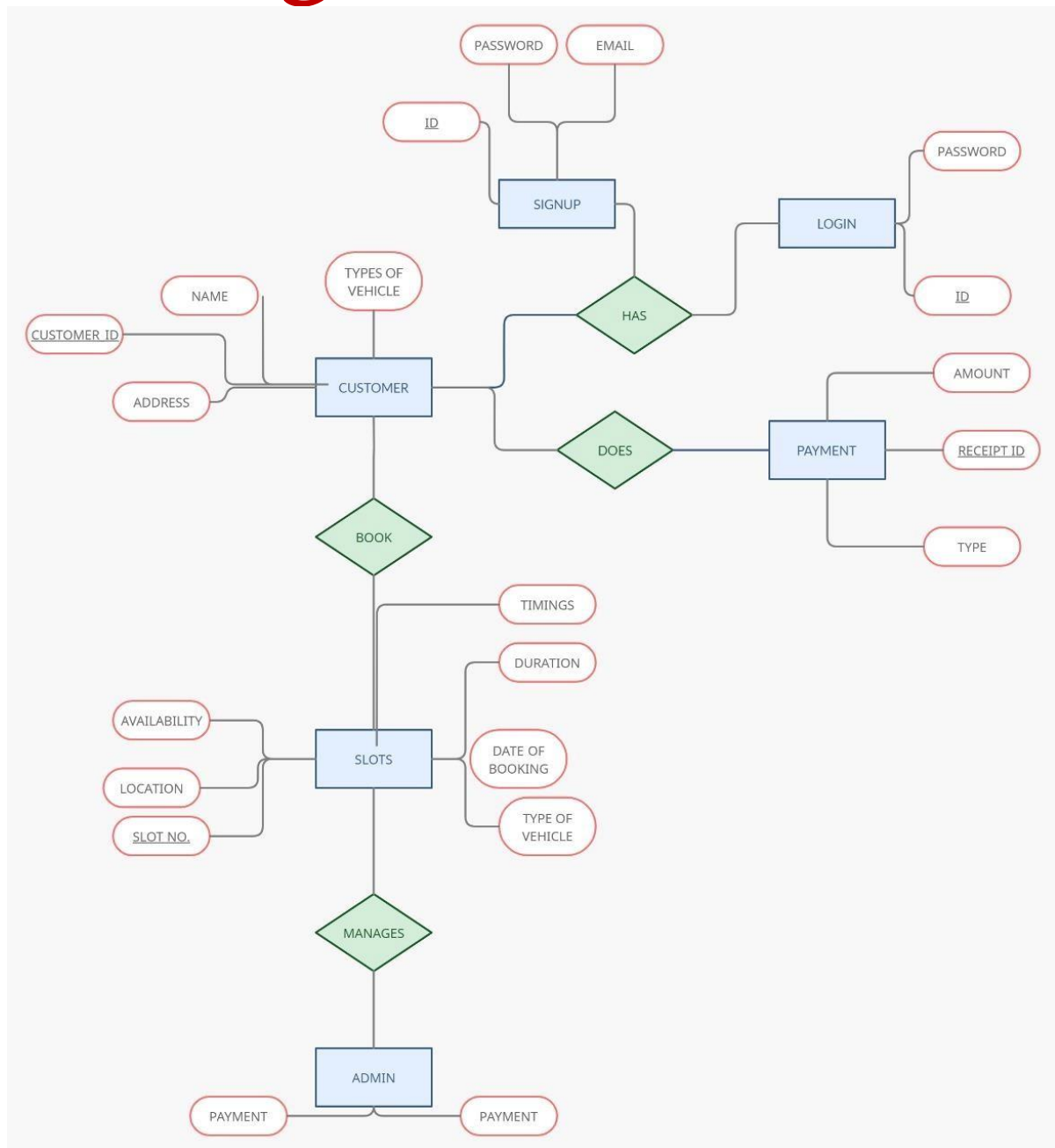
Enter the OTP sent to E-Mail

Don't receive the OTP? [RESEND OTP](#)

**VERIFY & PROCEED**

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
0		⌫

# Er diagram



# schema

Create schema parking;

Use parking;

Create table customer(

Name varchar(10) not null,

Address varchar(50) not null,

Type of vehicle varchar(10) not null,

Customer\_id char(10) not null primary key);

Create table login(

Id char(10) not null primary key,

Password varchar(15) not null,

Foreign key (password) references signup  
(password));

Create table signup (  
Name varchar(10) not null,  
Email varchar(10) not null primary key,  
Password varchar(15) not null );

Create table slots(  
Slot\_no int not null primary key,  
Location varchar(10) not null,  
Availability bool not null,  
Duration varchar(5) not null,  
Date of booking time\_stamp not null,  
Timings varchar(5) not null  
Foreign key(type of vehicle) references  
customer(type of vehicle));

```
Create table payment(  
Receipt_id varchar(10) not null primary key,  
Amount decimal(5,2) not null,  
Type varchar(10) not null  
Foreign key(Receipt_id) references  
slots(slot_no));
```

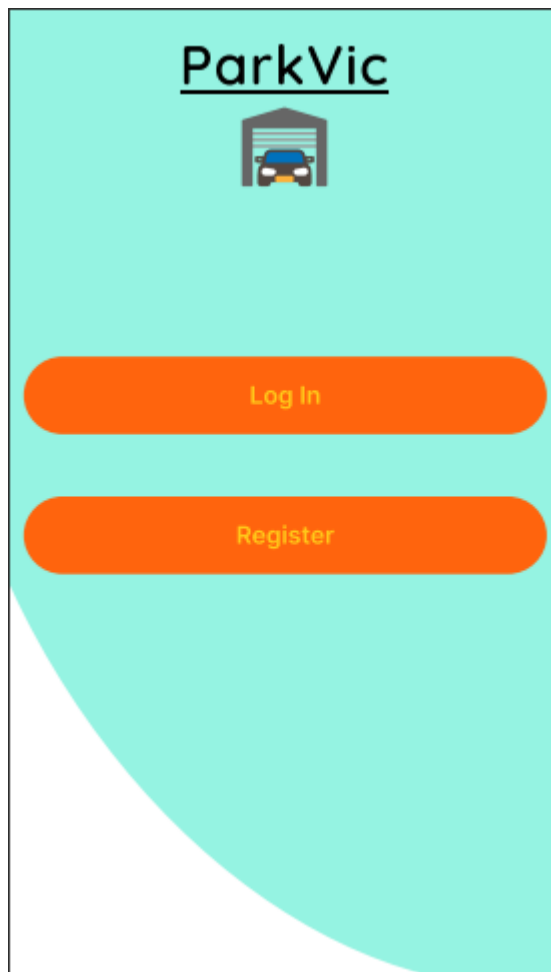


# ARCHITECTURAL DESIGN



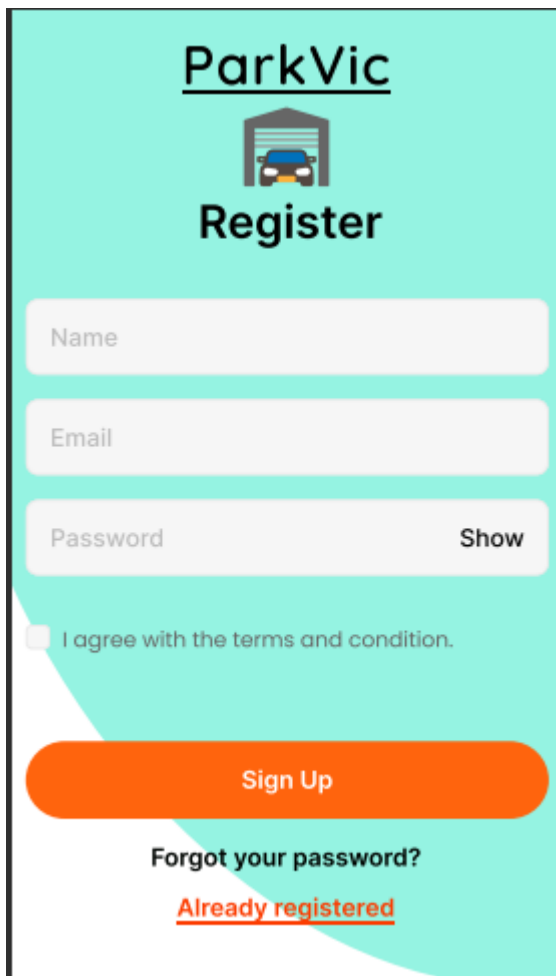
## 1. Index Page

```
if(button == "Get Started"){  
    go to sign-up page  
}  
else{  
    stay on the same page  
}
```




## 2. Log-in page

```
if(button == "Log In"){  
    go to Login page  
}  
else{  
    go to Registration page  
}
```



**ParkVic**



**Register**

Name

Email

Password  Show

☐ I agree with the terms and condition.

**Sign Up**

**Forgot your password?**

[Already registered](#)

### 3. Registration Page

```

Register(name, email, password){
    if(name)
        user=name
    if(email)
        mail_id=email
    if(password)
        pswd=password
    if(button == "Sign up"){
        store user, mail_id, pswd in the signup table in database
        go to OTP verification Page
    }
    else if
    {
        if(name==" ")
            show error
        if(email==" ")
            show error

        if(password==" ")
            show error
    }
}

```

```

else
    go to Login page
}

```



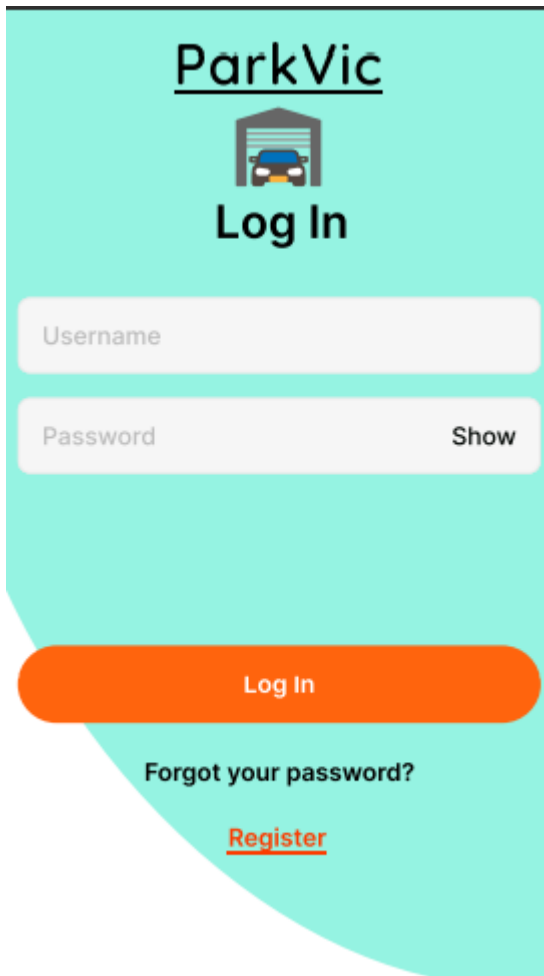
The image shows a mobile app interface for 'ParkVic'. At the top, the logo 'ParkVic' is displayed above a car icon. Below this, the text 'OTP VERIFICATION' is shown, followed by the instruction 'Enter the OTP sent to E-Mail'. There are four white square input fields for the OTP, with the first one highlighted by an orange border. Below the input fields, there is a link that says 'Don't receive the OTP? **RESEND OTP**'. A large orange button with the text 'VERIFY & PROCEED' is positioned below the link. At the bottom of the screen, there is a numeric keypad with digits 1-9, 0, and a backspace button (X).

#### 4. OTP verification Page

```

otp_generation(){
    otp = rand()%10000
    return otp
}
otp_verify(entered_otp){
    if(button == "Verify and Proceed"){
        if(otp_generation=entered_otp){
            go to Homepage
        }
    }
}

```




The image shows a login page for 'ParkVic'. At the top, the text 'ParkVic' is underlined. Below it is an icon of a car inside a garage. The text 'Log In' is centered below the icon. There are two input fields: 'Username' and 'Password'. The 'Password' field has a 'Show' button next to it. Below the input fields is a large orange rounded rectangle with the text 'Log In' inside. Below that is the text 'Forgot your password?' and a red underlined link 'Register'.


## 5. Login page


```
login(username, password){
  if(username)
    user=username
  if(password)
    pass=password
  if(button == "Forgot Password?"){
    go to Reset Password page
  }
  else{
    if(user=login.username AND pass=login.password){
      go to Homepage
    }
    else
      go to Registration page
  }
}
```

**ParkVic**



**CREATE NEW PASSWORD**

New password 

Confirm password 

**RESET PASSWORD**

## 6. Reset Password page

```

Reset_pswd(string){
    if(string != pswd(from registration page)){
        new_pswd = string
        return new_pswd
    }
    else
        return false
}

Confirm_pswd(new_pswd){
    if(string == new_pswd)
        update pswd to new_pswd in login table in database
}

```

## 7. Homepage

```

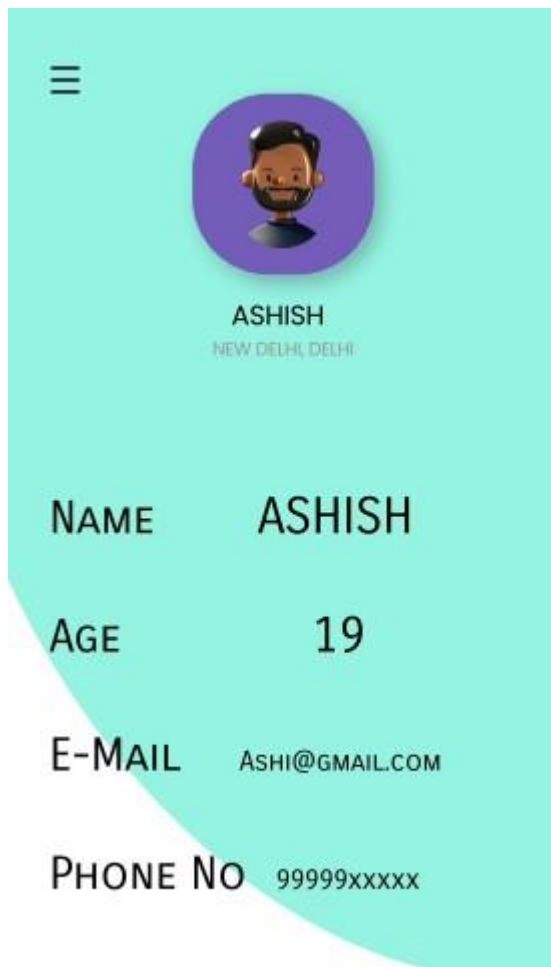
Home(){
    if(button == menu){
        show menu
        

- Available Parking Slots
- Payment page
- Directions
- Reservation Summary


    }
}

```

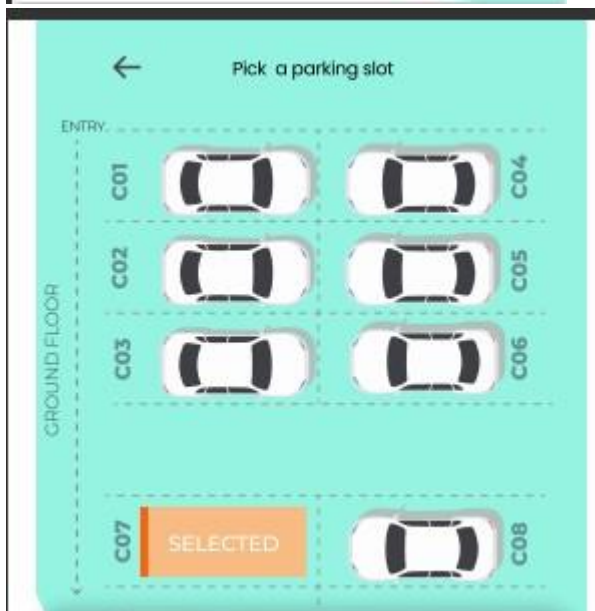
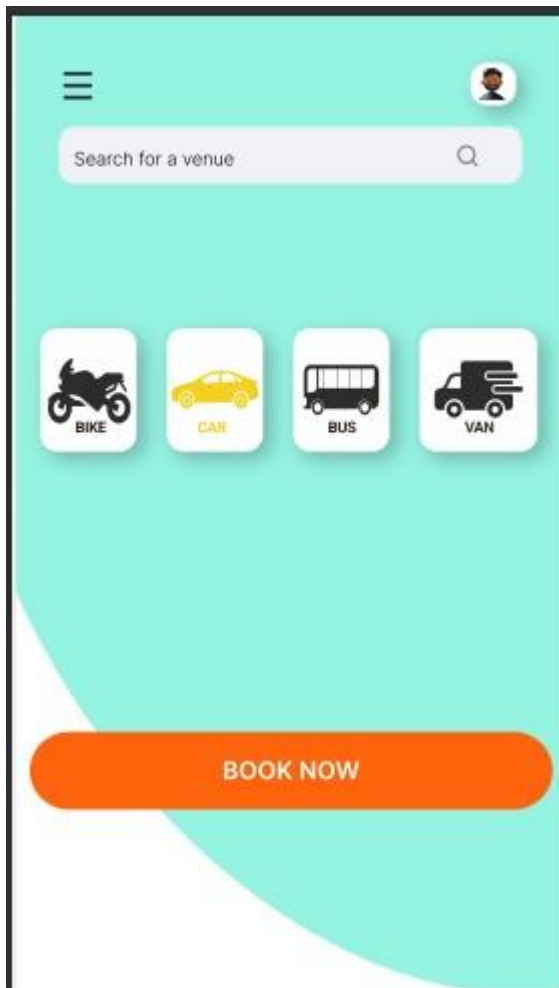
```
        if(button == profile){
            Go to Profile page
        }
    }
    details(destination, vehicle_type){
        if(destination)
            destination selected from the Type_of_vehicle in database
        if(vehicle_type)
            vehicle_type selected from the Location in database
        if(button == "Book Now"){
            go to Available Parking Slots Page
        }
    }
}
```



## 8. Profile Page

```
Profile(name, age, email, phone){
    if(name)
        name=name
    if(age>=18)
        age=age
    if(email)
        email=email
    if(phone)
        phone_no=phone
    store name, age, email, phone_no in the Signup in database
}
Profile_display(){
    Fetch and display all the information along with the previous parkings from the
Type_of_vehicle indatabase
}
```





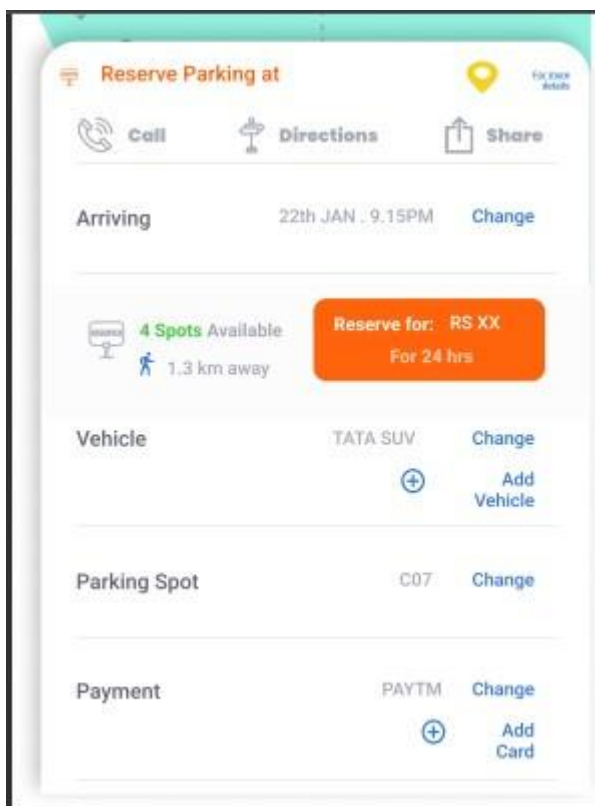
## 9. Available Parking Slots Page

```
Vehicle(vehicle_type){
    Select vehicle from the Type_of_vehicle in database based on vehicle type
}
Date_and_Time(date, time){
    if(date)
```

```

        Date=date
        if(time)
            Time=time
        Store date and time in the Slots.duration in database
    }
    parking_slots(space available){
        if(Slots.availability in == yes, according to Type_of_vehicle in &
Slots.Date_and_Time in,
        duration)
            go to Payment Page
        else
            return false
    }
}

```



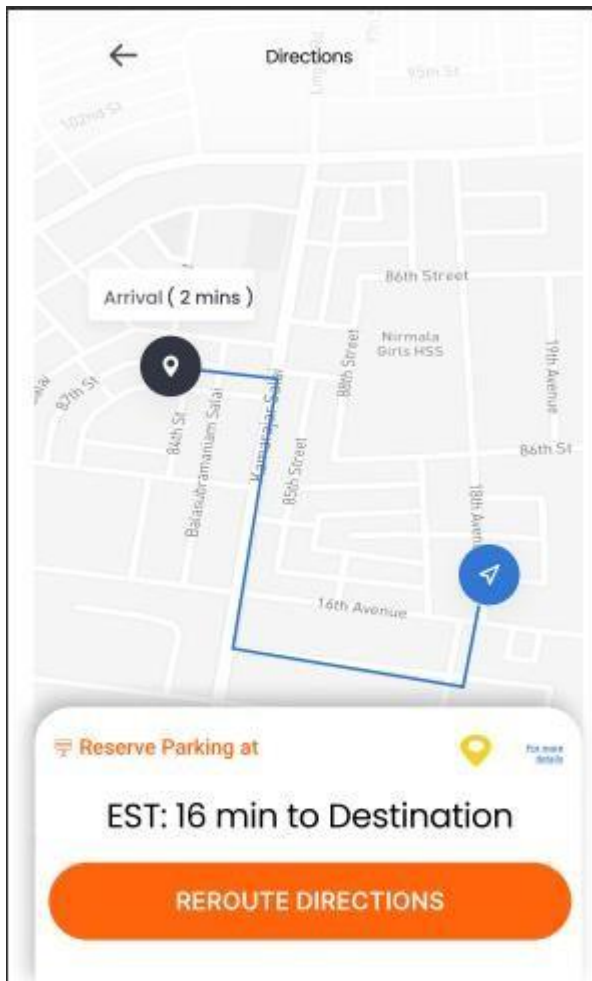
## 10.Payment Page

```

Payment(Payment.type_of_payment , Payment.amount ){
    Select mode pf payment from Payment in database
    • Paytm
    • UPI
    • Net Banking

    Make payment
}

```



## 11.Directions Page

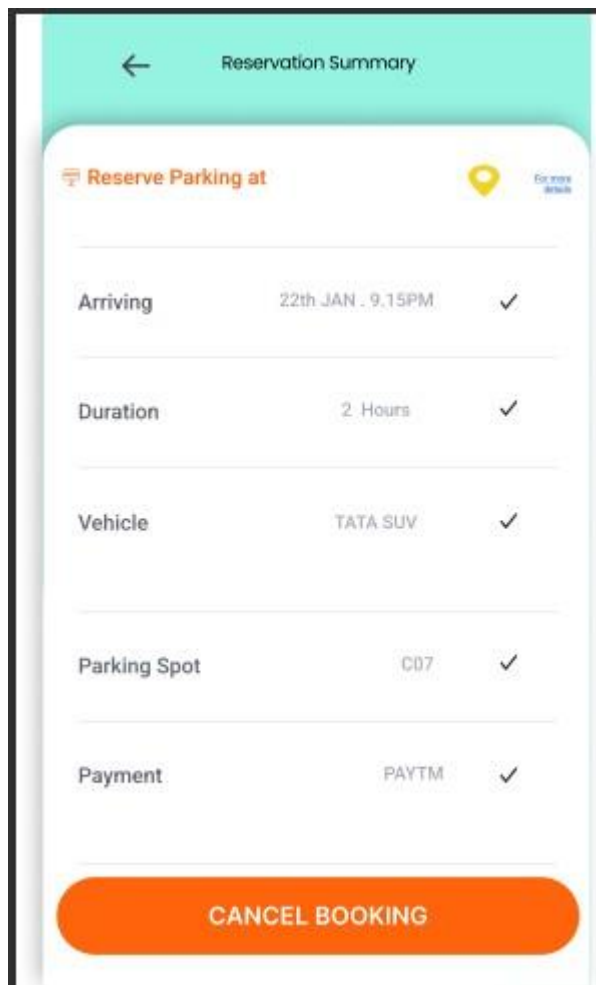
Route(destination, current location){

    Show the directions to the destination from Google maps

    if(button == "Reroute directions")

        Route(destination, current location)

}



## 12.Reservation Summary Page

```
Summary(Date_and_Time, Duration, vehicle, parking_Slot, mode_of_payment){
    Fetch and display all the information from the Slots.duration in, Type_of_vehicle in,
Payment in, Login in database
}
if(button == "Cancel booking"){
    Clear the data from all specified databases
}
```

# CH-5

MODULE	TEST CASE	EXPECTED OUTPUT
START UP PAGE	CLICK ON LOGIN PAGE	DIRECTED TO LOGIN PAGE
	CLICK ON REGISTER PAGE	DIRECTED TO REGISTER PAGE
REGISTER PAGE	FILL NAME,EMAIL,PASSWORD THEN CLICK ON REGISTER BUTTON	DATA STORED IN DATABASE
	ANY COLUMN REMAIN VACANT	STAY ON SAME PAGE SHOW ERROR
	WRONG EMAIL SYNTAX	SHOW ERRORS
	PASSWORD LENGTH<10	SHOW ERROR
	USER ALREADY REGISTERED	DIRECT TO LOGIN PAGE
LOGIN PAGE	VALID USERNAME AND PASSWORD CLICK ON LOGIN BUTTON	CHECK FROM DATABASE, DIRECT TO HOMEPAGE
	INVALID USERNAME	SHOW ERROR STAY ON SAME PAGE

	WRONG PASSWORD	GIVE ERROR STAY ON SAME PAGE
	CLICK ON FORGOT PASSWORD	DIRECT TO CREATE NEW PASSWORD PAGE
	UNREGISTERED USERNAME	SHOWING ERROR TO REGISTER FIRST
	CLICK ON REGISTER BUTTON	DIRECT TO REGISTER PAGE
<b>HOME PAGE</b>	CLICK ON HAMBURGER'S BUTTON	SHOW DIFFERENT OPTIONS OF MENU
	CLICK ON ANY OPTION IN MENU	MENU GET SELECTED AND THAT PAGE WILL BE OPEN
	CLICK ON PROFILE BUTTON	DIRECTED TO PROFILE PAGE

	CLICK ON HISTORY BUTTON	CHECK FROM DATABASE HISTORY IS PRESENT OR NOT, DIRECTED TO HISTORY PAGE
	CLICK ON SEARCH VENUE BUTTON	SHOE VENUES OPTIONS
	CHOOSE ANY VEHICLE	VEHICLE GETS CHOSEN
	CLICK ON IMAGE AT THE TOP RIGHT CORNER	PROFILE PAGE OPENS
	FIRST TIME USER CLICK ON HISTORY BUTTON	NO HISTORY WILL BE SHOWN
	CLICK ON BOOK NOW BUTTON	DIRECTED TO BOOK NOW PAGE
<b>BOOK NOW PAGE</b>	CHOSE THE VACANT SLOT	CHECK FROM DATABASE AND SLOT IS SELECTED
	TRY TO CHOOSE FILLED	UNABLE TO SELECT THE

	SLOT	FILLED SLOT
	CLICK ON CHOOSE VEHICLE TYPE BUTTON	ADD THE NAME AND DETAILS OF VEHICLE INFO STORED IN DATABASE
	CLICK ON CHANGE PARKING SLOT	DIRECTED TO SLOT PAGE
	VEHICLE AND SLOT ARE CHOSEN	DATA IS STORED IN DATABASE,BOOK NOW BUTTON IS ENABLED
	ANY EMPTY COLUMN	BOOK NOW BUTTON IS NOT ENABLED
	CLICK ON PAYMENT BUTTON	DIRECTED TO PAYMENT PAGE
<b>PAYMENT PAGE</b>	CLICK ON ADD CARD BUTTON	OPEN DETAILS FOR CARD TO ADD
	CLICK ON CHOOSE OTHER OPTION	OTHER OPTIONS LIKE GOGGLE PAY PHONE PAY WILL BE SHOWN
	CORRECT DETAILS ENTERED,CLICK ON BOOK BUTTON	CHECK FROM DATABASE.GO TO OTP VERIFICATION PAGE
	CORRECT OTP ENTERED	CHECK FROM DATABASE,

	CORRECT OTP ENTERED	CHECK FROM DATABASE, BACK TO PREVIOUS PAGE
	INCORRECT OTP	SHOW ERROR STAY ON SAME PAGE
	CLICK ON RESEND OTP	RESEND OTP TO USER
	PAYMENT DONE	TRANSACTION SLIP IS GENERATED AND SLOT
		BOOKED MESSAGE IS SHOWN
<b>CANCEL BOOKING PAGE</b>	CLICK ON CANCEL BOOKING BUTTON	CHECK FROM DATABASE SLOT IS BOOKED OR NOT, GO TO CANCEL BOOKING PAGE
	CLICK ON SELECTED SLOT	SLOT IS REJECTED

	CLICK ON UNBOOK	SLOT IS UNBOOKED,FINE IS DEDUCTED,UPDATE IN DATABASE

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['vpmsaid']==0)) {
    header('location:logout.php');
} else{

if(isset($_POST['submit']))
{
    $parkingnumber=mt_rand(100000000, 999999999);
    $catename=$_POST['catename'];
    $vehcomp=$_POST['vehcomp'];
    $vehreno=$_POST['vehreno'];
    $ownername=$_POST['ownername'];
    $ownercontno=$_POST['ownercontno'];
    $enteringtime=$_POST['enteringtime'];

    if($catename=="0")
        echo "<script>alert('select vehicle cateory!');</script>";
    else{
        $query=mysqli_query($con, "insert
into tblvehicle(ParkingNumber,VehicleCategory,VehicleCompanyname,Reg
istrationNumber,OwnerName,OwnerContactNumber)
value('$parkingnumber','$catename','$vehcomp','$vehreno','$ownername'
,'$ownercontno')");
        if ($query) {
echo "<script>alert('Vehicle Entry Detail has been
added');</script>";
echo "<script>>window.location.href = 'manage-
incomingvehicle.php'</script>";
        }
        else
        {
            echo "<script>alert('Something Went Wrong. Please try
again.');

```



```

    }

}

?>
<!doctype html>
<html class="no-js" lang="">
<head>

    <title>VPMS - Add Vehicle</title>

    <link rel="apple-touch-icon"
href="https://i.imgur.com/QRAUqs9.png">
    <link rel="shortcut icon" href="https://i.imgur.com/QRAUqs9.png">

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/normalize.min.
css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap
.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-
awesome@4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-icon@1.2.3/pe-
icon-7-stroke/dist/pe-icon-7-stroke.min.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-elastic.css">
    <link rel="stylesheet" href="assets/css/style.css">

    <link
href='https://fonts.googleapis.com/css?family=Open+Sans:400,600,700,8
00' rel='stylesheet' type='text/css'>

    <!-- <script type="text/javascript"

```

```
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.min.js"></script> -->
```

```
</head>
```

```
<body>
```

```
<?php include_once('includes/sidebar.php');?>
```

```
<!-- Right Panel -->
```

```
<?php include_once('includes/header.php');?>
```

```
<div class="breadcrumbs">
```

```
<div class="breadcrumbs-inner">
```

```
<div class="row m-0">
```

```
<div class="col-sm-4">
```

```
<div class="page-header float-left">
```

```
<div class="page-title">
```

```
<h1>Dashboard</h1>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col-sm-8">
```

```
<div class="page-header float-right">
```

```
<div class="page-title">
```

```
<ol class="breadcrumb text-right">
```

```
<li><a
```

```
href="dashboard.php">Dashboard</a></li>
```

```
<li><a href="add-
```

```
vehicle.php">Vehicle</a></li>
```

```
<li class="active">Add
```

```
Vehicle</li>
```

```
</ol>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="content">
```

```
<div class="animated fadeIn">
```

```

<div class="row">
    <div class="col-lg-6">
        <div class="card">

            </div> <!-- .card -->

        </div><!--/.col-->

    <div class="col-lg-12">
        <div class="card">
            <div class="card-header">
                <strong>Add </strong> Vehicle
            </div>
            <div class="card-body card-block">
                <form action="" method="post"
enctype="multipart/form-data" class="form-horizontal">
                    <p style="font-size:16px;
color:red" align="center"> <?php if($msg){
    echo $msg;
} ?> </p>

                    <div class="row form-group">
                        <div class="col col-md-
3"><label for="select" class=" form-control-
label">Select</label></div>

                        <div class="col-12 col-md-9">
                            <select name="catename"
id="catename" class="form-control">

                                <option
value="0">Select Category</option>

                                <?php
$query=mysqli_query($con,"select * from tblcategory");
                                while($row=mysqli_fetch_array($query))
                                {
                                    ?>

                                    <option value="<?php
echo $row['VehicleCat'];?>"><?php echo $row['VehicleCat'];?></option>
                                    <?php } ?>

```

```

        </select>
    </div>
</div>
<div class="row form-group">
    <div class="col col-md-
3"><label for="text-input" class=" form-control-label">Vehicle
Company</label></div>
        <div class="col-12 col-md-
9"><input type="text" id="vehcomp" name="vehcomp" class="form-
control" placeholder="Vehicle Company" required="true"></div>
    </div>

    <div class="row form-group">
        <div class="col col-md-
3"><label for="text-input" class=" form-control-label">Registration
Number</label></div>
        <div class="col-12 col-md-
9"><input type="text" id="vehreno" name="vehreno" class="form-
control" placeholder="Registration Number" required="true"></div>
    </div>
    <div class="row form-group">
        <div class="col col-md-
3"><label for="text-input" class=" form-control-label">Owner
Name</label></div>
        <div class="col-12 col-md-
9"><input type="text" id="ownername" name="ownername" class="form-
control" placeholder="Owner Name" required="true"></div>
    </div>
    <div class="row form-group">
        <div class="col col-md-
3"><label for="text-input" class=" form-control-label">Owner Contact
Number</label></div>
        <div class="col-12 col-md-
9"><input type="text" id="ownercontno" name="ownercontno"
class="form-control" placeholder="Owner Contact Number"
required="true" maxlength="10" pattern="[0-9]+"></div>
    </div>

    <p style="text-align: center;">
<button type="submit" class="btn btn-primary btn-sm" name="submit"

```

```
>Add</button></p>

        </form>
    </div>

</div>

</div>

<div class="col-lg-6">

</div>

</div>

</div>

    </div><!-- .animated -->
</div><!-- .content -->

<div class="clearfix"></div>

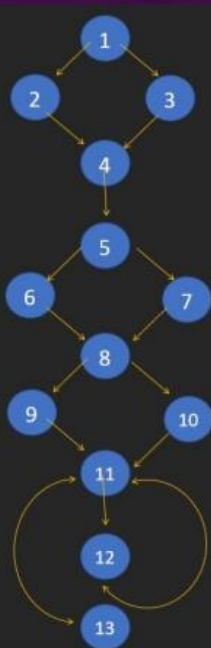
<?php include_once('includes/footer.php');?>

</div><!-- /#right-panel -->

<!-- Right Panel -->

<!-- Scripts -->
<script
src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery.min.js"></
script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/umd/popper.mi
n.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.m
in.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery-match-
height@0.7.2/dist/jquery.matchHeight.min.js"></script>
<script src="assets/js/main.js"></script>
```

```
</body>  
</html>  
<?php } ?>
```



1).  $ec = E - N + 2$   
 $E = 16 \quad N = 13$

$$ec = 16 - 13 + 2$$
$$ec = 5$$

2). Total No of Region = 5

3).  $V(G) = P + 1$  (WHERE P IS TOTAL NO OF PREDICATE NODES CONTAINED IN FLOW G)

$$P = 4$$
$$V(G) = 4 + 1 = 5$$

The Cyclometric Complexity is 5 for Given Code.

## REFERENCES:

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

[Template.](#)

2. Aggarwal, K. K., Software Engineering