# 1. INTRODUCTION

## 1.1 PROJECT PURPOSE AND SCOPE

The scope of the project is the system has many options and features when it comes to Booking a Parking Site and Security. This System can be implemented anywhere who wants to Rent their parking site. This can be also used in Commercial or Public places where there is chaos for parking.

Parking is an issue in many countries due to lack of parking places, to overcome this many countries have come up with dedicated parking sites. We propose a complete parking System for the Parking sites. This system can help User as well as the Parking sites have a good management of booking and managing the sites. This system has one user and admin. Admin who is responsible to manage Parking sites and user is the car owner who is in need of Parking. It also have a camera section.

# 2. SYSTEM ANALYSIS

## 2.1 SYSTEM STUDY

### 2.1.1 CURRENT SYSTEM

The current system is difficult to find a parking location that have a free slot. We can find location easily by GPS. But we can't find there is free space or available space to park our vehicle.

### 2.1.2 PROPOSED SYSTEM

This application is designed to illuminate the problem of the current system. The proposed system admin managing the parking location and update the location and sites. User can view the locations around 1km. User can also view the free slots and can book the slot.

# 3. PROJECT PLAN

## 3.1 SCOPE MANAGEMENT

Parking management system will cover the following areas

- View location
- View available slot
- Booking free slot
- Payment

## 3.2 PEOPLE MANAGEMENT

**OWNERS**

- Fida Shirin N
- Rahina Banu O
- FathwaBinth Umer
- Nihana Sherin P

## 3.3 RISKMANAGEMENT PLAN

The decisions by which to make modifications to the project plan should be coordinated using the following process:

Step 1: As soon as change which impacts project scope, managing parking location is identified, the project management team will document the issue.

Step 2: The project management team will review the change and determine the associated impact to the project and will forward the issue, along with a recommendation, to weekly technical call for review and decision.

## 3.4 COMMUNICATION MANAGEMENT

This plan provides a framework for informing, involving and obtaining buy in from all participants throughout duration of the project.

# 4. FEASIBILITY ANALYSIS

## 4.1 TECHNICAL FEASIBILITY

In this type of feasibility, the present hardware and software compatibility with the new one is checked to run the new system.

Findings:

The computer have sufficient configuration (hardware and software) to adopt new software along with present configuration.

## 4.2 ECONOMIC FEASIBILITY

In this type of feasibility, the cost of hardware, software and overall budget is evaluated to the new system. Tangible and intangible benefits are also considered in evaluation.

Findings:

Estimated cost of our project, car parking system is rupees 50,000.

## 4.3 OPERATIONAL FEASIBLILTY

In this type of feasibility, researcher has checked the issue like, operational scope for the fast acceptability of the alternative solution, human resources issues, social issues (organizational conflicts) and legal issues, etc.

# 5. REQUIREMENT SPECIFICATIONS

Requirement specification is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

## 5.1 SOFTWARE SPECIFICATION

| | | |
|---|---|---|
| Front end | - | JSP+ Android |
| Back end | - | MySQL server |
| Web server | - | WAMPServer |
| Operating system | - | Windows10 |

## 5.2 HARDWARE SPECIFICATION

- Processor : Intel(R) Core(TM)i3-7100U CPU @ 2.40GHz 2.40GHz
- RAM : 4GB
- Monitor
- Pen drive : 16GB
- Keyboard
- Mouse
- Smart phone

## 5.3 FUNCTIONAL REQUIREMENTS

These are the functions done by system such as show locations, show free slots, and done payments book the slot etc..

- Show locations
- Show free slot
- We can book free slot
- We can done payment

## 5.4 NON-FUNCTIONAL REQUIREMENTS

- Usability
  Any familiar in using windows operation can operate the system since it is have user friendly user interface. Which is a self-directive application and then can use the system without any ambiguity.
- Reliability
  The Parkimine system is available based on the user needs. The Parkimine system is password protected. The locations and slots controlled by only the administrator.
- Performance
  The Parkimine management system operates its function in small amount of time and can be accessed by user. To access system must login in system.
- Implementation
  The system is implemented in: Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz 2.40GHz

- User interface
  The user interface is friendly which is easy to use.
- Supportability
  The Parkimine system operates in any android version. And the admin module operates in any version of windows operating system.
- Operation
  The Parkimine management system is operated and controlled by the admin for safe work.

## 5.5 TECHNOLOGY USED

**ANDROID:**

Android is a mobile operating system developed by Google. It is used by several smart phones and tablets. The Android operating system (OS) is based on the Linux kernel.

Main features of Android are:

- Messaging
- Web
- Browser
- Voice-Based features
- Multi-touch
- Multitasking
- Screen capture
- TV recording
- Video calling
- Multiple language support
- Accessibility

**JSP:**

Java Server Page (JSP) is a technology for controlling the content or appearance of Web pages through the use of servlets, small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it. Sun Microsystems, the developer of Java, also refers to the JSP technology as the Servlet application program interface (API). JSP is comparable to Microsoft's Active Server Page (ASP) technology. Whereas a Java Server Page calls a Java program that is executed by the Web server, an Active Server Page contains a script that is interpreted by a script interpreter (such as VBScript or JScript) before the page is sent to the user. An HTML page that contains a link to a Java servlet is sometimes given the file name suffix of .JSP

Main features of jsp are:

- Extension to servlet
- Powerful
- Portable
- Flexible
- Easy to read data
- Easy to maintain

**SQL:**

The Structured Query Language (SQL) comprises one of the fundamental building blocks of modern database architecture. SQL is an ANSI (American National Standard Institution) standard computer language for accessing and manipulating database systems. SQL statements are used retrieve and update data in a database. SQL works with database programs like MS Access, Oracle, DB2, Informix, MS Server and Sybase etc.

A database most often contains one or more tables. Each table is identified by a name (E.g. "User"). A table contains record (rows) with data. With SQL we can query a database and have a result set returned. SQL is the syntax for executing queries. But the SQL language also includes the syntax to insert and delete records. These query and update commands together form the Data Manipulation Language (DML) part of SQL. The Data Definition Language (DDL) part of SQL permits database tables to be created or detected. We can also define indexes (keys), specify links between tables and imposes constraints between databases.

# 6. MODULE DESCRIPTION

There are basically three modules in this project

- Admin module
- User module
- Camera section

## 6.1 ADMIN MODULE

- Login
- Add/delete parking locations
- View location
- Add/delete slots
- View slots
- View user
- View feedback
- Send notification
- View notification
- View booking
- View payments
- View parked vehicles

## 6.2 USER MODULE

- Registration
- Login
- View parking location
- View slots
- Booking
- Enter feedback
- View notification
- Payment

## 6.3 CAMERA SECTION

- Number detect and read number when Check-in
- Number detect and read number when Check-out

# 7. SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy special requirements. System design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

## 7.1 TYPES OF SYSTEM DESIGN

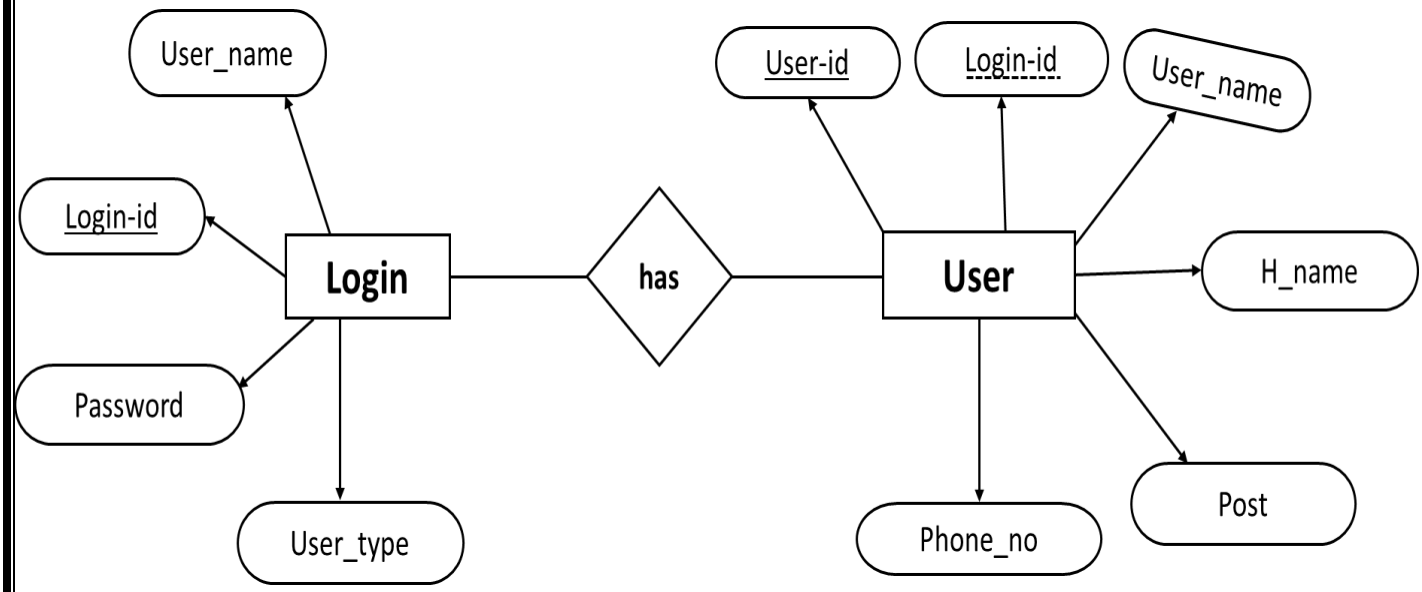
### 7.1.1 LOGICAL DESIGN

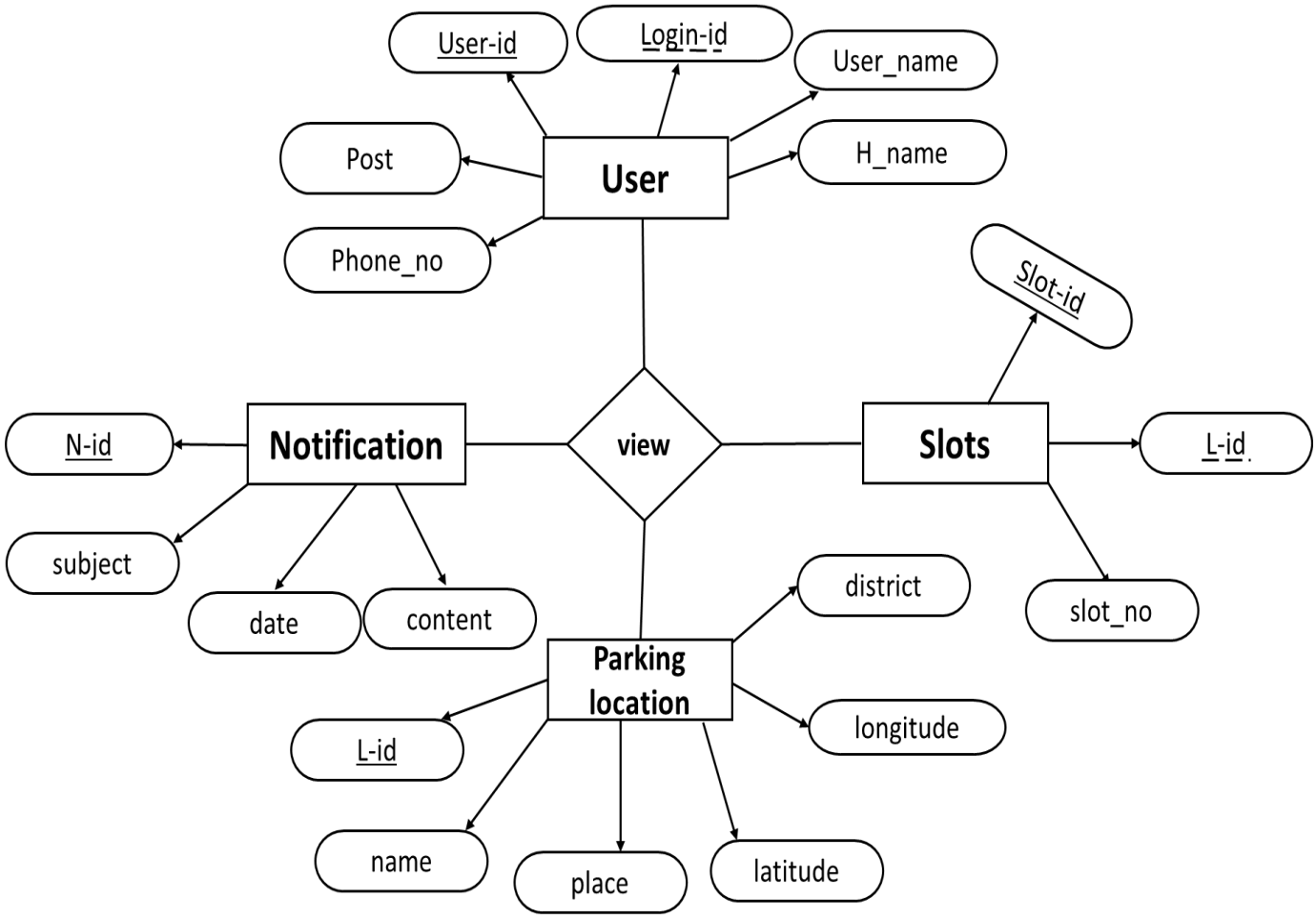
The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system.

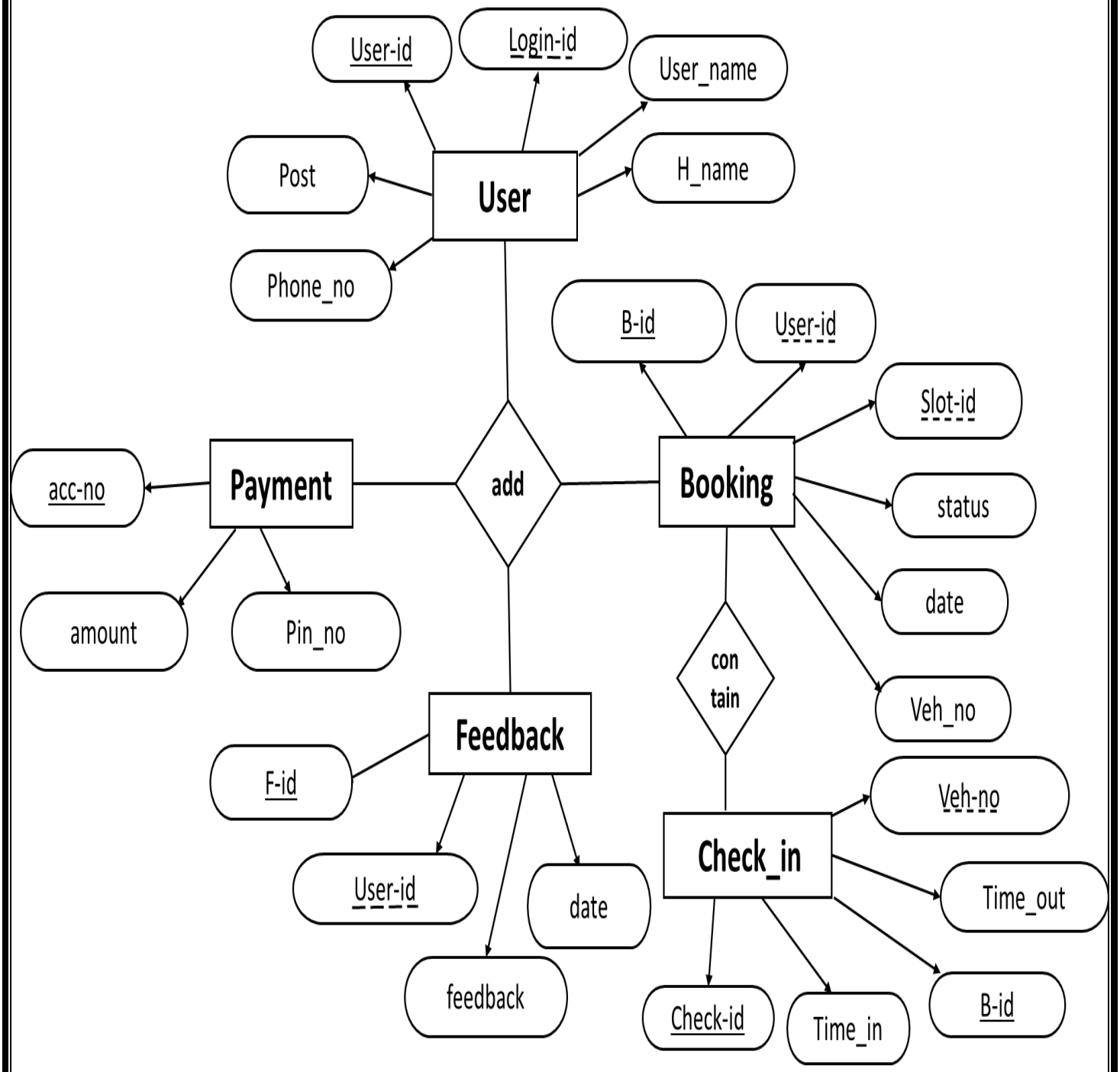
Logical design is graphical representation of a system showing the systems processes and the flows of data into and out of the processes. We use logical design to document information systems because we can represent the logical nature of a system-what task the system is doing, without having to specify how, where or by whom the tasks are accomplished. What a system is doing will change less over time than how it is doing it.

To represent the logical design of a system we can use different diagrams like ER diagram and DFD.

### 7.1.1.1 ER DIAGRAM

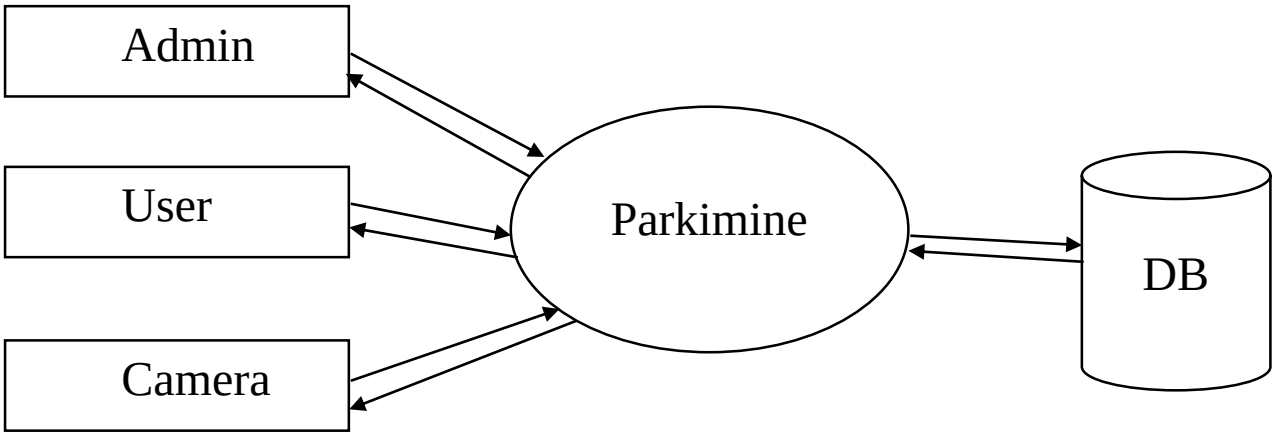



User-id
Login-id
User_name
Post
User
H_name
Phone_no
B-id
User-id
Slot-id
acc-no
Payment
add
Booking
status
date
amount
Pin_no
con
tain
Veh_no
Feedback
F-id
Veh-no
Check_in
User-id
date
Time_out
feedback
Check-id
Time_in
B-id

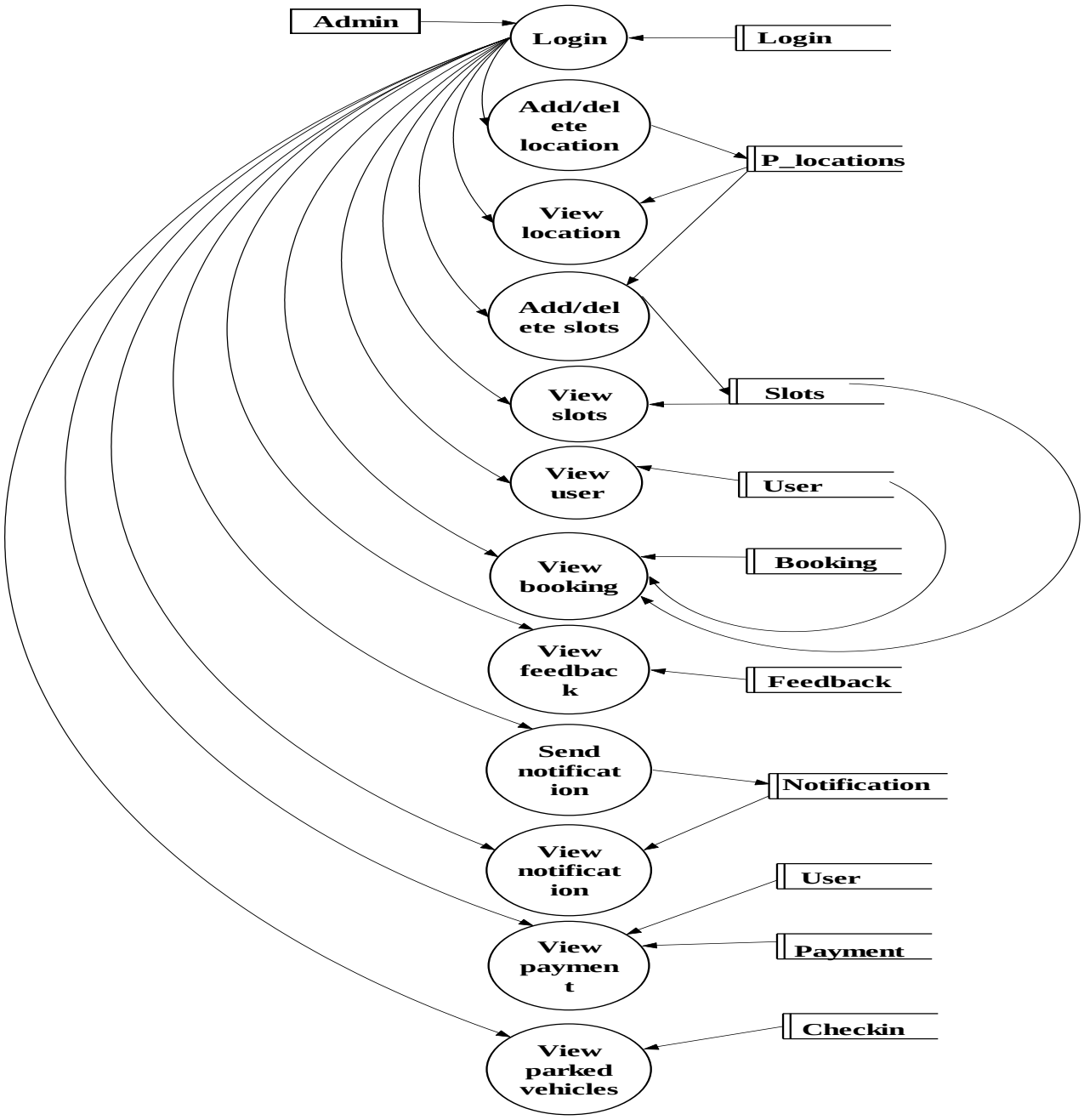
#### 7.1.1.2 DFD

**Level 0**

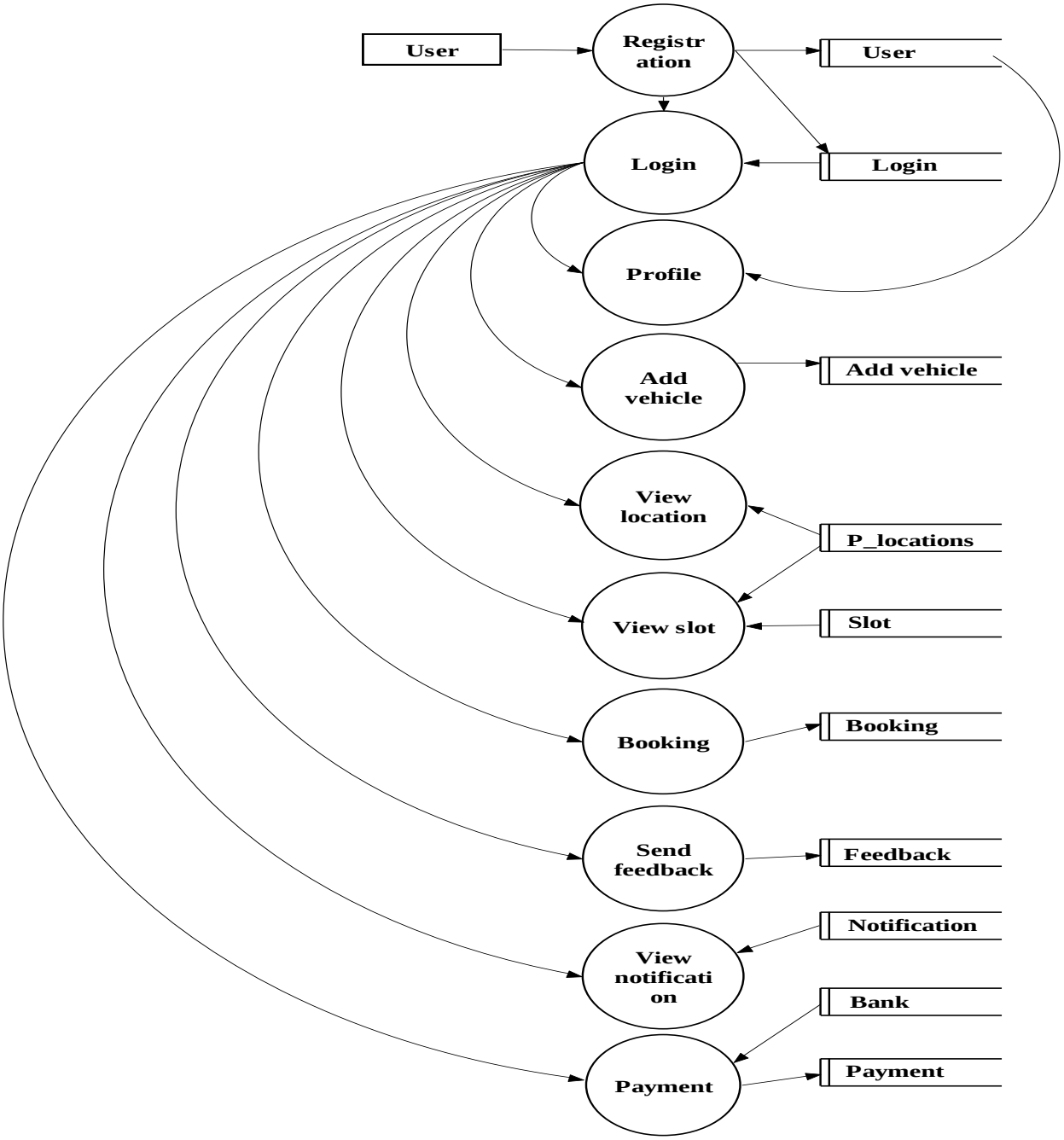


**Level 1.1**

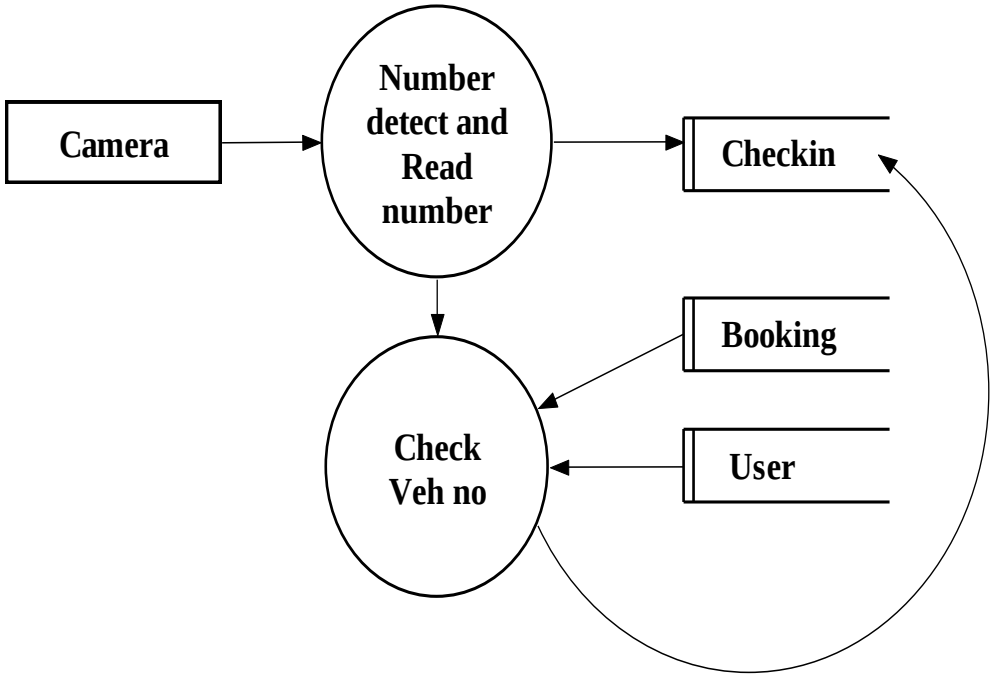
Admin:



User:


User
Registr ation
User
Login
Login
Profile
Add vehicle
Add vehicle
View location
P_locations
View slot
Slot
Booking
Booking
Send feedback
Feedback
Notification
View notificati on
Bank
Payment
Payment

Camera:


Camera
Number detect and Read number
Checkin
Booking
Check Veh no
User

### 7.1.2 PHYSICAL DESIGN

The physical design is the graphical representation of a system showing the systems internal and external entities, and the flow of data into and out of these entities.

The physical design can be broken into three sub-tasks:

1. User interface diagram
2. Database design
3. Process design

#### 7.1.2.1 DATABSE DESIGN

Database design is concerned with how the data represented and stored within the system.

**TABLE DESIGN**

TABLE NAME: LOGIN
DESCRIPTION: USER LOGIN ACCOUNT DETAILS
PRIMARY KEY: LOGIN_ID

| Field name | Data type | Constraints |
|---|---|---|
| Login_id | Int | Primary key, Auto increment |
| User_name | Varchar | Not null |
| Password | Varchar | Not null |
| User_type | Varchar | Not null |

TABLE NAME: USER
DESCRIPTION: USER DETAILS
PRIMARY KEY: USER_ID

| Field name | Data type | Constraints |
|---|---|---|
| User_id | Int | Primary key, Auto increment |
| Name | Varchar | Not null |
| H_name | Varchar | Not null |
| Post | Varchar | Not null |
| District | Varchar | Not null |
| Phno | Varchar | Not null |
| Login_id | Int | Foreign key |

TABLE NAME: P_LOCATIONS
DESCRIPTION: PARKING LOCATIONS
PRIMARY KEY: LOC_ID

| Field _name | Data type | Constraints |
|---|---|---|
| Loc_id | Int | Primary key, Auto increment |
| Loc_name | Varchar | Not null |
| Place | Varchar | Not null |
| District | Varchar | Not null |
| Latitude | Varchar | Not null |
| Longitude | Varchar | Not null |

TABLE NAME: ADD_VEHICLE
DESCRIPTION: USER VEHICLE DETAILS
PRIMARY KEY: VEH_ID

| Field name | Data type | Constraints |
|---|---|---|
| Veh_id | Int | Primary key, Auto increment |
| Login_id | Int | Foreign key |
| Veh_no | Varchar | Not null |
| Veh_type | Varchar | Not null |

TABLE NAME: BOOKING
DESCRIPTION: USER BOOKING DETAILS
PRIMARY KEY: B_ID

| Field name | Data type | Constraints |
|---|---|---|
| B_id | Int | Primary key, Auto increment |
| Veh_id | Int | Foreign key |
| Slot_id | Int | Foreign key |
| Date | Date | Not null |
| Status | Varchar | Not null |
| Time | Time | Not null |

TABLE NAME: CHECK_IN
DESCRIPTION: USER CHECK IN DETAILS
PRIMARY KEY: CHECK_ID

| Field name | Data type | Constraints |
|---|---|---|
| Check_id | Int | Primary key, Auto increment |
| B_id | Int | Foreign key |
| Time_in | Time | Not null |
| Time_out | Time | Not null |

TABLE NAME: FEEDBACK
DESCRIPTION: USER FEEDBACK DETAILS
PRIMARY KEY: F_ID

| Field name | Data type | Constraints |
|---|---|---|
| F_id | Int | Primary key, Auto increment |
| Login_id | Int | Foreign key |
| Date | Date | Not null |
| Feedback | Varchar | Not null |

TABLE NAME: NOTIFICATION
DESCRIPTION: NOTIFICATION DETAILS
PRIMARY KEY: N_ID

| Field name | Data type | Constraints |
|---|---|---|
| N_id | Int | Primary key, Auto increment |
| Subject | Varchar | Not null |
| Date | Date | Not null |
| Content | Varchar | Not null |

TABLE NAME: SLOT
DESCRIPTION: LOCATION SLOT DETAILS
PRIMARY KEY: SLOT_ID

| Field name | Data type | Constraints |
|---|---|---|
| Slot_id | Int | Primary key, Auto increment |
| Slot_no | Varchar | Not null |
| Status | Varchar | Not null |
| Loc_id | Int | Foreign key |

TABLE NAME: TEMP_DATA
DESCRIPTION: TEMPARORY CHECK-IN DETAILS
PRIMARY KEY: ID

| Field name | Data type | Constraints |
|---|---|---|
| Id | Int | Primary key, Auto increment |
| Photo | Varchar | Not null |
| Vno | Varchar | Not null |
| Status | Varchar | Not null |
| Date | Date | Not null |
| Time | Time | Not null |

TABLE NAME: UK_USER
DESCRIPTION: NOT BOOKING DETAILS
PRIMARY KEY: UK_ID

| Field name | Data type | Constraints |
|---|---|---|
| Uk_id | Int | Primary key, Auto increment |
| Veh_no | Varchar | Not null |
| Photo | Varchar | Not null |
| Date | Date | Not null |
| Time | Time | Not null |

TABLE NAME: PAYMENT
DESCRIPTION: USER PAYMENT DETAILS
PRIMARY KEY: PAYMENT_ID

| Field name | Data type | Constraints |
|---|---|---|
| Payment_id | Int | Primary key, Auto increment |
| Ac_no | Varchar | Not null |
| Amount | Varchar | Not null |
| Check_id | Int | Foreign key |

TABLE NAME: BANK
DESCRIPTION: BANK NAME
PRIMARY KEY: BANK_ID

| Field name | Data type | Constraints |
|---|---|---|
| Bank_id | Int | Primary key, Auto increment |
| Name | Varchar | Not null |

TABLE NAME: BRANCH
DESCRIPTION: BANK DETAILS
PRIMARY KEY: BANK_ID

| Field name | Data type | Constraints |
|---|---|---|
| Br_id | Int | Primary key, Auto increment |
| Branch_name | Varchar | Not null |
| IFSC_code | Varchar | Not null |
| MICR_code | Varchar | Not null |
| Bank_id | Int | Foreign key |

TABLE NAME: BANK
DESCRIPTION: BANK DETAILS
PRIMARY KEY: BANK_ID

| Field name | Data type | Constraints |
|---|---|---|
| Ac_id | Int | Primary key, Auto increment |
| C_name | Varchar | Not null |
| Ac_no | Varchar | Not null |
| Card_no | Varchar | Not null |
| Cvv | Varchar | Not null |
| Expr_month | Varchar | Not null |
| Expr_year | Varchar | Not null |
| Balance | Bigint | Not null |
| Br_id | Int | Not null |
| Email | Varchar | Not null |

# 8. IMPLEMENTATION AND TESTING

## 8.1 TEST PLAN

The proposed project has undergone the formal process of independent in the same manner as is every other system would undergo. The test plan was designed test quality of the system.

- **UNIT TESTING**
  For each module these test cases are implemented and it is checked whether the module is executed as per the requirement and output the desired result.

- **INTEGRATION TESTING**
  The modules are tested separately for accuracy and modules are integration is rectified

- **VALIDATION TESTING**
  Entering incorrect values does the validation test and it is checked whether the errors are being considered. Incorrect values are to be discarded. The errors are rectified.

- **SYSTEM TESTING**
  It was performed to verify that all system elements have been properly integrated and performed allocated function. Security testing was done to check the security mechanisms build in to the system, which will protect it from improper penetration; performance testing was done to test the run time performance of the system. For user acceptance testing, the system was given to the end-user to use. The error found was rectified.

## 8.2 TESTING

Test cases are developed using various test techniques to achieve more effective testing. By this, software completeness is provided and condition of testing which get the greatest probability of finding errors are chosen. So, testers do not guess which test cases to choose, and test techniques enable them to design testing conditions in a systematic way. Also, if one combines all sorts of existing test techniques, one will obtain better results rather if one uses just one test technique.

### 8.2.1 BLACK BOX TESTING

Black box testing is software testing technique in which the functionality of the software under test (SUT) is tested without looking the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based on the software requirements and specification (Functional testing).

In black box testing we just focused on inputs and outputs of the Parkimine software without bothering about internal knowledge of the software program.

Techniques used:

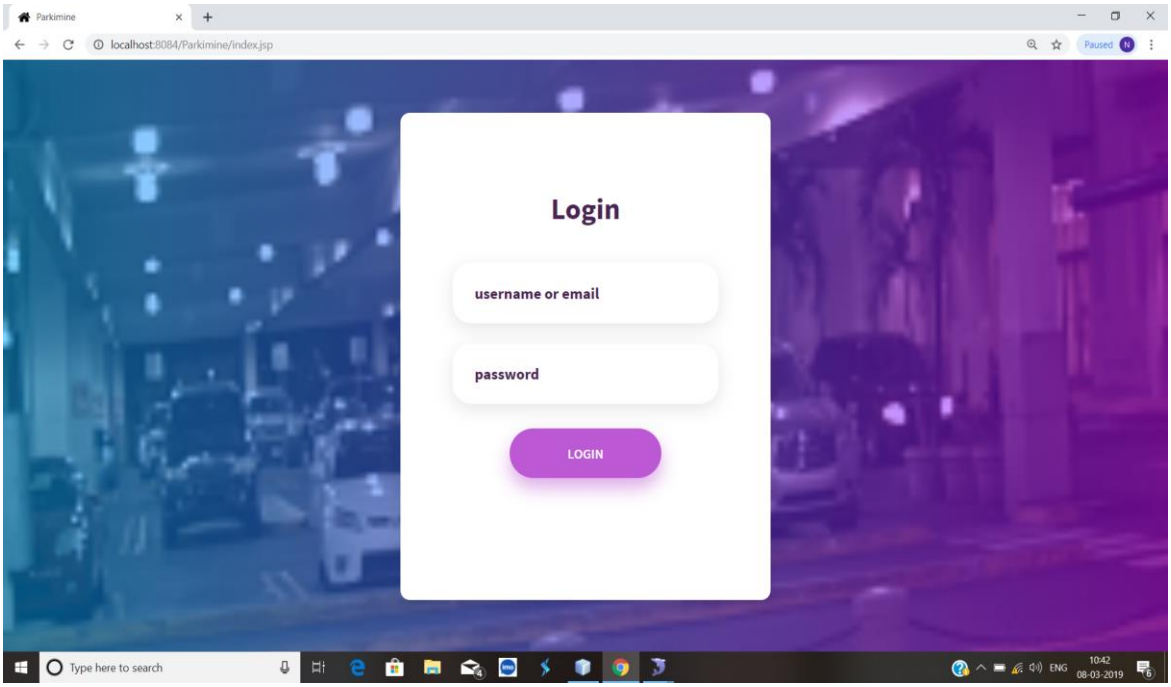
- Boundary value analysis
- Error guessing
- Syntax testing
- Equivalence partitioning

For our Parkimine software we initially examined the requirements and specification of the system. We choose valid inputs to check whether SUT processes them correctly. Also some invalid inputs are chosen to verify that the SUT is able to detect them. And then we determined the expected output for all those inputs. We compared the actual output with the expected output.

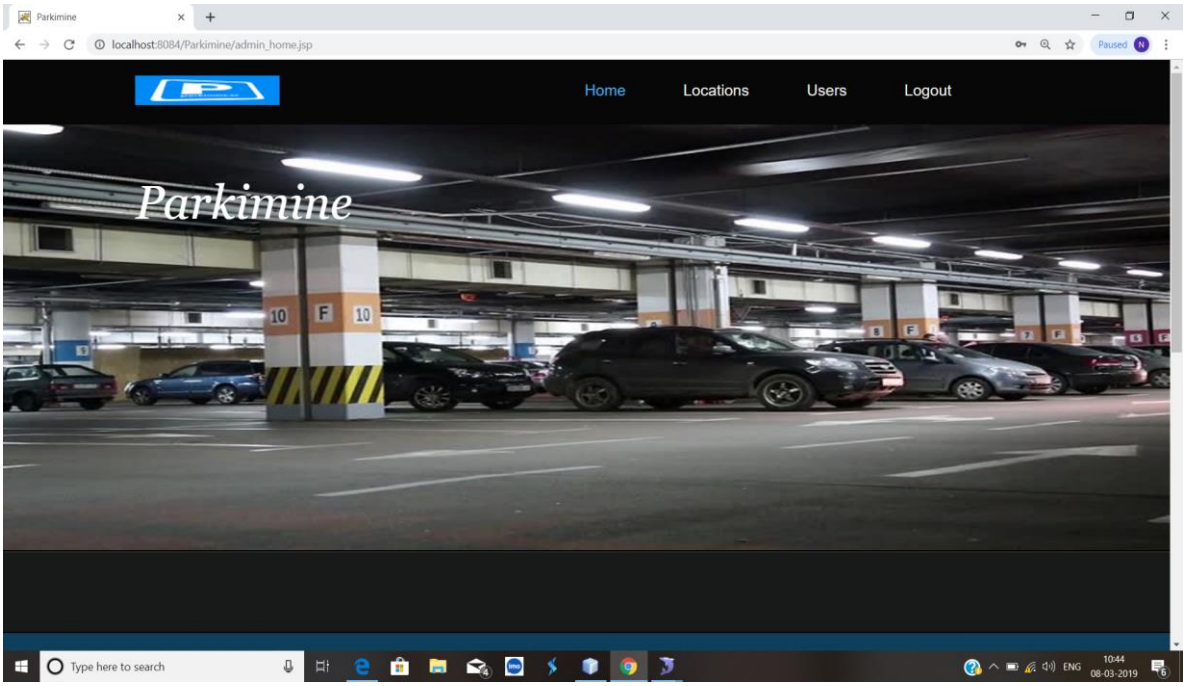
## 8.3 OVERVIEW

### ADMIN MODULE:

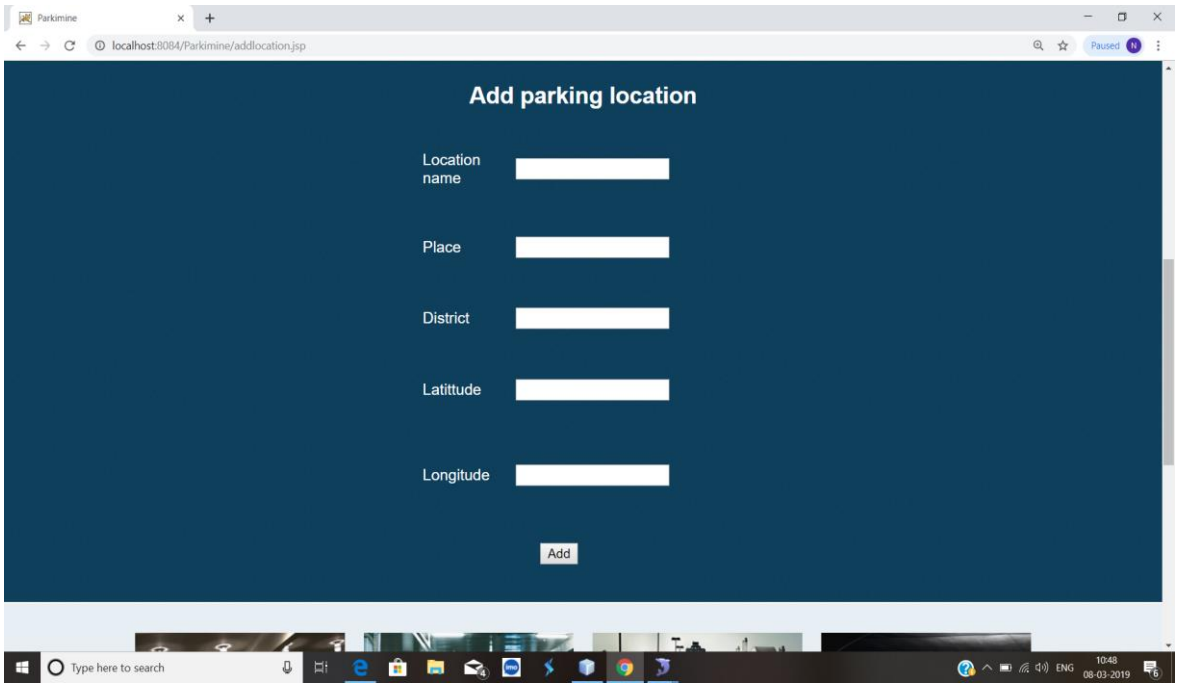
### Login page



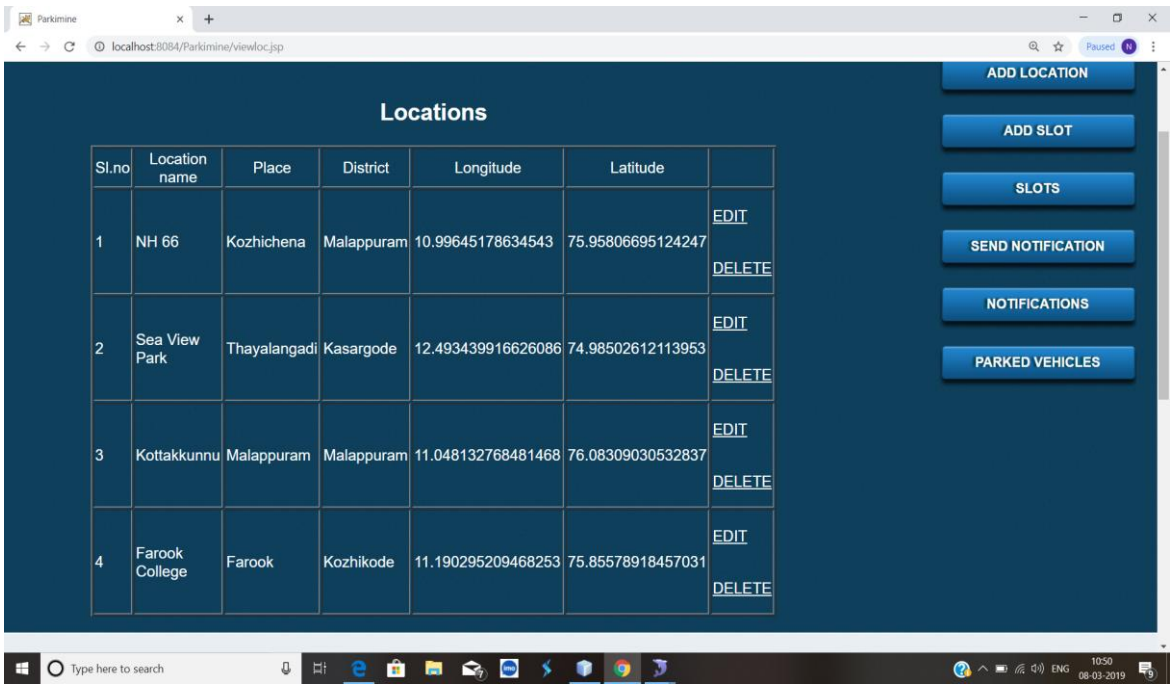

### Home page



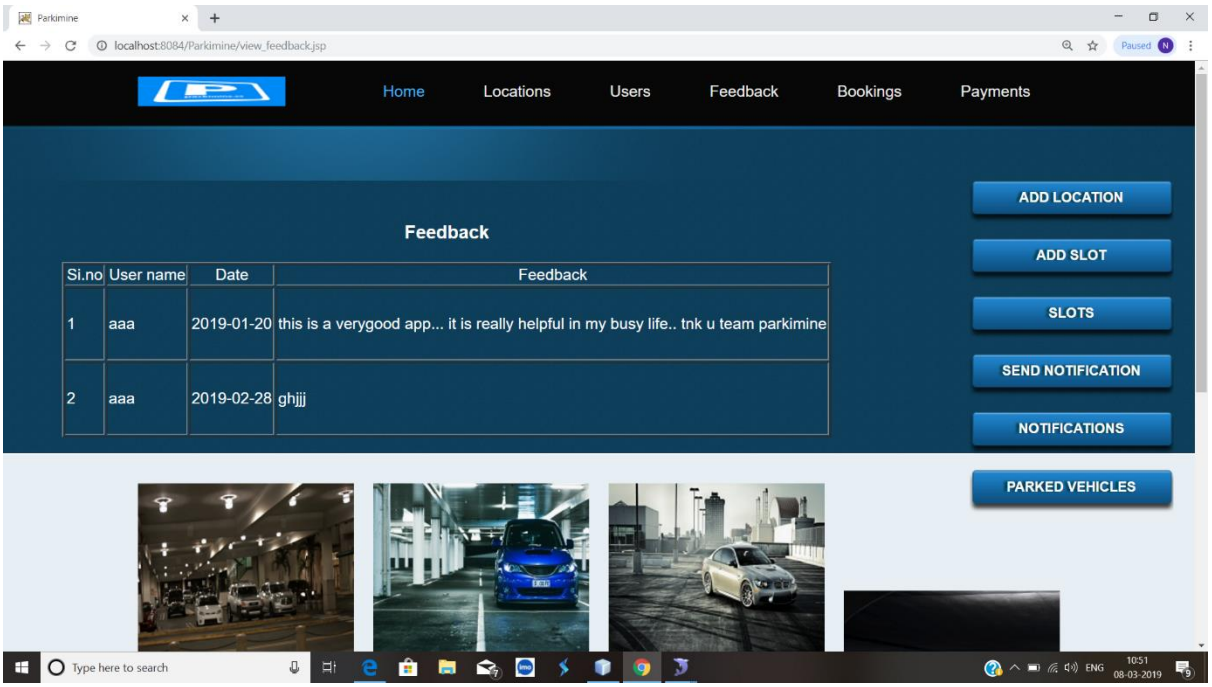
## Add parking location



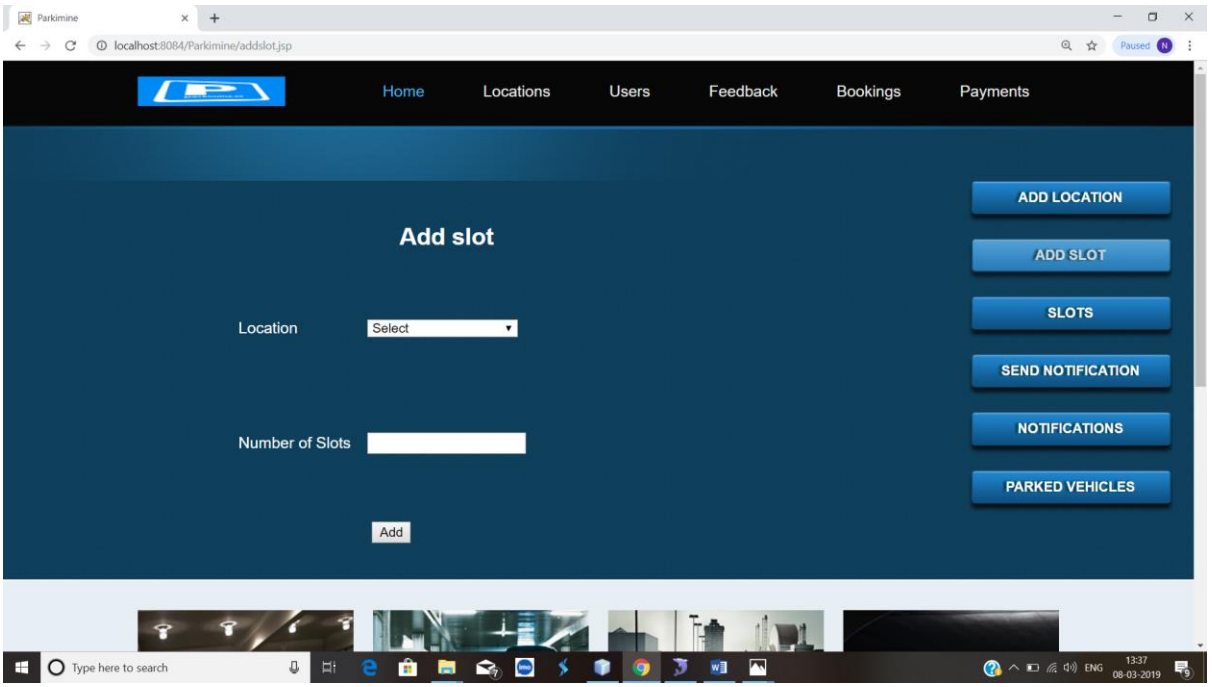

## Locations



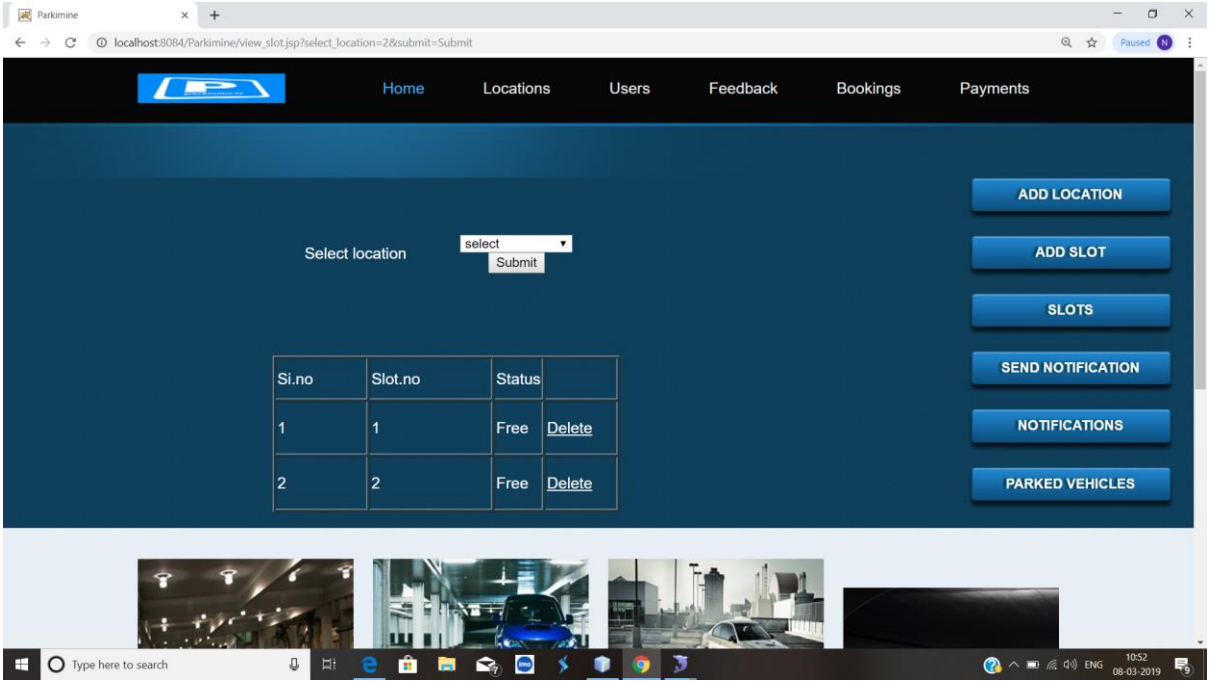
## Feedback



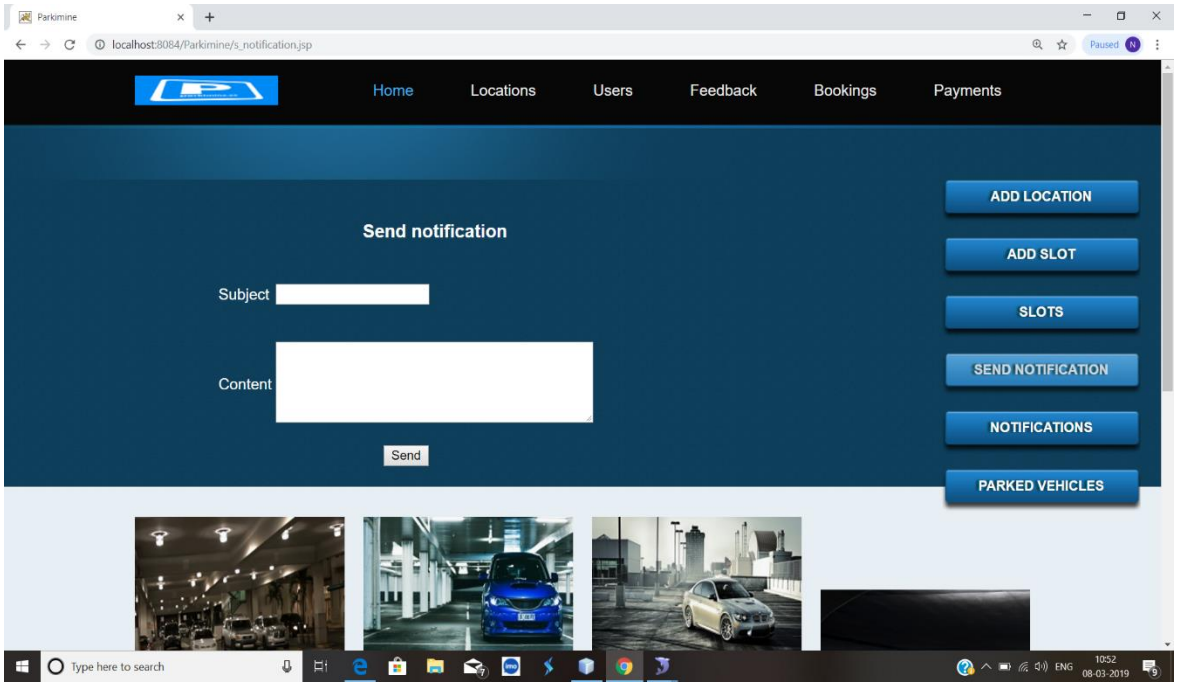

## Add slot



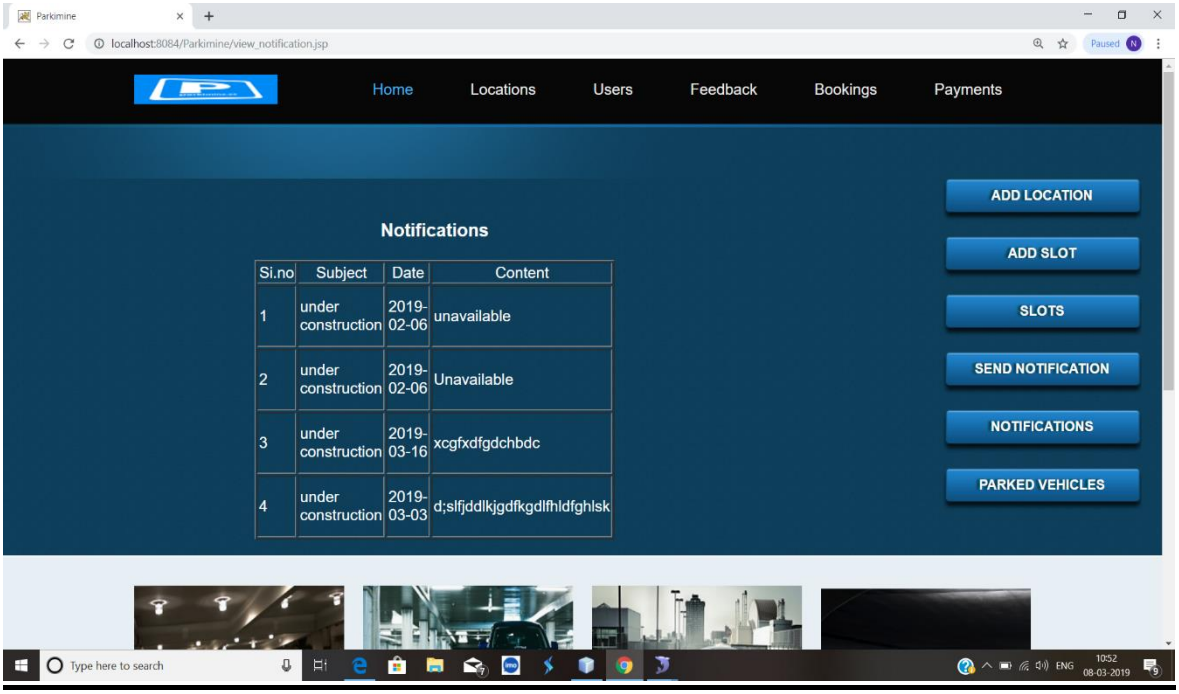
## View slots



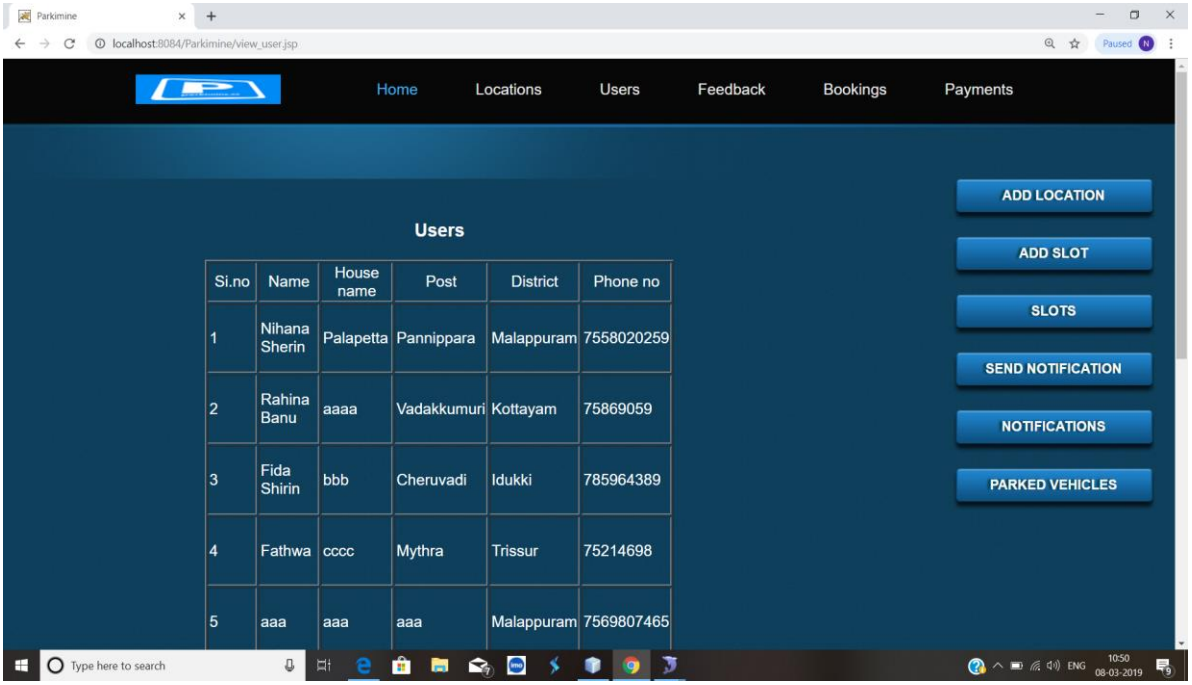

## Send notification



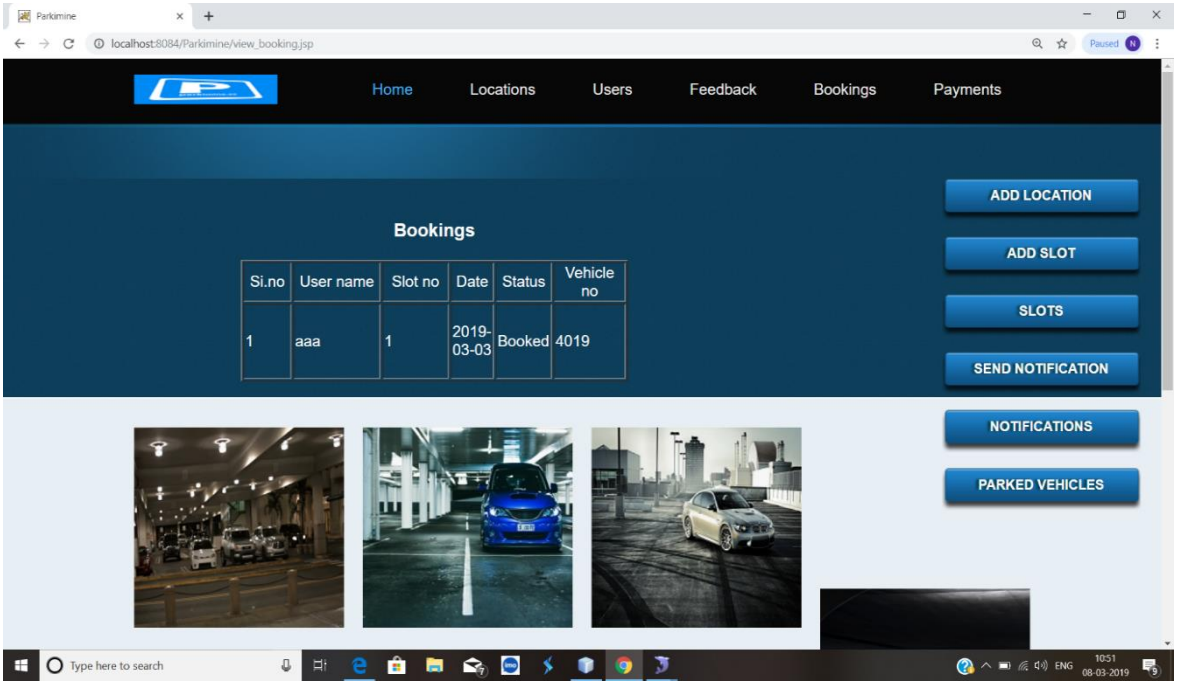
## View notification



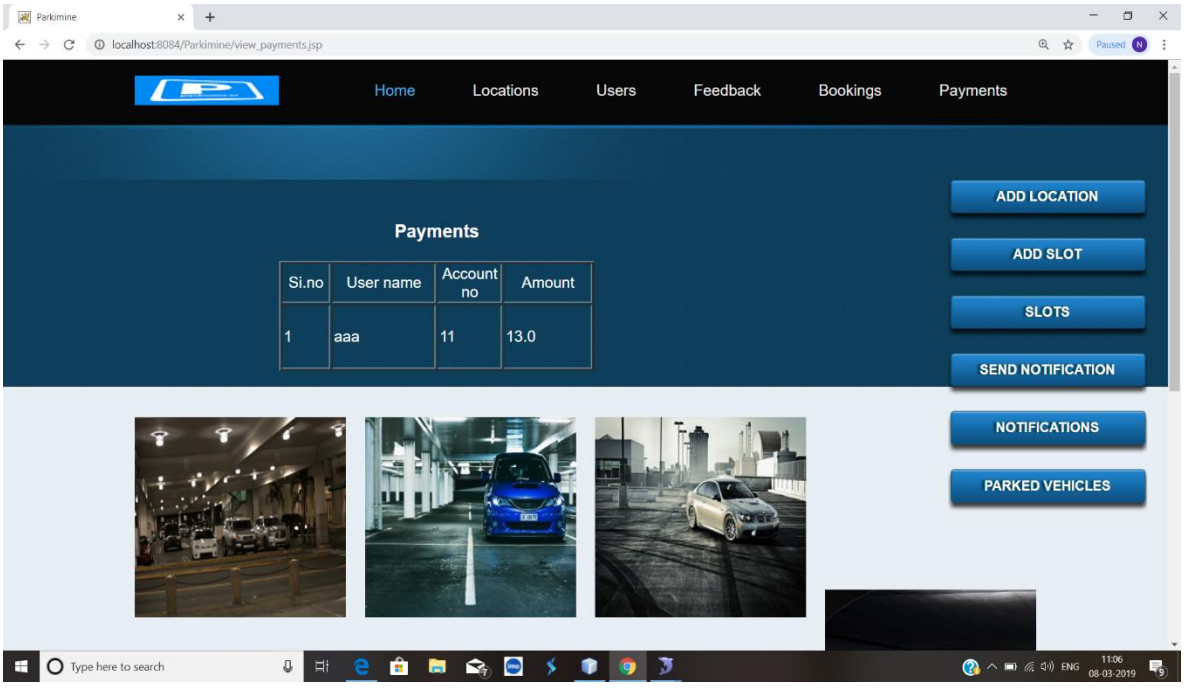

## Users



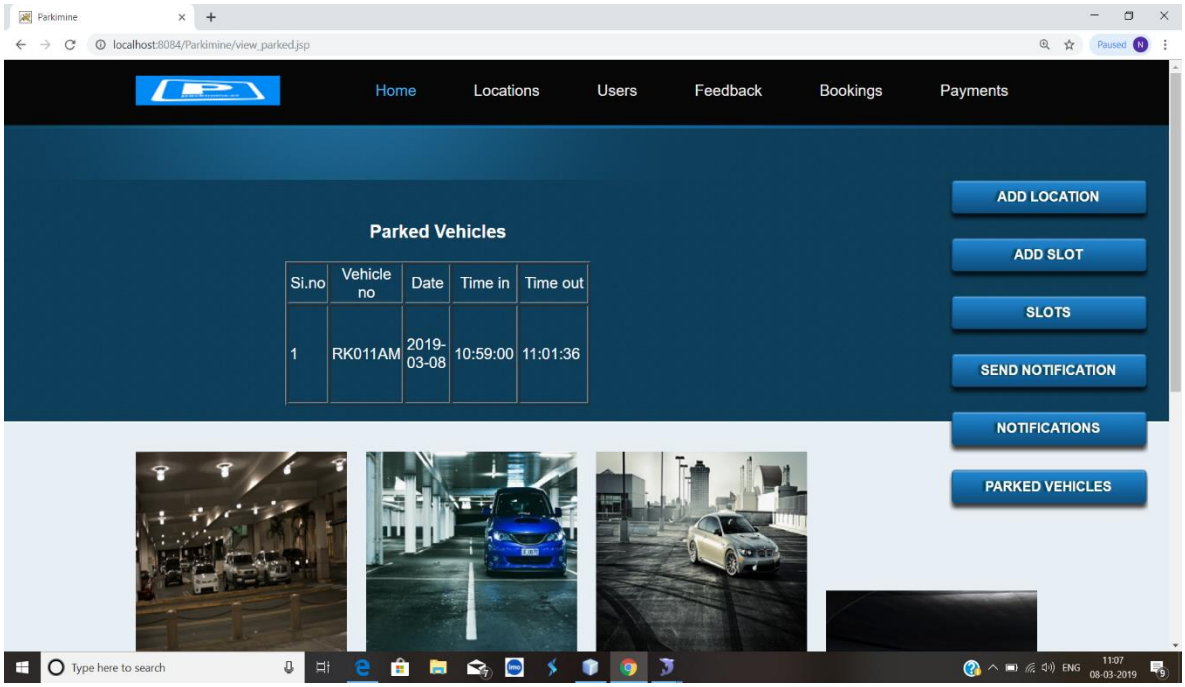
## Bookings




## Payments

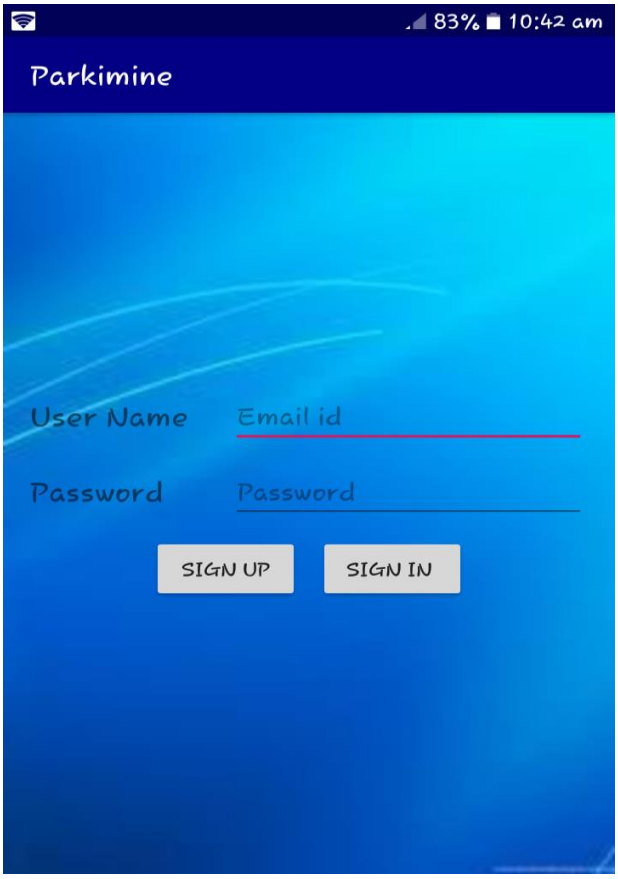


**Parked Vehicles**

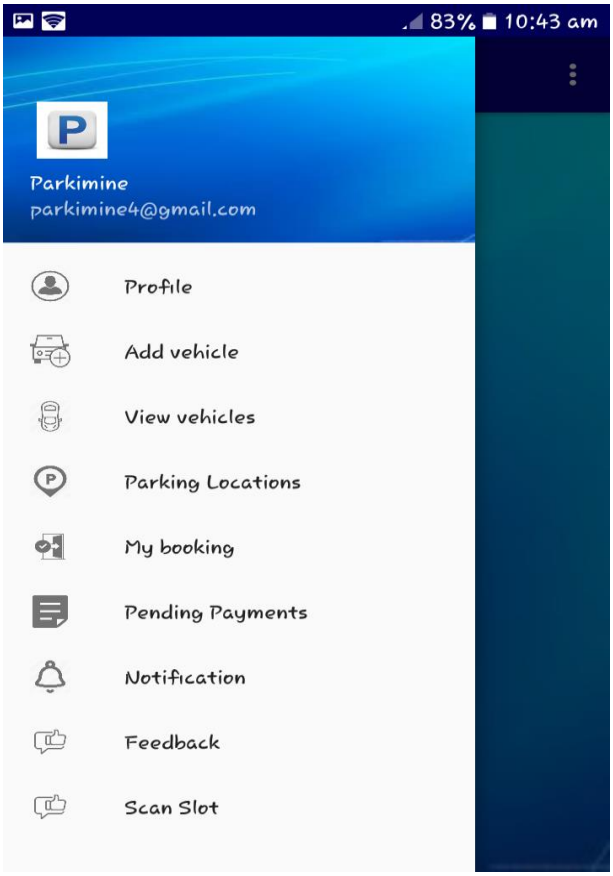



**USER MODULE:**

**Login page**

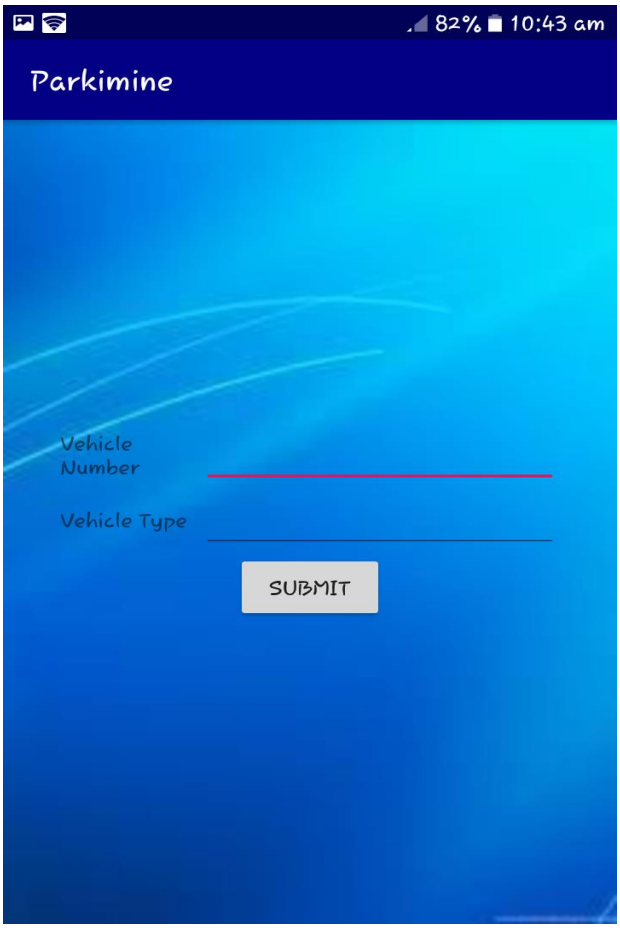


**User home**

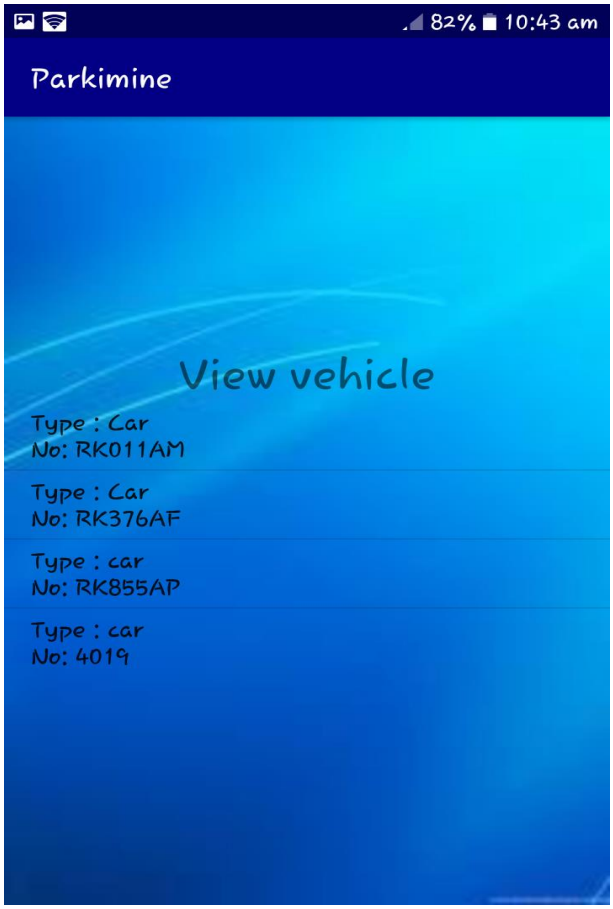



**Profile**

**Add vehicle**



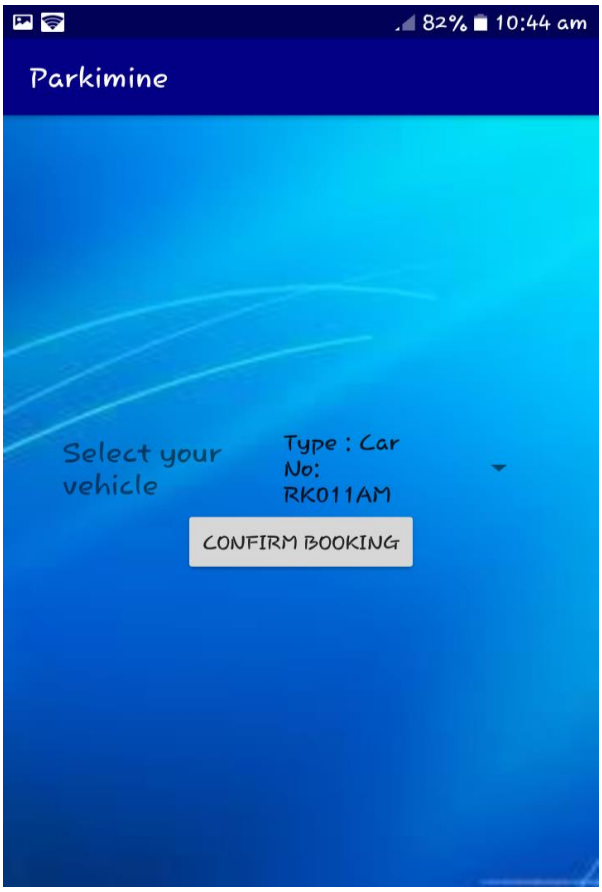

**View vehicle**



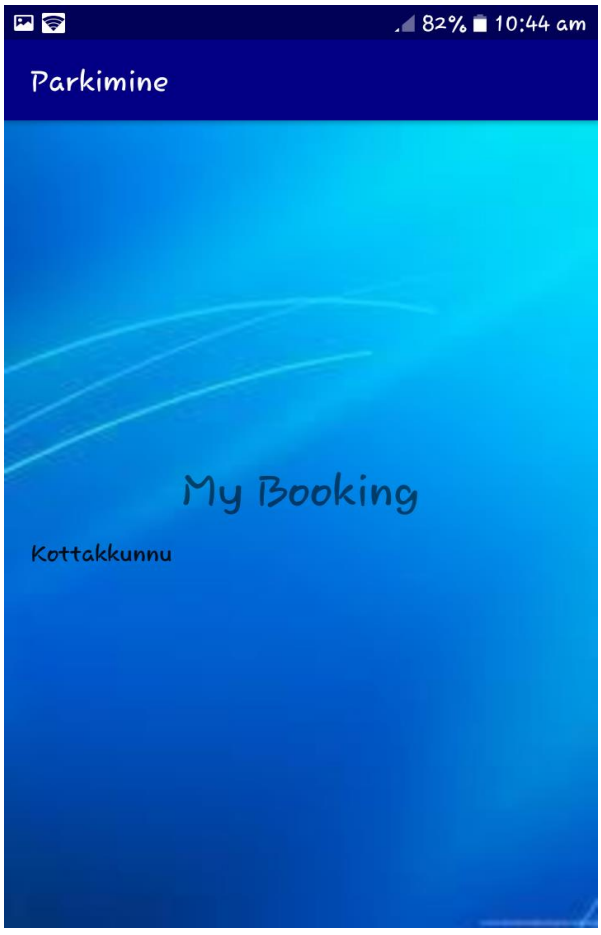
**View location**


**View slots**

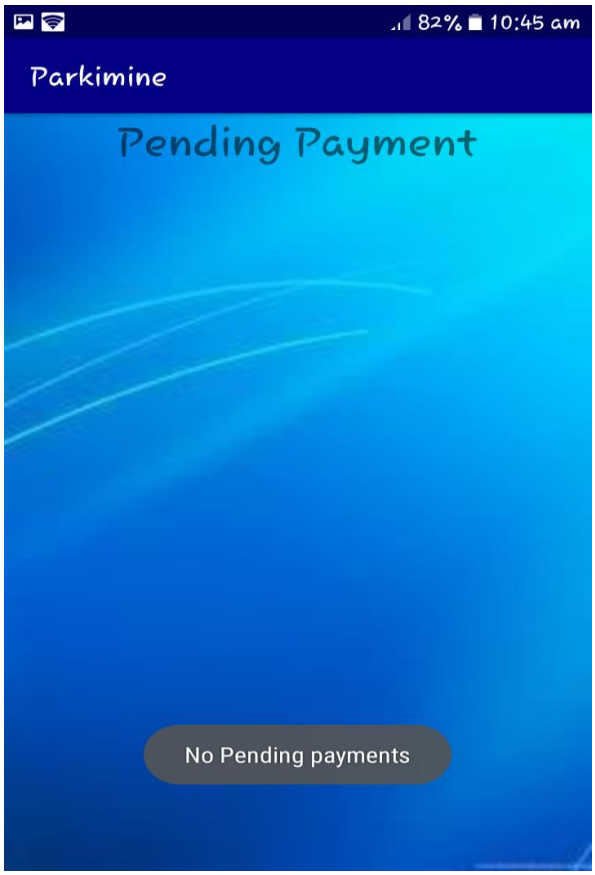
**Booking**



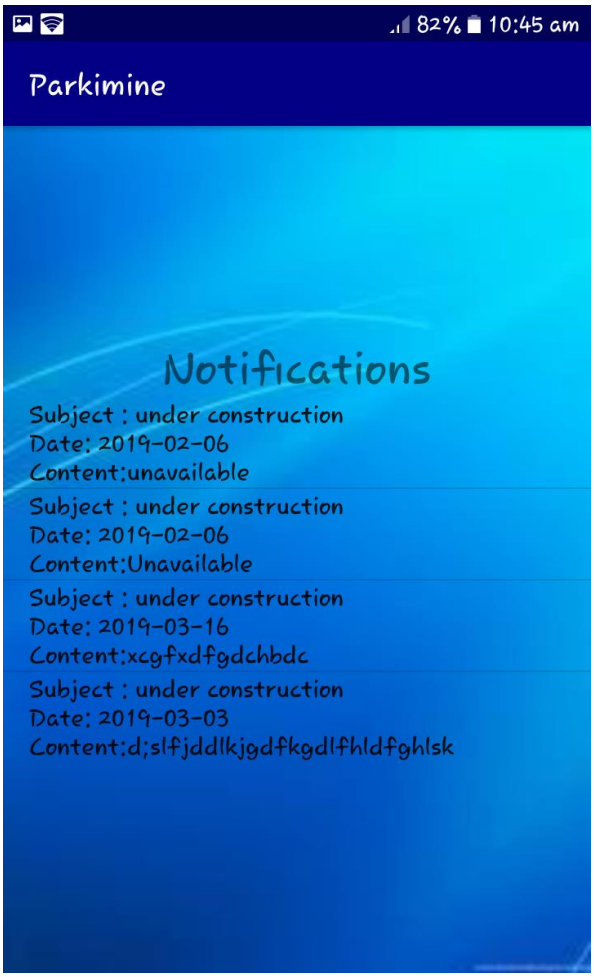

**View booking**



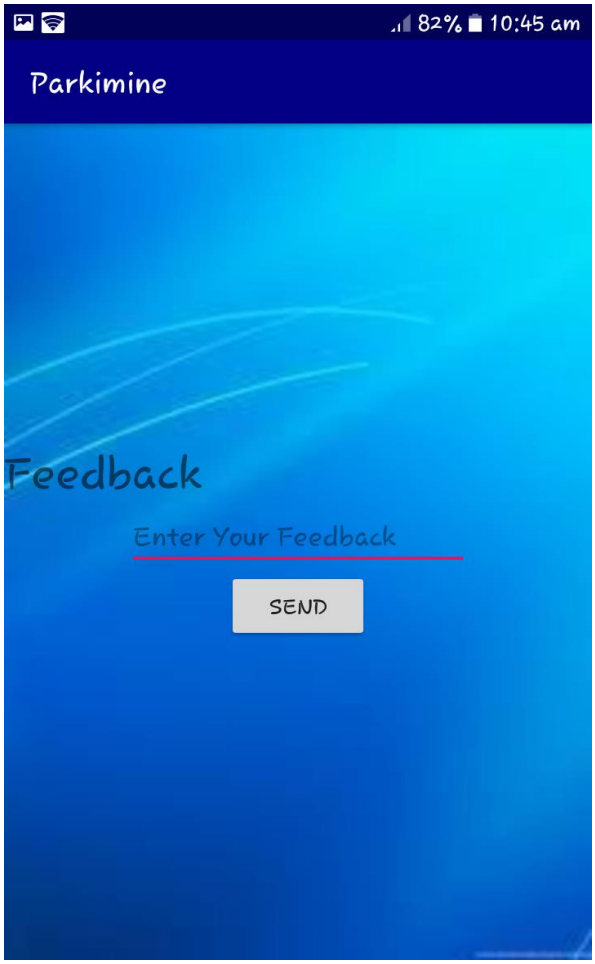
**Pending payments**



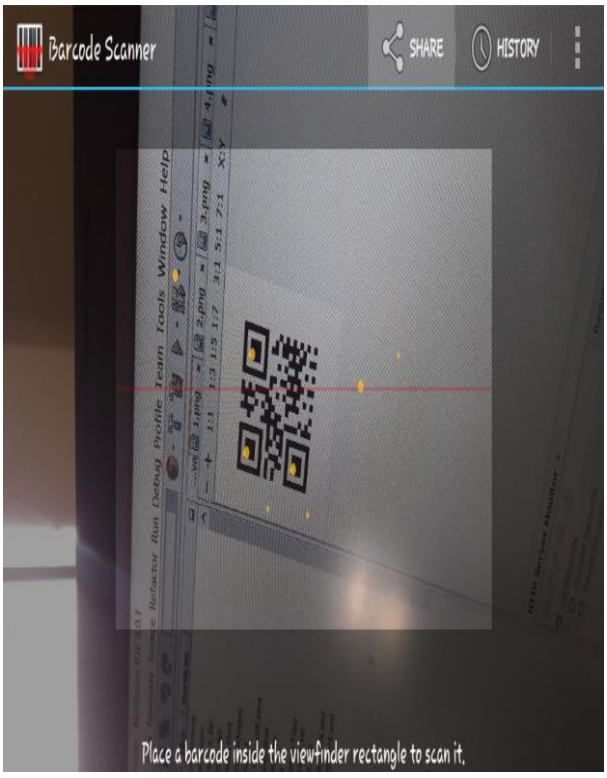

**Notifications**



**Feedback**




**Scan slot**



# 9. MAINTENANCE

Once the system is delivered and deployed, it enters the maintenance phase. All systems need maintenance, but for other systems it is largely due to problems that are introduced due to aging. It needs to be maintained not because of some of its components wear out and need to be replaced, but because there are often some residual error remaining in the system that must be removed as they are discovered. These errors, once discovered, need to be removed, leading to the software getting changed. This is sometimes called "Corrective Maintenance".

## 9.1 FUTURE ENHANCEMENT

Our proposed system does not have a mechanism to avoid the entry of who does not booked the parking slot. We can include a belt or something like that to avoid this problem.

## 10. CONCLUSION

Parking is an issue in many countries due to lack of parking places, we proposed a complete Parking System for the parking slots. This system helps the user as well as the Parking sites have a good management of booking and managing the parking slots. This system contain many options and features when it comes to booking a parking slot and security. This system can be implemented anywhere who wants to rent their parking slots, it can also be used in Commercial or Public places where there is chaos for parking.

## 11. REFERENCE

### Web site

www.stackoverflow.com

www.jpsierens.com

www.javatpoint.com

www.nevonprojects.com

**Books**

Professional Android4 Application Development [Reto Meier]

Learn Java For Android Development [Jeff Friesen]