ONLINE PARKING SPACE MECHANISM WITH PRIVACY PROTECTION

A PROJECT REPORT

Submitted by

FEMINA S [211418104059]

KIRTHIGA N [211418104125]

KIRUTHIKA M [211418104126]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

(An Autonomous Institution, Affliated to Anna University,Chennai)

MAY 2022

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"ONLINE PARKING SPACE MECHANISM WITH PRIVACY PROTECTION "** is the bonafide work of "**FEMINA S (211418104059), KIRTHIGA N (211418104125), KIRUTHIKA M (211418104126)** "who carried out the project work under my supervision.

SIGNATURE
Dr.S.MURUGAVALLI,M.E.,Ph.D..
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE
Mrs.S.T. SANTHANALAKSHMI,
SUPERVISOR
ASSISTANT PROFESSOR (GRADE-I)

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above mentioned students were examined in End Semester project viva-voice held on _____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **Femina S (211418104059), Kirthiga N (211418104125), Kiruthika M (211418104126)** hereby declare that this project report titled **"ONLINE PARKING SPACE MECHANISM WITH PRIVACY PROTECTION",** under the guidance of **Mrs.S.T.SanthanaLakshkmi**, **ASSISTANT PROFESSOR (GRADE-I)** is the original work done by us and we have not plagiarized or submitted to any other degree in any other university by us.

1.FEMINA S

2.KIRTHIGA N

3.KIRUTHIKA M

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR, M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **Mrs.S.T.SanthanaLakshmi, ASSISTANT PROFESSOR (GRADE-I)** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

FEMINA S

KIRTHIGA N

KIRUTHIKA M

ABSTRACT

Sharing private parking spots has indicated an incredible potential for tending to metropolitan gridlock and unexpected stopping issues in urban areas. Planning to address the internet parking spots sharing issue while guaranteeing the protection of client stopping objective areas, we propose a novel objective privacy-preserving internet. The internet parking spot sharing issue is formalized as a social government assistance expansion issue in a two-sided market, where parking spot suppliers and clients are viewed as dealers and purchasers. At that point, these are based on standards that are intended to decide installments and repayment. We examine the upper bound of the productivity loss of our plan. Broad assessment results exhibit that our plan cannot just accomplish great execution with respect to social government assistance, Supplier fulfillment proportion, protection conservation.

LIST OF TABLES

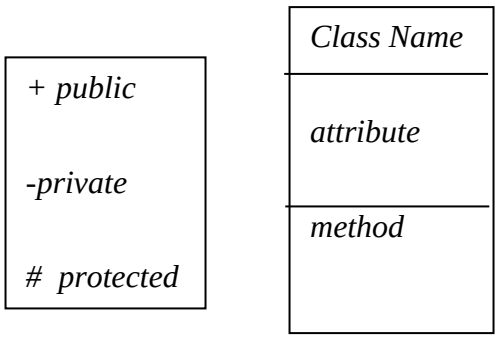
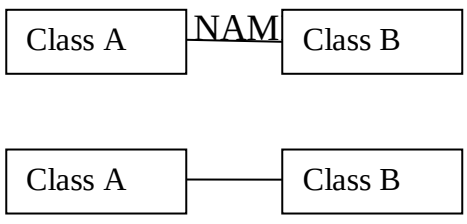
Table No.	Table Description	Page Number
4.4	Data Dictionary	
4.4	User	22
4.4	Owner	22
4.4	Payment	23
4.4	Admin	23

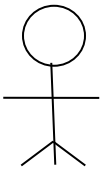
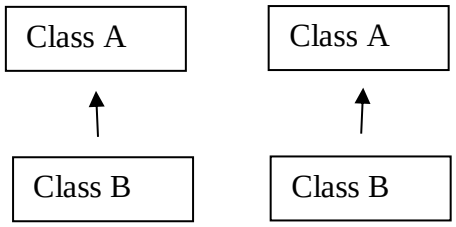
LIST OF FIGURES

Figure	Figure Description	Page no
4.1	ER DIAGRAM	13
4.2	DATA FLOW DIAGRAM	14
4.3	UML DIAGRAM	18
4.3	USECASE DIAGRAM	18
4.3	SEQUENCE DIAGRAM	19
4.3	ACTIVITY DIAGRAM	20
4.3	COLLABORATION DIAGRAM	20
5.1	SYSTEM ARCHITECTURE	24
A.1	SCREENSHOTS	52
A.1	INDEX PAGE	52
A.2	USER REGISTER	53
A.3	USER LOGIN	54
A.4	OWNER REGISTER	54
A.5	OWNER LOGIN	55
A.6	HOME PAGE	55
A.7	VIEW PARKING	56
A.8	BOOKING SLOT	56
A.9	PAYMENT	57

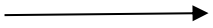
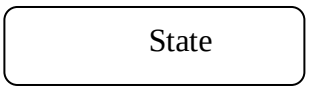
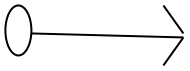
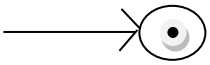
A.10	OWNER PROFILE	57
A.11	ADD SLOT	58
A.12	ADMIN INDEX	58
A.13	OWNER DETAILS	59
A.14	USER DETAILS	58
A.15	BOOKING DETAILS	60
A.16	ABOUT PAGE	60

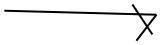
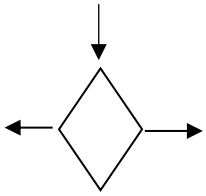
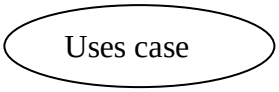
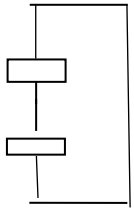
LIST OF SYMBOLS

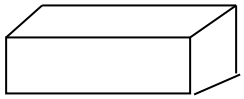
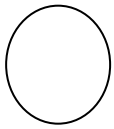
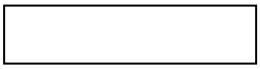
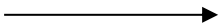
S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles

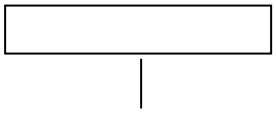
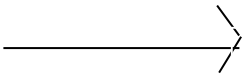
			represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment

5.	**Relation (uses)**	uses	Used for additional process

			communication.
6.	Relation (extends)	extends 	Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processs.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object

11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.
14.	Component		Represents physical modules which is a

			collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or acion.
17.	External entity		Represents external entities such as keyboard,sensors,etc.
18.	Transition		Represents communication that

			occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message	Message 	Represents the message exchanged.

ABBREVIATIONS

S.NO	ABBREVATION	EXPANSION
1.	DB	DATABASE
2.	PHP	HYPERTEXT PREPROCESSOR
3.	HTML	HYPERTEXT MARKUP LANGUAGE
4.	CSS	CASCADING STYLE SHEET
5.	JS	JAVASCRIPT
6.	DFD	DATA FLOW DIAGRAM
7.	UML	UNIFIED MODELLING LANGUAGE
8.	OS	OPERATING SYSTEM

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	V
	LIST OF TABLES	Vi
	LIST OF FIGURES	Vii
	LIST OF SYMBOLS, ABBREVIATIONS	Viii
1.	**INTRODUCTION**	
	1.1 Overview	1
	1.2 Problem Definition	2
2.	**LITERATURE SURVEY**	3
3.	**SYSTEM ANALYSIS**	
	3.1 Existing System	9
	3.2 Proposed system	9
	3.3 Feasibility Study	10
	3.4 Hardware Environment	12
	3.5 Software Environment	12
4.	**SYSTEM DESIGN**	
	4.1. ER diagram	13
	4.2. Data Flow Diagram	14

CHAPTER NO.	TITLE	PAGE NO.
	4.4 Data Dictionary	18
5.	**SYSTEM ARCHITECTURE**	
	5.1 Modules and design specification	25
	5.2 Algorithms	26
6.	**SYSTEM IMPLEMENTATION**	
	6.1 Client-side coding	28
	6.2 Server-side coding	35
7.	**PERFORMANCE ANALYSIS**	
	7.1 Testing and test results	48
8.	**CONCLUSION**	
	8.1 Conclusion and Future Enhancements	51
	APPENDICES	
	A.1 Sample Screens	51
	A.2 References	61

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In recent years, motor vehicle and non-motor vehicles in small cities is the parking lot is far behind the growth rate of the motor vehicle, resulting in small urban areas especially the demand and the supply of parking facilities,the downtown area of the planning and construction of parking policy and management issues have become increasingly prominent, to solve the parking problem has very urgent. Parking space sharing model is more and more attention at home and abroad, Mou Zhenhua put forward a strategy aimed at reducing the shared parking in the city land use of the city center area, analyzes the feasibility of the shared parking policy. Generation method is proposed to determine the percolation of Sichuan, sharing parking behavior coefficient for domestic actual shared parking demand forecasting model and realize the sharing of parking mode specific implementation measures and policies. Institute of Transportation Engineers (ITE) found more than half of the American local governments have shared parking theory into the local parking management mode, although the practical approach may be used directly or selective use. Related researchers discussed the parking sharing the feasibility of the theory; put forward the application strategy, in the concrete example of application. But the shared parking method is mainly used in the new project berth planning stage, no scholars after the implementation of sharing measures on urban traffic congestion and parking is difficult to solve the problem of the utility is analyzed by a numerical analysis, the utility can be directly obtained by the utility value, easy implementation of shared parking analysis function, providing theoretical support for parking planning and design.

1.2 PROBLEM DEFINITION

Nowadays, the availability of parking spaces is far behind the quick rising number of cars. Rather than building more lots, a better way is to share private-owned parking spaces. However, this faces the challenge that users are not willing to expose their privacy to the public. The increasing rate of private car usage in the urban areas as a result of fast-growing economy, derelict policies and subsidies are the main causes making car parking one of the main concerns for transport and traffic management all over the world. The coordination between parking policies and traffic management revealed how parking is becoming a barrier to the through-traffic operation. Also, it is responsible for the inefficient use of available resources, even the decisions are made on an ad-hoc basis while making policy. Hence, it is necessary to understand the parking choice behaviour and actual demand of parking space. By parking the vehicle in public place the vehicle can be claimed by towing person but in this case there is no towing problems and no need to give fine for anything we can park our vehicle with securely. To solve this problem, we propose a new architecture for parking space sharing, into the design of a secure protocol for parking space searching and booking by using maps.

CHAPTER 2

LITRATURE SURVEY

1. GuoQingsheng, WengXiaoxiong, SongMinglei, " The Utility of Shared Parking in Small Towns of Mixed Use Lands. " - 2015

At present, the number of small town's rapid growth of motor vehicles, the dynamic and static traffic put forward more requirements. This paper first analyzes the characteristics of small town traffic, put forward the problems caused by the traffic congestion and parking is difficult because of small cities and towns, proposed the application of car sharing model is the effective measure to solve the problems. At the same time, this paper defines the concept of the solve the traffic congestion utility index & The solve the parking difficulty utility index and has carried on the concrete application of.The application sharing to solve the traffic congestion and parking is difficult to enhance the effectiveness of the obvious conclusion.

2. Biao Xu, Guojun Da, Zhen Jiang, Jie Wu Peng Liu," MDP: Minimum delay hot-spot parking" - 2017

In Minimum Delay Hotspot Parking, a new PGI, denoted by MDP, has been proposed to mitigate the impact of the parking hassle of delay in average and worst cases. We provide a spatio temporal assignment, to take advantage of the vacancy that grows along the time scale when the demands goes beyond the supply. The unique directive is to solve the forementioned starvation problem in other PGI schemes or similar vehicle dispatching. The contribution is to reduce delay without increasing the facility supply. Both analytical and experimental results demonstrate that our approach can achieve a bounded service, in terms of vehicle cruising time and the overhead cost of information collection and computation. Moreover, we study the extension by trading

in the local waiting when the driver knows how soon the vacancy becomes available. The corresponding assignment is denoted by MDP+. Later full service can be provided for scheduling every parking request. In future work, we will consider the capacity deca-dence when both assisted and non-assisted drivers co-exist in the parking field. We will study the tradeoff between the global optimization and the greedy approximation algorithm, so that even more practical solutions can be achieved. We also expect to apply this spatio temporal assignment scheme to other resource shortage problems (e.g., [4, 22]), while a global optimization is desired.

3. MAHMOUD M. BADR , WESAM AL AMIRI , MOSTAFA M. FOUDA, MOHAMED MAHMOUD, ABDULAH J. ALJOHANI and WALEED ALASMARY, " Smart Parking System with Privacy Preservation and Reputation Management Using Blockchain ” - 2017

Block Chain Drivers and the dependence on them will mitigate the traffic congestion and the air pollution negatively affecting many communities. With the emerging intelligent transportation system, modern vehicles are equipped with internet accessing facilities and self-parking functions, and also self-driven cars. The smart parking system can be perfectly applied to all types of vehicles. The internet access can facilitate the communication between vehicle and parking system. Moreover, once a car reaches its reserved parking slot or respective destination, the self-parking functions can be activated to park the car.In the future, we expect to hybrid smart parking system that allows both public and private parking owners to participate in the system. Private parking owners such as home inhabitants can share their parking slots suppose they are not using it. This will increase the potential of number of slots available for parking. However, in such cases the privacy of the private parking owners should also be consider into account, and the parking system needs to be updated.

4. Wenhui Zhang, FanGao, ShuruiSun, QiuyingYu, JinjunTang, BohangLiu," A Distribution Model for Shared Parking in Residential Zones that considers the Utilization Rate and the Walking Distance " - 2019

Efficient parking tends to be challenging in most large cities in China. Drivers often spend substantial amount so time looking for parking lots while driving at low speeds, thereby resulting in interference with road traffic. This paper focuses on efficiently allocating parking spaces to the demanders. A double-objective model is proposed that considers both the utilizing rate and the walking distance. First, managers want utilize parking resources fully. However, demanders typically choose parking spaces according to convenience. The second objective is the acceptable walking distance from the parking space to the destination. We collected parking demand and supply data in a Central Business District (CBD) of Harbin in China and evaluated the feasibility of the model.The results demonstrate that the proposed model increases the occupying rates of parking lots in residential zones while decreasing the walking distance. The shared use of parking spaces maximizes the utility and alleviates the shortage of parking spaces in downtown.

5. Xin Huang, Xueqin Long, Jianjun Wang, LanHe, ” Research on parking sharing strategies considering user overtime parking. “ - 2019

A parking sharing strategy is proposed to solve the problems of parking difficulty caused by the imbalance between parking spaces and parking demand. The vacant parking spaces of residential area can be efficiently utilized to meet the parking demands of those who are working at nearby or come for other activities based on the parking sharing strategy. The paper analyzes the distribution of vehicle arrival numbers and parking durations, and then establishes a shared parking allocation model aiming to maximize the parking benefit considering the overtime parking behavior of the parking users. Simulation methods are used to the analyze the relationship among

the parking benefit, proportion of reserved parking, numbers of parking demand, acceptance rate of parking demand and utilization of shared parking spaces. Then, based on that of maximum parking benefit, we can determine the optimal proportion of reserved parking, number of shared parking spaces that should be purchased from the residents. Taking the utilization of shared parking spaces as an indicator, the validity of the static allocation is proved to be effective.

6. Oanh Tran Thi Kim, Nguyen H. Tran, Chuan Pham, Tuan LeAnh," Parking Assignment: Minimizing Parking Expenses and Balancing Parking Demand among Multiple Parking Lots. " - 2020

Recently, a rapid growth in the number of vehicles on the road has led to an unexpected surge of parking demand. Consequently, finding a parking space has become increasingly difficult and expensive. One of the viable approaches is to utilize both public and private parking lots (PLs) to effectively share the parking spaces. However, when the parking demands are not balanced among PLs, a local congestion problem occurs where some PLs are overloaded, and others are underutilized. Therefore, in this article, we formulate the parking assignment problem with two objectives:

1. Minimizing parking expenses
2. Balancing parking demand among multiple PLs.

First, we derive a matching solution for minimizing parking expenses. Then, we extend our study by considering both parking expenses and balancing parking demand, formulating this as a mixed integer linear programming problem. We solve that problem by using an alternating direction method of multipliers (ADMM) based algorithm that can enable a distributed implementation. Finally, the simulation results show the ADMM-based algorithm produces performance gains up to 27.5%

7. **Xin Huang, Xueqin Long, Jianjun Wang, LanHe, "Research on parking sharing strategies considering user overtime parking. " - 2020**

A parking sharing strategy is proposed to solve the problems of parking difficulty caused by the imbalance between parking spaces and parking demand. The vacant parking spaces of residential area can be efficiently utilized to meet the parking demands of those who are working at nearby or come for other activities based on the parking sharing strategy. The paper analyzes the distribution of vehicle arrival numbers and parking durations, and then establishes a shared parking allocation model aiming to maximize the parking benefit considering the overtime parking behavior of the parking users. Simulation methods are used to the analyze the relationship among the parking benefit, proportion of reserved parking, numbers of parking demand, acceptance rate of parking demand and utilization of shared parking spaces. Then, based on that of maximum parking benefit, we can determine the optimal proportion of reserved parking, number of shared parking spaces that should be purchased from the residents.Taking the utilization of shared parking spaces as an indicator, the validity of the static allocation is proved to be effective.

8. Tingting Fu, Peng Liu, Kun Liu, and Peng Li," Privacy-Preserving Vehicle Assignment in the Parking Space Sharing System" - 2020

In privacy-preserving vehicle assignment problem in the parking space sharing system is studied and analyzed. A homomorphic encryption-based privacy protection matching scheme (PPMS) is introduced and designed. In order to enhance the technology and security, a block algorithm based on the longitude and latitude (BABLL) is proposed. Through the security analysis, the scheme is proved to be able to protect the privacy of sensitive information such as location, time, identity of both requestors, and space providers. The scheme is also robust against attacks, e.g., Replay. We implement the prototype system and conduct comparative experiments. The results show that the proposed scheme can ensure very good success rate of

matching with high time efficiency. In addition to that, the system reduces multiple rounds of practical attacks, while maintains normal operations.

9. Xumin Huang, Peichun Li, Rong Yu, Yuan Wu, Kan Xie, and Shengli Xie" FedParking: A Federated Learning based Parking Space Estimation with Parked Vehicle assisted Edge Computing. " - 2021

Here, they introduced FedParking to study federated learning based parking space estimation with PVEC management. PLOs were instructed to train shared LSTM model for parking space estimation without exchanging the raw data. A parking space constraint is presented to each PLO, which acts as an incentive designer to determine how to stimulate the vehicles to enter the parking spaces and share their idle computing resources in PVEC. The interactions between PLOs and vehicles as a multi-leader multi-follower Stackelberg game and provided theoretical Stackelberg equilibrium analysis under complete information. Dynamic arrivals of the vehicles and time-varying parking capacity constraints, a DRL approach was specifically proposed to reach the Stackelberg equilibrium in privacy-friendly way. Finally, results are demonstrated that the scheme is effective for high-accuracy parking space estimation, and it can seek a solution under the complicated situations.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

We presently model the internet parking spot sharing issue as a two-sided market, where the PSSs go about as dealers and the PSCs go about as purchasers. The market will be set off when there are purchasers and dealers. The framework works in a period opened style. The time allotment is set by the representative and the schedule opening is 60 minutes. Likewise, the PSSs and the PSCs can show up and withdraw from the market progressively, without advance information on the bartering.

DISADVANTAGES:

- It is not represent parking spaces and slots.
- It is does not have authorized parking slots.
- It requires large database.
- It does not have reserved for particular timing.

3.2 PROPOSED SYSTEM:

The proposed system of project is that provides easy way of reserving a parking space online using web portal. It overcomes the problem of finding a parking space in areas that unnecessarily consumes time. Hence, this project offers a web application based reservation system where users can view various parking spaces and select nearby or specific area of their choice to view whether space is available or not. If the booking space is available, then user can book it for specific time slot.

The booked space will be marked and will not be available for anyone else for the specified time.

ADVANTAGES:

- It is representing clearly with maps.
- It has authorized parking slots with authorized address.
- It is so easy to use with simple UI.
- It has reserved for particular timing.
- Users can get details about parking areas for particular locations.
- The system provides a view of the parking spaces.
- It excludes the need of human efforts for managing parking spaces.
- It is representing clearly with clear locations.
- It has authorized parking slots with authorized address.
- It is so easy to use with simple User Interface.
- It has reserved for particular timing.

3.3 FEASIBILITY STUDY:

- ECONOMICAL
- TECHNICAL
- SCHEDULE
- OPERATIONAL

3.3 ECONOMICAL FEASIBILITY:

Economic feasibility attempts to weigh the cost of developing and implementing a new system, against the benefits that would accurate from having the new system in place.

This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. These could include increased customer satisfaction, improved accuracy of operation, better documentation and recordkeeping, faster retrieval of information.

3.3 TECHNICAL FEASIBILITY:

Technical feasibility evaluates the hardware requirements, software technology, available personnel etc., As per the requirements it provides sufficient memory to hold and process the data as it uses Google cloud Server, as the back end. Technically it is feasible as it is platform independent It is easily portable i.e. it is written in java with no platform-native code. The results obtained from technical analysis from the basic for html which is used to display content in the browser, CSS to make content look user friendly, and JavaScript for making the web page interactive. At the server side, the logic is implemented using android studio and web framework for dynamic web page generation and to display the predicted result in the browser as well as to handle page requests. A server, client, and internet connection are required to function properly.

3.3 SCHEDULE FEASIBILITY:

Schedule Feasibility means that the project can be completed on time. The project does not have a deadline but according to the proposed system the development process is on schedule. Therefore it is feasible.

3.3 OPERATIONAL FEASIBILTY:

Proposed project is beneficial only if it can be turned into information systems that will meet the organization operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed.

3.4 HARDWARE ENVIRONMENT:

PROCESSOR	: Intel Core i3.
RAM	: 4 GB
MONITOR	: 15” COLOR
HARD DISK	: 25 GB

3.5 SOFTWARE ENVIRONMENT:

FRONT END	: HTML, CSS, JAVASCRIPT
BACK END	: MYSQL, PHP
OPERATING SYSTEM	: Windows 07
SOFTWARE	: NOTEPAD, XAMPP

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM:

ER diagram illustrates the logical structure of a database by defining entities, their attributes, and their relationships.

ENTITY:

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent. The entities used here are user, owner, payment and admin

ATTRIBUTES:

The attributes are represented in the form of ellipse. Every attribute is connected to its entity.

RELATIONSHIP:

Relationships are represented in the form of diamond. A Relationship has is used here are authentication, search availability, post availability, payments. A Relationship which has two entities are called as binary relationship. These relationships can be represented in four ways as follows,

- One to one
- One to many
- Many to one
- Many to many

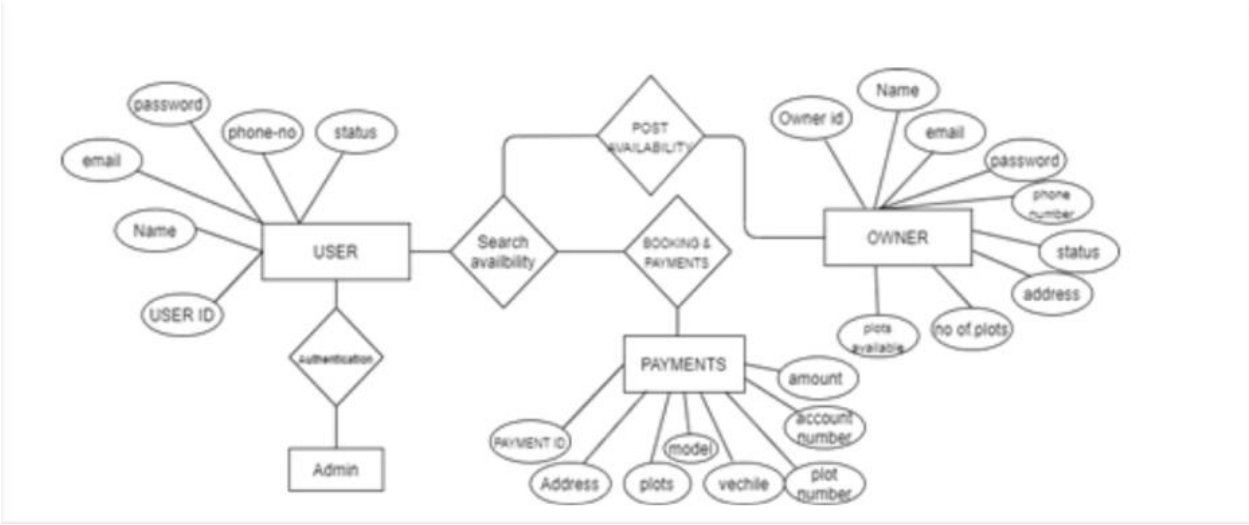


FIG.4.1 ER DIAGRAM OF ONLINE PARKING SPACE SHARING MECHANSIM WITH PRIVACY PROTECTION

4.2 DATA FLOW DIAGRAM:

A data flow diagram shows how information flows through a system or process. These include data inputs and outputs, data stores, and the various processes through which the data moves. DFDs use standard symbols and notations to describe the relationships between entities. The four components of data flow diagrams are,

- External entity
- Process
- Data store
- Data flow

LEVEL 0:

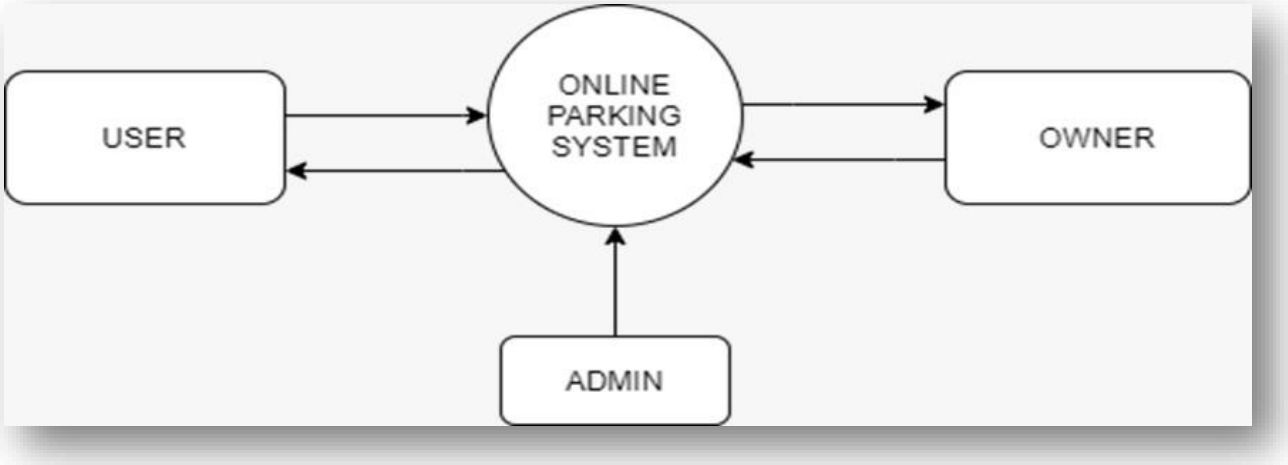


FIG 4.2 DFD LEVEL 0 OF ONLINE PARKING SPACE SHARING MECHANSIM WITH PRIVACY PROTECTION

LEVEL 1:

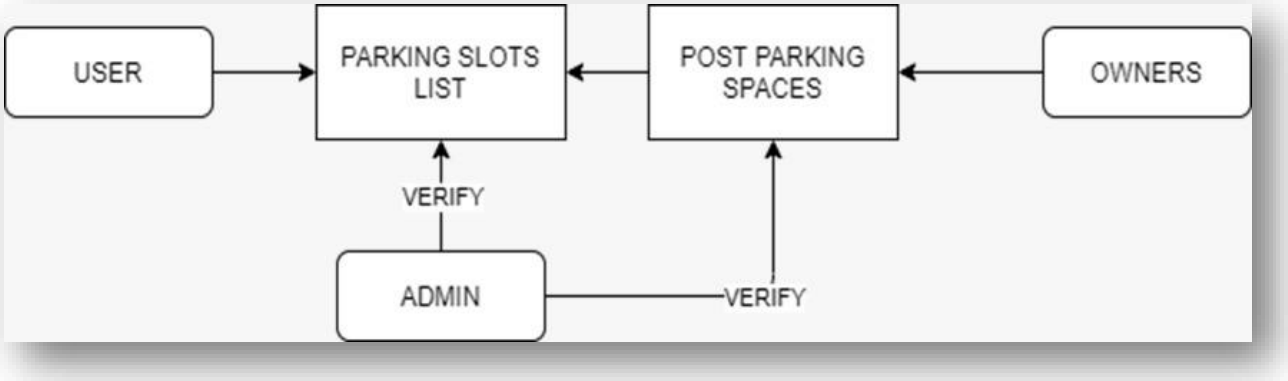


FIG 4.3 DFD LEVEL 1 OF ONLINE PARKING SPACE SHARING MECHANSIM WITH PRIVACY PROTECTION

LEVEL 2:

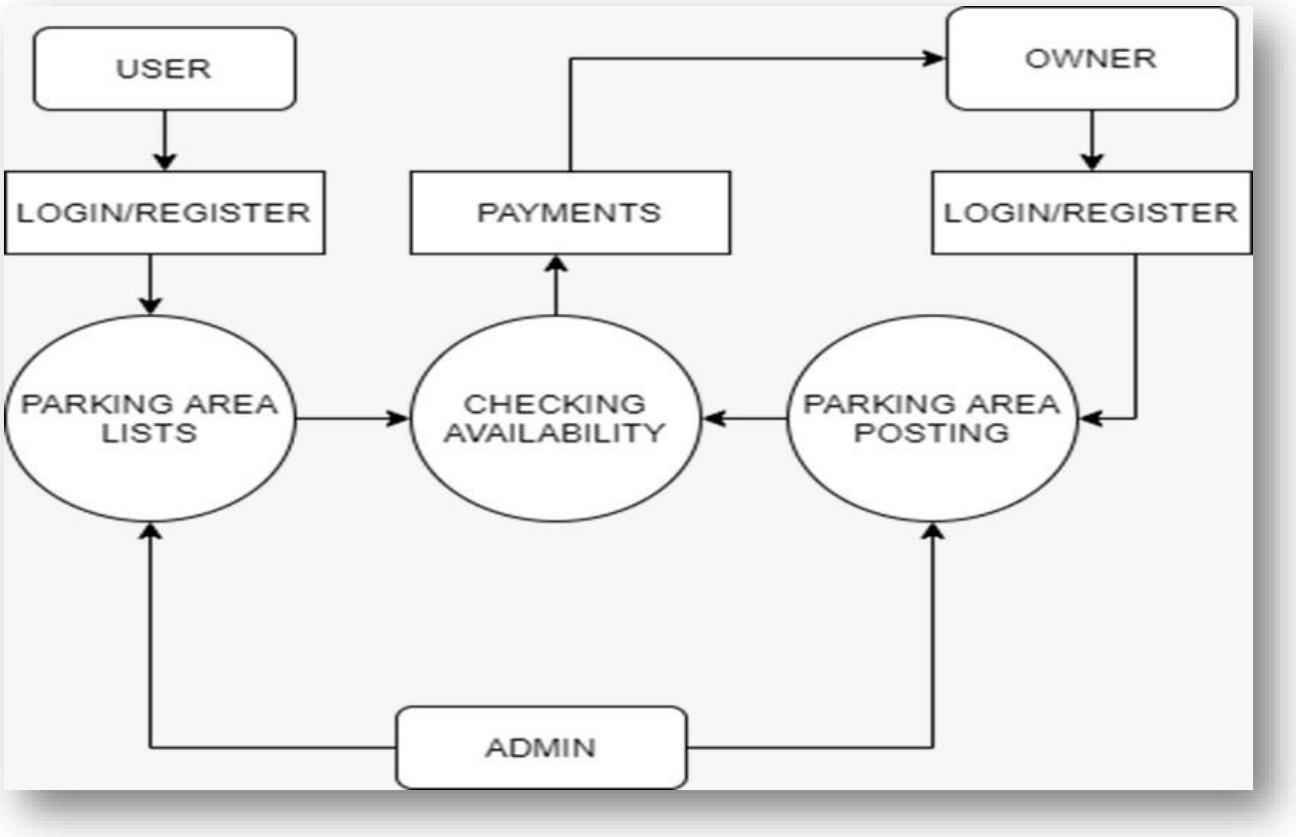


FIG 4.4 DFD LEVEL 2 OF ONLINE PARKING SPACE SHARING MECHANSIM WITH PRIVACY PROTECTION

OVERALL DFD:

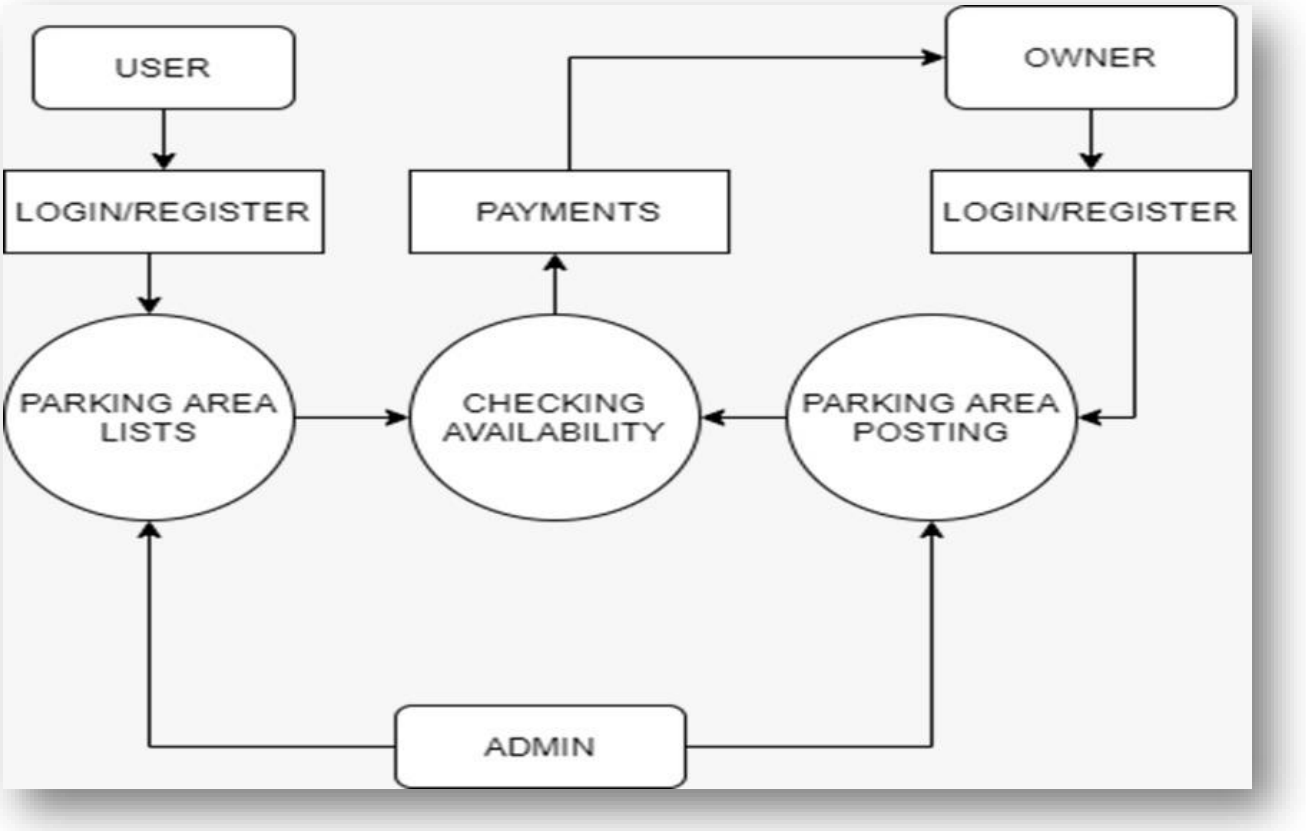


FIG 4.6 OVERALL DFD OF ONLINE PARKING SPACE SHARING MECHANSIM WITH PRIVACY PROTECTION

4.3 UML DIAGRAMS:

4.3.USE CASE:

A use case represents a particular functionality of a system. Use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

FIG 4.7 USECASE DIAGRAM OF ONLINE PARKING SPACE SHARING MECHANISM WITH PRIVACY PROTECTION

4.3. SEQUENCE DIAGRAM:

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. The sequence diagram includes a group of objects which are represented by lifelines, and the messages they exchange over time during their interaction.Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design level models for modern business application development.

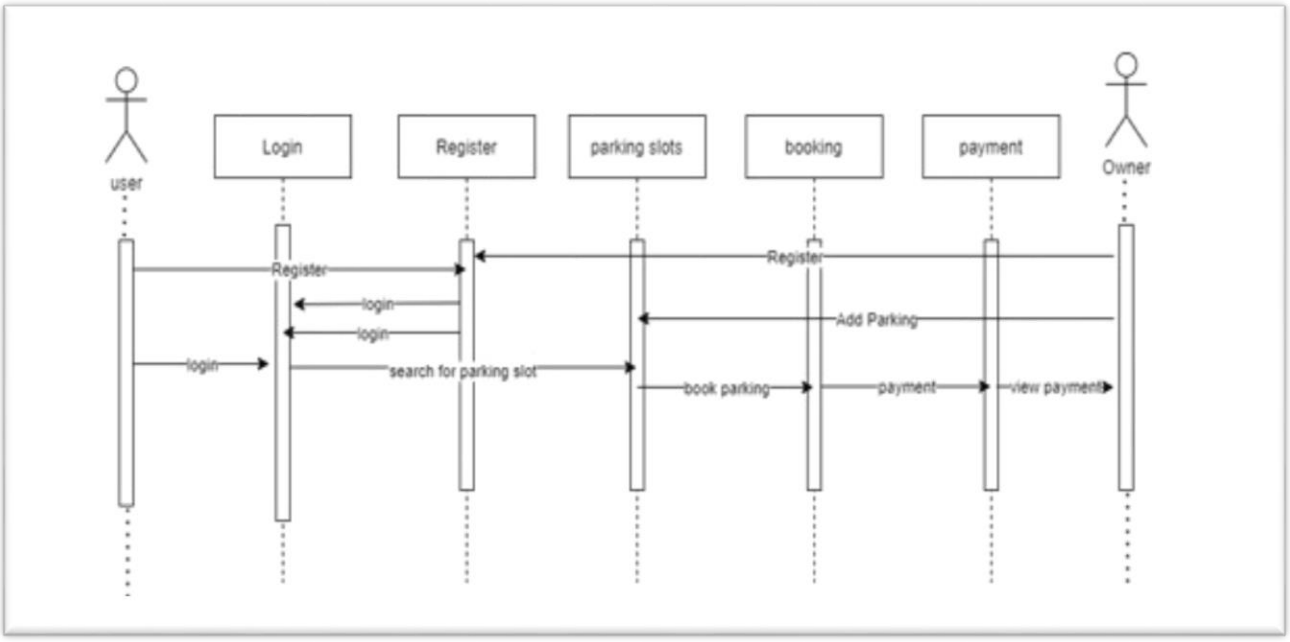


FIG 4.8 SEQUENCE DIAGRAM OF ONLINE PARKING SPACE SHARING MECHANISM WITH PRIVACY PROTECTION

4.3. COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram. It is an illustration of the relationships and interactions among software objects in the UML. Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links.

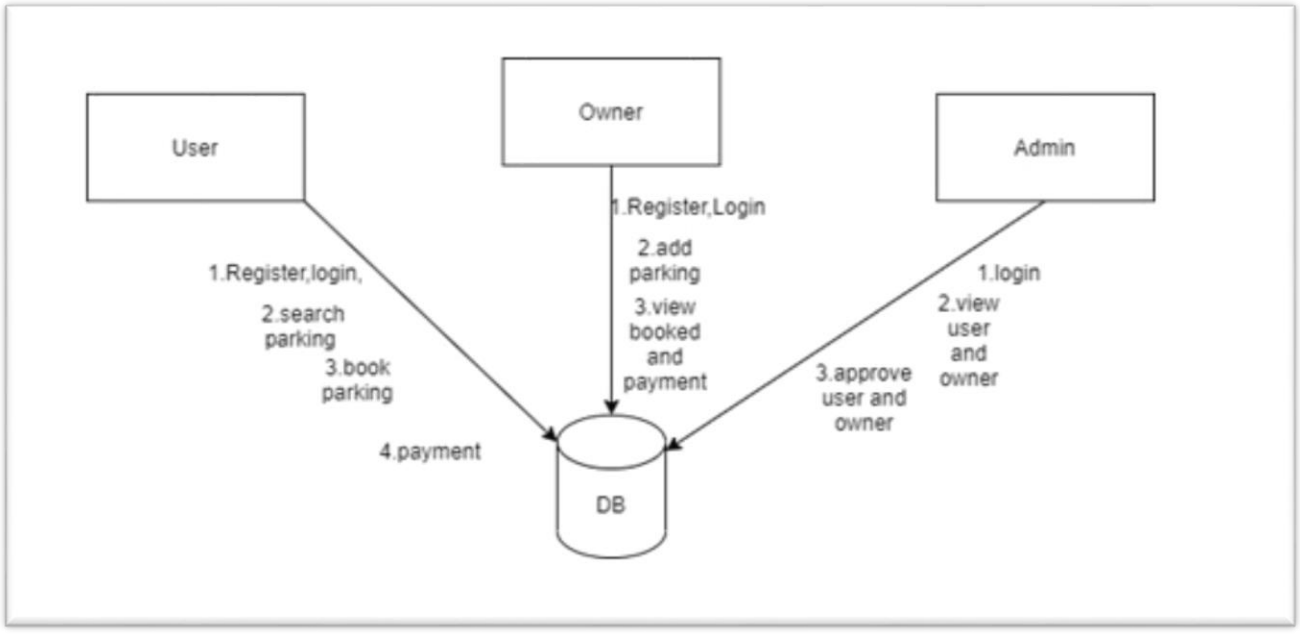


FIG 4.9 COLLABORATION DIAGRAM OF ONLINE PARKING SPACE SHARING MECHANISM WITH PRIVACY PROTECTION

4.3. ACTIVITY DIAGRAM:

Activity diagram represents the flow of activities from one to another. The activity can be described as an operation of the system. An activity diagram captures the dynamic behaviour of the system. It is also known as an object-oriented flowchart. Activity

diagrams include swim lanes, branching, parallel flow, control nodes, expansion nodes, and object nodes.

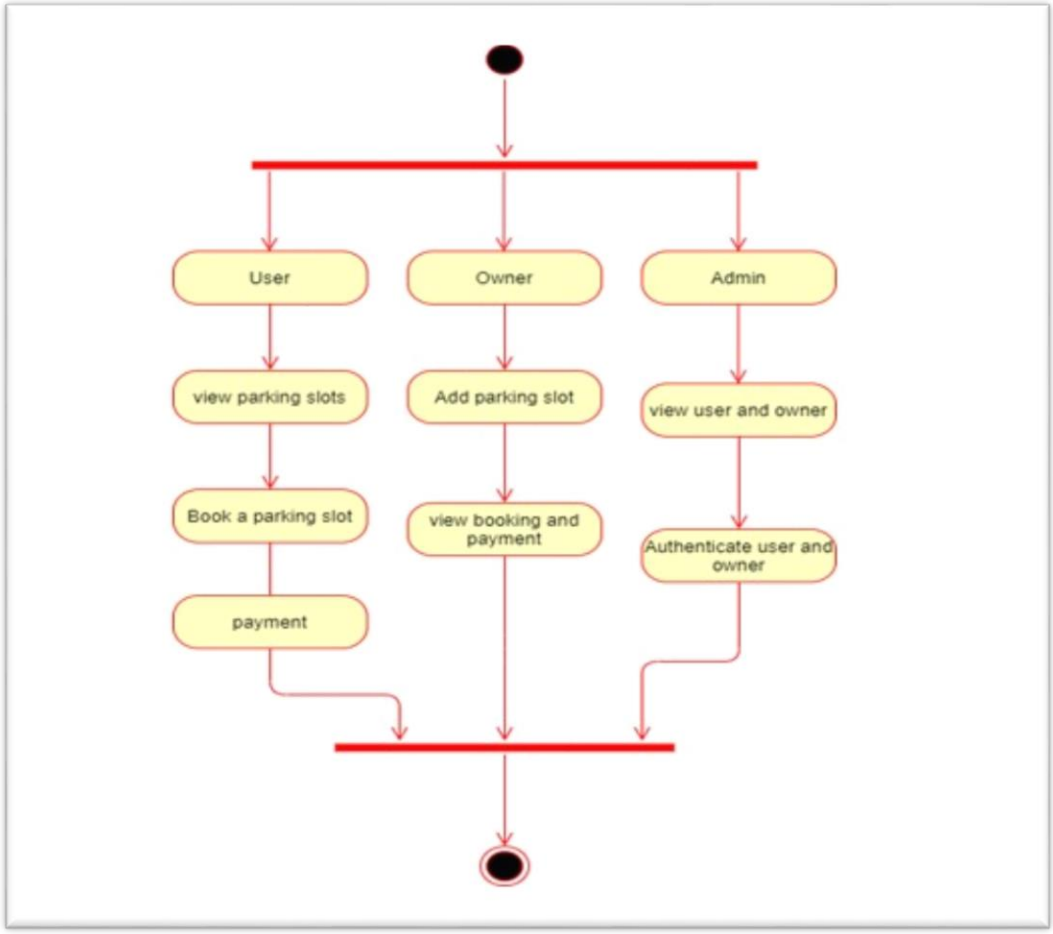


FIG 4.10 ACTIVITY DIAGRAM OF ONLINE PARKING SPACE SHARING MECHANISM WITH PRIVACY PROTECTION

4.4 DATA DICTIONARY

4.4 USER

COLUMN NAME	DATA TYPE	CONSTRAINTS
Id	Integer	Primary key
Username	Varchar	Not null
Password	Varchar	Not null
Email	Varchar	Not null
Phone number	Integer	Not null

4.4 OWNER

COLUMN NAME	DATA TYPE	CONSTRAINTS
Id	Integer	Primary Key
User name	Varchar	Not null
Password	Varchar	Not null
Email	Varchar	Not null
Phone number	Integer	Not null
Status	Varchar	Not null
Address	Varchar	Not null
No. of plots	Integer	Not null
Plot available	Array	Not null

4.4 ADMIN

COLUMN NAME	DATA TYPE	CONSTRAINTS
Id	Integer	Primary Key
User name	Varchar	Not null
Password	Varchar	Not null

4.4 PAYMENTS

COLUMN NAME	DATA TYPE	CONSTRAINTS
Payment – Id	Integer	Primary Key
Address	Varchar	Not null
Plot	Varchar	Not null
Plot no.	Integer	Not null
Vehicle	Varchar	Not null
Vehicle model	Varchar	Not null
Status	Varchar	Not null
Email	Varchar	Not null
Account number	Integer	Not null
Plot available	Varchar	Not null
Starting time	Varchar	Not null
Ending time	Varchar	Not null
Payment	Varchar	Not null

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 SYSTEM OVERVIEW

The system architecture is the basic structure of the system. The chest x-ray image is taken from the dataset and the image will be preprocessed using convolutional neural network. By using preprocessed image model is c

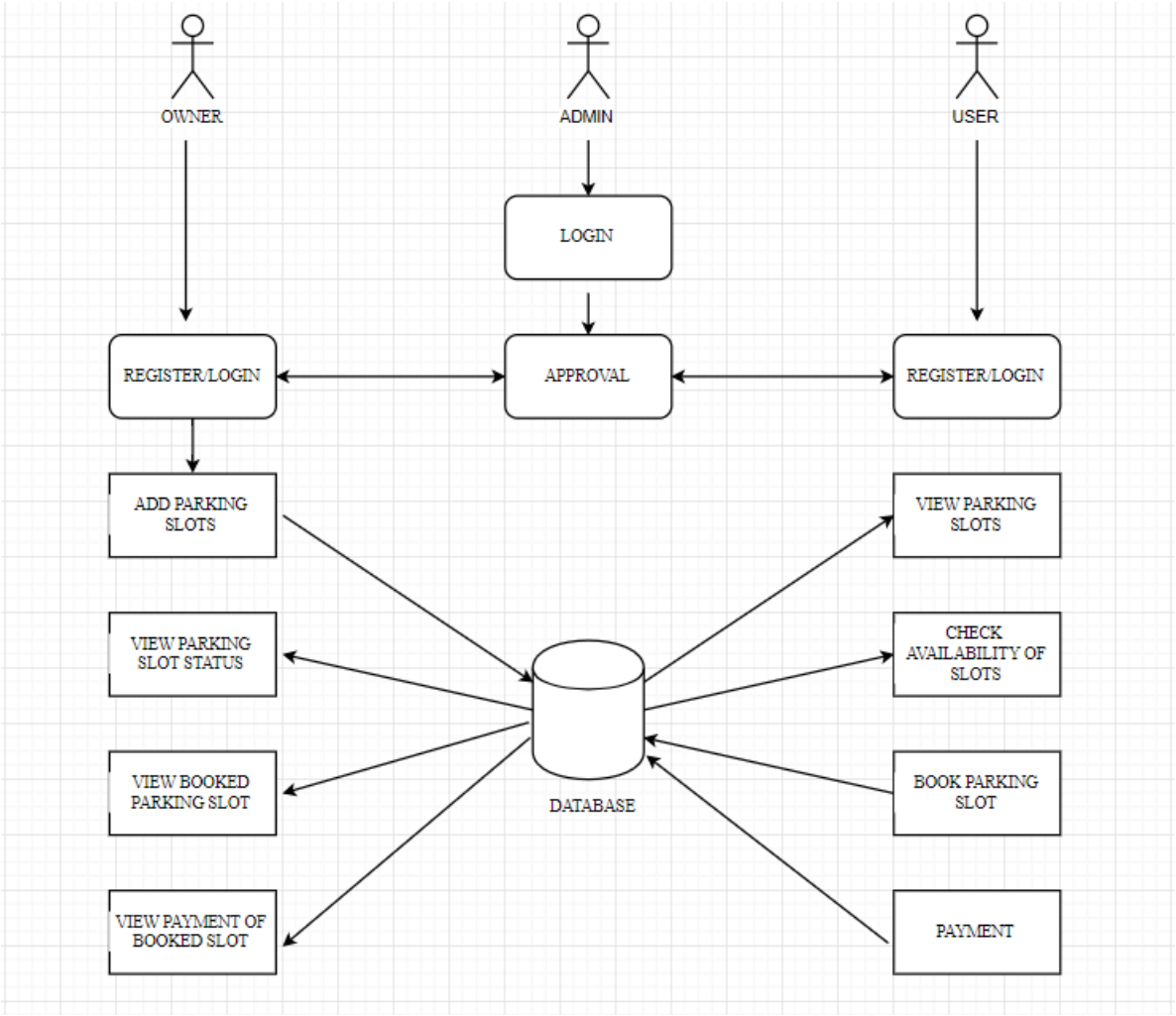


FIG.5.1 SYSTEM ARCHITECTURE OF ONLINE PARKING SPACE SHARING MECHANISM WITH PRIVACY PROTECTION

5.2 MODULES, DESIGN SPECIFICATION

The following are the list of modules of online parking space sharing mechanism with privacy protection

- Login\Register
- Admin
- User
- Owner
- Parking zone
- Payment

Login /registration- Module

Registration module is used to register the details about the user. That contain create a unique name and password. That also needs a full name of user and email id of user for authentication. The basic module login is used to web page. The module has username and password. That will be verified with database and allow to login to the web page.

Admin–Module:

This module is used to verify the user, its helps to prevent from the unauthorized problems. Admin add the owners for the parking availability.

User-Module:

The user module is used to reserve the parking slots for their purpose and required timing. User can pay the payment for their reserving parking slot, it helps reduce the time and traffic in public place.

Owner-Module:

The purpose of owner module is post the availability of their parking areas and allots the parking slot for the specified pre-booking user. Owner can receive the payments from user for reserved parking slots.

Parking zone -Module:

This module is used to get the get the details of parking slots from the owners and show the parking slots to the users. They can see the empty parking slots whenever chosen areas.

Payment Module:

The payment module is used to user pay the deserved amount for the selected parking slots.

5.2. ALGORITHM:

5.2. DIJKSTRA'S ALGORITHM :

- This algorithm finds the shortest path between a given node (which is called the "source node") and all other nodes in a graph.
- This uses the weights of the edges to find the path that minimizes the total distance (weight) between the source node and all other nodes.
- Suppose if we draw a graph in which nodes represent the cities and weighted edges represent the driving distances between pairs of cities connected by a direct road, then Dijkstra's algorithm when applied gives the shortest route between one city and all other cities.

ALGORITHM

- Let a node be allotted as the preliminary node. The distance of node Y be the gap among preliminary node and Y. This is a step by step process where the initial distance is allocated beforehand.
- Initially mark all nodes present as unvisited and group them as an unvisited set.
- Set tentative distance value for every node: set it to zero for our initial node and to infinity for all other nodes. Set the initial node as current.

- From the current node, consider all of its unvisited neighbors and calculate distance between them and assign the smallest value.
- Remove all unvisited nodes that are visited from the current node from the unvisited set. A visited node will never be checked again.
- If the destination node has been marked visited or if the smallest tentative distance among the nodes in the unvisited set is infinity, then stop. The algorithm has finished.
- If not, then select the unvisited node marked with the smallest tentative distance and set it as the new "current node", and repeat step iv.

BREADTH FIRST SEARCH

For entering, when all the spots are free:

1. Firstly the green color symbolizes all the spots that are free or Visited[node]=0

2. Then upon request from the vehicle driver for a free parking space, the algorithm will find a free spot for the vehicle. The node which has been found will then be assigned as visited[node]=" vehicle number";

Here, SLOT34 spot will be assigned for the vehicle.

For entering, when some spots are occupied:

In this case, we use the color Red to represent occupied cells.

1. The vehicle driver will request for a spot and the algorithm will run and find the next free space for the vehicle. Here the color Yellow represents the spot that has been found.

2. Upon finding a spot, it will assign the vehicle a spot say SLOT34. Visited["SLOT34"]="Vehicle Number". The algorithm will work in this manner to assign parking spots upon getting requests for free parking spaces.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 CLIENT-SIDE CODING:

```
<style>
* {box-sizing: border-box}
body {font-family: "Lato", sans-serif;}

/* Style the tab */
.tab {
  float: left;
  border: 1px solid #ccc;
  background-color: #f1f1f1;
  width: 30%;
  height: auto;
}

/* Style the buttons inside the tab */
.tab button {
  display: block;
  background-color: inherit;
  color: black;
  padding: 22px 16px;
  width: 100%;
  border: none;
  outline: none;
  text-align: left;
```

```
  cursor: pointer;
  font-size: 17px;
}

/* Change background color of buttons on hover */
.tab button:hover {
  background-color: #ddd;
}

/* Create an active/current "tab button" class */
.tab button.active {
  background-color: #ccc;
}

/* Style the tab content */
.tabcontent {
  float: left;
  padding: 0px 12px;
  border: 1px solid #ccc;
  width: 70%;
  border-left: none;
  height: auto;
  display: none;
}

/* Clear floats after the tab */
.clearfix::after {
  content: "";
```

```
  text-align: left;
  padding: 12px;
}

#myTable tr {
  border-bottom: 1px solid #ddd;
}

#myTable tr.header, #myTable tr:hover {
  background-color: #f1f1f1;
}

table {
  border-collapse: collapse;
  border-spacing: 0;
  width: 100%;
  border: 1px solid #ddd;
}

th, td {
  text-align: left;
  padding: 8px;
}

.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  max-width: 300px;
  margin: auto;
```

```
 (126, 'chennai', 'nungampakkam', 12.9863, 80.2432, 9874563210, 3, 'No', 2),
(136, 'chennai', 'tharamani', 12.9863, 80.2432, 9966887744, 0, '6', 1),
(137, 'chennai', 'solinganallur', 12.9863, 80.2432, 7410852096, 0, '10', 5),
(138, 'chennai', 'T-nagar', 12.9863, 80.2432, 9966887744, 0, '40', 6),
(139, 'chennai', 'kundarthur', 12.9863, 80.2432, 9966887744, 0, '41', 7);

CREATE TABLE `admin` (
  `email` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
-- Dumping data for table `admin`
INSERT INTO `admin` (`email`, `password`) VALUES
('hari@gmail.com', '123456');
-- Table structure for table `booking`
CREATE TABLE `booking` (
  `booking_id` int(20) NOT NULL,
  `property-id` int(20) NOT NULL,
  `user_id` int(20) NOT NULL,
  `fdate` date NOT NULL,
  `tdate` date NOT NULL,
  `stime` time NOT NULL,
  `etime` int(24) NOT NULL,
  `payment` text NOT NULL,
  `slot_d` int(11) NOT NULL,
  `owner_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO `owner` (`owner_id`, `full_name`, `email`, `password`, `phone_no`, `address`, `id_photo`, `approvel`) VALUES
(1, 'Nikesh ', 'nikesh3230@gmail.com', '12345', 9596939874, 'Kirtipur-3', 'owner-photo/nikesh.png', 'Yes'),
(2, 'yyy', 'yyy@gmail.com', '123456', 9874563210, 'kpks', 'owner-photo/img.jpg', 'Yes'),
(5, 'anbu', 'anbu@gmail.com', 'qwe', 7410852096, '251,anna nagar west', 'owner-photo/use.png', 'Yes'),
(6, 'MATHI', 'MATHI@GMAIL.COM', '123456', 8855774411, 'USMAN ROAD T NAGAR', 'owner-photo/avatar.png', 'Yes'),
(7, 'xxx', 'xxxx@gmail.com', 'qwer', 7788994455, 'kpks nagar', 'owner-photo/02.png', 'Yes'),
(8, 'sathis', 'sathis@gmail.com', '1233', 7410852096, 'USMAN ROAD T NAGAR', 'owner-photo/fp.jpg', 'no');

-- --------------------------------------------------------

--
-- Table structure for table `property_photo`
--

CREATE TABLE `property_photo` (
  `property_photo_id` int(12) NOT NULL,
  `p_photo` varchar(500) NOT NULL,
  `property_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
```

```
-- Dumping data for table `property_photo`
--

INSERT INTO `property_photo` (`property_photo_id`, `p_photo`, `property_id`) VALUES
(24, 'product-photo/img1.jpg', 123),
(25, 'product-photo/img3.jpg', 125),
(26, 'product-photo/img.jpg', 126),
(36, 'product-photo/logo1.jpg', 136),
(37, 'product-photo/home.jpg', 137),
(38, 'product-photo/home1.jpg', 138),
(39, 'product-photo/fp.jpg', 139);

-- --------------------------------------------------------

--
-- Table structure for table `slot`
--

CREATE TABLE `slot` (
  `slot_d` int(50) NOT NULL,
  `property_id` int(11) NOT NULL,
  `available` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `slot`
--
```

```
INSERT INTO `slot` (`slot_d`, `property_id`, `available`) VALUES
(1, 136, 'no'),
(2, 136, 'no'),
(3, 136, 'no'),
(4, 136, 'yes');

-- --------------------------------------------------------

--
-- Table structure for table `user`
--

CREATE TABLE `user` (
  `user_id` int(10) NOT NULL,
  `full_name` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `vehicle_m` varchar(100) NOT NULL,
  `vehicle_n` varchar(100) NOT NULL,
  `id_type` varchar(100) NOT NULL,
  `id_photo` varchar(1000) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `user`
--
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101
SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101
SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */
```

6.2 SERVER-SIDE CODING:

```
<?php
session_start();
if(!isset($_SESSION["email"])){
  header("location:../index.php");
}

include("navbar.php");

 ?>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="style.css">
</head>

<body style="background-color: #34edd3">

 <div class="container-fluid">
  <ul class="nav nav-pills nav-justified">
    <li class="active" style="background-color: #FFF8DC"><a data-toggle="pill"
href="#home">Property Lists</a></li>
    <li style="background-color: #FAF0E6"><a data-toggle="pill"
href="#menu1">Owners Details</a></li>
    <li style="background-color: #FFFACD"><a data-toggle="pill"
href="#menu2">Tenant Details</a></li>
    <li style="background-color: #FAFACD"><a data-toggle="pill"
href="#menu3">Booked Property</a></li>
  </ul>
  <div class="tab-content">
```

```
   <div id="home" class="tab-pane fade in active">
     <center><h3>Parking Lists</h3></center>
     <div class="container-fluid">
     <input type="text" id="myInput" onkeyup="myFunction()"
placeholder="Search..." title="Type in a name">
          <div style="overflow-x:auto;">
            <table id="myTable">
              <tr class="header">
                <th>Id.</th>
                <th>City</th>
                <th>Zone</th>
                <th>Contact No.</th>
                <th>Owner Id.</th>
                <th>Property_photo.</th>

              </tr>
              <?php
      include("../config/config.php");

      $sql="SELECT * from add_property";
      $result=mysqli_query($db,$sql);

      if(mysqli_num_rows($result)>0)
     {
        while($rows=mysqli_fetch_assoc($result)){
        $property_id=$rows['property_id'];
      ?>
              <tr>
                <td><?php echo $rows['property_id'] ?></td>
                <td><?php echo $rows['city'] ?></td>
                <td><?php echo $rows['zone'] ?></td>
                 <td><?php echo $rows['contact_no'] ?></td>
                <td><?php echo $rows['owner_id'] ?></td><td>
<?php $sql2="SELECT * from property_photo where property_id='$property_id'";
      $query=mysqli_query($db,$sql2);

      if(mysqli_num_rows($query)>0)
     {
        while($row=mysqli_fetch_assoc($query)){ ?>
              <img src="../owner/<?php echo $row['p_photo'] ?>" width="50px">
             <?php }}}} ?>
             </td>
```

```
    <center><h3>Booked Property</h3></center>
    <div class="container">
      <input type="text" id="myInput4" onkeyup="myFunction4()"
placeholder="Search..." title="Type in a name">

          <table id="myTable4">
           <tr class="header">
             <th>Booking  Id</th>
             <th>Property Id</th>
             <th>User Id</th>
             <th>Starting Date</th>
             <th>Ending Date</th>
             <th>Time</th>
             <th>Amount</th>
             <th>Payment Status</th>
           </tr>

    <?php
     include("../config/config.php");


     $sql="SELECT * from booking";
     $result=mysqli_query($db,$sql);

     if(mysqli_num_rows($result)>0)
    {
       while($rows=mysqli_fetch_assoc($result)){

     ?>
           <tr>
             <td><?php echo $rows['booking_id'] ?></td>
             <td><?php echo $rows['property-id'] ?></td>
             <td><?php echo $rows['user_id'] ?></td>
             <td><?php echo $rows['fdate'] ?></td>
             <td><?php echo $rows['tdate'] ?></td>
             <td><?php echo $rows['stime'] ?></td>
             <td><?php echo $rows['etime'] ?></td>
             <td><?php echo $rows['payment'] ?></td>
           </tr>

          <?php }} ?>
          </table>
```

```
  th = table.getElementsByTagName("th");
  for (i = 1; i < tr.length; i++) {
    tr[i].style.display = "none";
      for(var j=0; j<th.length; j++){
        td = tr[i].getElementsByTagName("td")[j];
        if (td) {
          if (td.innerHTML.toUpperCase().indexOf(filter.toUpperCase()) > -1)
          {
            tr[i].style.display = "";
            break;
          }
        }
      }
  }
}
</script>

<script>
function myFunction3() {
  var input, filter, table, tr, td, i, txtValue;
  input = document.getElementById("myInput3");
  filter = input.value.toUpperCase();
  table = document.getElementById("myTable3");
  tr = table.getElementsByTagName("tr");
  th = table.getElementsByTagName("th");
  for (i = 1; i < tr.length; i++) {
    tr[i].style.display = "none";
      for(var j=0; j<th.length; j++){
        td = tr[i].getElementsByTagName("td")[j];
        if (td) {
          if (td.innerHTML.toUpperCase().indexOf(filter.toUpperCase()) > -1)
          {
            tr[i].style.display = "";
            break;
          }
        }
      }
  }
}
</script>
<script>
function myFunction4() {
```

```
</body>
</html>
 navbar.php
 <?php

include("navbar.php");

 ?>
<body style="background-color: #34edd3">
<div class="container">
  <h3 style="font-weight: bold; text-align: center;">Owner Register</h3><hr><br>
  <form method="POST" action="owner-engine.php" enctype="multipart/form-data">
    <div class="form-group">
      <label for="full_name">Full Name:</label>
      <input type="text" class="form-control" id="full_name" placeholder="Enter Full
Name" name="full_name" required>
    </div>
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" class="form-control" id="email" placeholder="Enter Email"
name="email" required>
    </div>
    <div class="form-group">
      <label for="password1">Password:</label>
      <input type="password" class="form-control" id="password1" placeholder="Enter
Password" name="password" required>
    </div>
    <div class="form-group">
      <label for="password2">Confirm Password:</label>
      <input type="password" class="form-control" id="password2" placeholder="Enter
Password Again" required>
    </div>
    <div class="form-group">
      <label for="phone_no">Phone No.:</label>
      <input type="text" class="form-control" id="phone_no" placeholder="Enter
Phone No." name="phone_no" required>
    </div>

    <div class="form-group">
      <label for="address">Address:</label>
      <input type="text" class="form-control" id="address" placeholder="Enter
Address" name="address" required>
```

```
$zone='';
$district='';
$city='';
$vdc_municipality='';
$ward_no='';
$tole='';
$contact_no='';
$property_type='';
$estimated_price='';
$total_rooms='';
$bedroom='';
$living_room='';
$kitchen='';
$bathroom='';
$description='';
$latitude='';
$longitude='';
$booked='';
$owner_id='';



$db = new mysqli('localhost','root','','parking');

if($db->connect_error){
	echo "Error connecting database";
}

if(isset($_POST['add_property'])){
	add_property();
}

if(isset($_POST['owner_update'])){
	owner_update();
}


function add_property(){

	global
$property_id,$country,$province,$zone,$district,$city,$vdc_municipality,$ward_no,$t
ole,$contact_no,$property_type,$estimated_price,$total_rooms,$bedroom,$living_roo
```

```
m,$kitchen,$bathroom,$description,$latitude,$path,$p_photo,$property_photo_id,$lo
ngitude,$owner_id,$booked,$db;

        $country=validate($_POST['country']);
        $province=validate($_POST['province']);
        $zone=validate($_POST['zone']);
        $district=validate($_POST['district']);
        $city=validate($_POST['city']);
        $vdc_municipality=validate($_POST['vdc_municipality']);
        $ward_no=validate($_POST['ward_no']);
        $tole=validate($_POST['tole']);
        $contact_no=validate($_POST['contact_no']);
        $property_type=validate($_POST['property_type']);
        $estimated_price=validate($_POST['estimated_price']);
        $total_rooms=validate($_POST['total_rooms']);
        $bedroom=validate($_POST['bedroom']);
        $living_room=validate($_POST['living_room']);
        $kitchen=validate($_POST['kitchen']);
        $bathroom=validate($_POST['bathroom']);
        $description=validate($_POST['description']);
        $latitude=validate($_POST['latitude']);
        $longitude=validate($_POST['longitude']);
        $booked='No';
        $u_email=$_SESSION['email'];
      $sql1="SELECT * from owner where email='$u_email'";
      $result1=mysqli_query($db,$sql1);

      if(mysqli_num_rows($result1)>0)
    {
       while($rowss=mysqli_fetch_assoc($result1)){
         $owner_id=$rowss['owner_id'];

        $sql = "INSERT INTO
add_property(country,province,zone,district,city,vdc_municipality,ward_no,tole,conta
ct_no,property_type,estimated_price,total_rooms,bedroom,living_room,kitchen,bathro
om,description,latitude,longitude,booked,owner_id)
VALUES('$country','$province','$zone','$district','$city','$vdc_municipality','$ward_n
o','$tole','$contact_no','$property_type','$estimated_price','$total_rooms','$bedroom','$l
iving_room','$kitchen','$bathroom','$description','$latitude','$longitude','$booked','$ow
ner_id')";
                $query=mysqli_query($db,$sql);
```

```
</style>
<script>
        window.setTimeout(function() {
    $(".alert").fadeTo(1000, 0).slideUp(500, function(){
        $(this).remove();
    });
}, 2000);
</script>
<div class="container">
<div class="alert" role='alert'>
  <span class="closebtn"
onclick="this.parentElement.style.display='none';">&times;</span>
  <center><strong>Your Product has been uploaded.</strong></center>
</div></div>


<?php
                }
                else{
                        echo "error";
                }

}
}}

function owner_update(){
        global $owner_id,$full_name,$email,$password,$phone_no,$address,$id…
 engine.php
 <!DOCTYPE html>
<html lang="en">
<head>
  <title>Parking</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</nav>

</body>
```

```
</html>
 navbar.php in owner
<?php
session_start();
if(!isset($_SESSION["email"])){
  header("location:../index.php");
}

include("navbar.php");
include("engine.php");

?>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body style="background-image:url('../images/ob.jpg')">
  <style type="text/css" style="background-color: #005aa4"></style>
  <ul class="nav nav-pills nav-justified" style="margin-top: -
15px;width:25%:padding-left:50px">


  </ul>
  <div class="tab-content">
    <div id="home" class="tab-pane fade in active">
      <center><h3 style="color: #fff">Owner Profile</h3></center>
      <div class="container" >
        <?php
        include("../config/config.php");
        $u_email= $_SESSION["email"];

        $sql="SELECT * from owner where email='$u_email'";
        $result=mysqli_query($db,$sql);

        if(mysqli_num_rows($result)>0)
        {
          while($rows=mysqli_fetch_assoc($result)){
         <div class="container">
          <center><h3>See Messages</h3></center>
          <?php
```

```
                </div>

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


                <?php
        //echo '<a href="send-
message.php?owner_id='.$owner_id.'&tenant_id='.$tenant_id.'">'.$rows["full_name"]
.'</a>';
                }
              }}}}}?>
            </div>
          </div>
 -->




        <!--   <div id="menu1" class="tab-pane fade">
            <center><h3>Add Property</h3></center>
            <div class="container">


              <div id="map_canvas"></div>
              <form method="POST" enctype="multipart/form-data">
                <div class="row">
                  <div class="col-sm-6">
                    <div class="form-group">
                      <label for="country">Country:</label>
                      <select class="form-control" name="country" required="required">
                        <option value="">--Select Country--</option>
                        <option value="Nepal">Nepal</option>
                      </select>
                    </div>
                    <div class="form-group">
                      <label for="province">Province/State:</label>
                      <select class="form-control" name="province" required="required">
                        <option value="">--Select Province/State--</option>
                        <option value="Province No. 1">Province No. 1</option>
```

```
                    while($rows=mysqli_fetch_assoc($result)){
                      $property_id=$rows['property_id'];
                      ?>
                      <tr>
                        <td><?php echo $rows['property_id'] ?></td>
                        <td><?php echo $rows['country'] ?></td>
                        <td><?php echo $rows['province'] ?></td>
                        <td><?php echo $rows['zone'] ?></td>
                        <td><?php echo $rows['district'] ?></td>
                        <td><?php echo $rows['city'] ?></td>
                        <td><?php echo $rows['vdc_municipality'] ?></td>
                        <td><?php echo $rows['ward_no'] ?></td>
                        <td><?php echo $rows['tole'] ?></td>
                        <td><?php echo $rows['contact_no'] ?></td>
                        <td><?php echo $rows['property_type'] ?></td>
                        <td><?php echo $rows['latitude'] ?></td>
                        <td><?php echo $rows['longitude'] ?></td>
                        <td>Rs.<?php echo $rows['estimated_price'] ?></td>
                        <td><?php echo $rows['total_rooms'] ?></td>
                        <td><?php echo $rows['bedroom'] ?></td>
                        <td><?php echo $rows['living_room'] ?></td>
                        <td><?php echo $rows['kitchen'] ?></td>
                        <td><?php echo $rows['bathroom'] ?></td>
                        <td><?php echo $rows['description'] ?></td><td>
                          <?php $sql2="SELECT * from property_photo where
property_id='$property_id'";
                          $query=mysqli_query($db,$sql2);

                          if(mysqli_num_rows($query)>0)
                          {
                            while($row=mysqli_fetch_assoc($query)){ ?>
                              <img src="<?php echo $row['p_photo'] ?>" width="50px">
                            <?php }}}}}} ?>
                          </td>
                        </tr>
                      </table>
                    </div>
                  </div>
                </div>

                <div id="menu3" class="tab-pane fade">
                  <center><h3>Update Property</h3></center>
```

```
                <div class="container-fluid">
                  <input type="text" id="myInput" onkeyup="updateProperty()"
placeholder="Search..." title="Type in a name">
                  <div style="overflow-x:auto;">
                    <table id="myTable">
                      <tr class="header">
                        <th>Id.</th>
                        <th>Country</th>
                        <th>Province/State</th>
                        <th>Zone</th>
                        <th>District</th>
                        <th>City</th>
                        <th>VDC/Municipality</th>
                        <th>Ward No.</th>
                        <th>Tole</th>
                        <th>Contact No.</th>
                        <th>Property Type</th>
                        <th>Latitude</th>
                        <th>Longitude</th>
                        <th>Estmated Price</th>
                        <th>Total Rooms</th>
                        <th>Bedroom</th>
                        <th>Living Room</th>
                        <th>Kitchen</th>
                        <th>Bathroom</th>
                        <th>Description</th>
                        <th>Photos</th>
                        <th>Edit/Delete</th>
                      </tr>
                      <?php

                      $sql="SELECT * from add_property where owner_id='$owner_id'";
                      $result=mysqli_query($db,$sql);

                      if(mysqli_num_rows($result)>0)
                      {
                        while($rows=mysqli_fetch_assoc($result)){
                          $property_id=$rows['property_id'];
                          ?>
                          <tr>
                            <td><?php echo $rows['property_id'] ?></td>
```

```
                        </div>
                        <div class="form-group">
                          <label for="living_room">No. of Living Room:</label>
                          <input type="number" class="form-control" id="living_room"
placeholder="Enter No. of Living Room" name="living_room">
                        </div>
                        <div class="form-group">
                          <label for="kitchen">No. of Kitchen:</label>
                          <input type="number" class="form-control" id="kitchen"
placeholder="Enter No. of Kitchen" name="kitchen">
                        </div>
                        <div class="form-group">
                          <label for="bathroom">No. of Bathroom/Washroom:</label>
                          <input type="number" class="form-control" id="bathroom"
placeholder="Enter No. of Bathroom/Washroom" name="bathroom">
                        </div>
                        <div class="form-group">
                          <label for="description">Full Description:</label>
                          <textarea type="comment" class="form-control"
id="description" placeholder="Enter Property Description"
name="description"></textarea>
                        </div>
                        <hr>
                        <div class="form-group">
                          <input type="submit" class="btn btn-primary btn-lg col-lg-12"
value="Update Property" name="add_property">
                        </div>
                      </div>
                    </div>
                  </form>
                  <br><br>

                </div>
              </div>

              <div id="menu6" class="tab-pane fade">
                <center><h3>Booked Property</h3></center>
                <div class="container">
                  <input type="text" id="myInput" onkeyup="bookedProperty()"
placeholder="Search..." title="Type in a name">
                            <tr>
```

CHAPTER 7

PERFORMANCE ANALYSIS

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides away to check the functionality of components, sub-assemblies, assemblies and/or afinished product It is the process of exercising software with the intent of ensuringthat the Software system meets its requirements and user expectations and does notfail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. TYPES OF TESTS Unit testing Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases**.**

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

TEST OBJECTIVES

• All field entries must work properly.

• Pages must be activated from the identified link

.• The entry screen, messages and responses must not be delayed.

FEATURES TO BE TESTED

• Verify that the entries are of the correct format

• No duplicate entries should be allowed

• All links should take the user to the correct page.

TEST RESULTS

All the test cases mentioned above passed successfully. No defects encountered. Acceptance Testing User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 8

8.1 CONCLUSION AND FUTURE ENHANCEMENT:

This shared parking allocation problems between parking demands in commercial buildings and parking supplies in residential zones. The concept of shared parking is proposed, which is according to the preconditions of shared parking implementation. Then, the feasibility of shared parking between parking requests from commercial buildings and private paid or public free parking lots in residential zones is initially evaluated by analyzing the characteristics of shared parking, which include win-win, convenience, economy, and real-time performance. Next, a bitrate parking spaces allocating model involving the minimum walking distance and the maximum utilization is proposed. The model comprehensively considers the drivers' walking distance and the utilization of parking spaces. It not only receives reception requests for buildings in commercial zones, but also assigns them to corresponding vacant parking lots in accordance with the model hypothesis and parking space-time constraints. PSO algorithms applied to solve the parking allocation model.

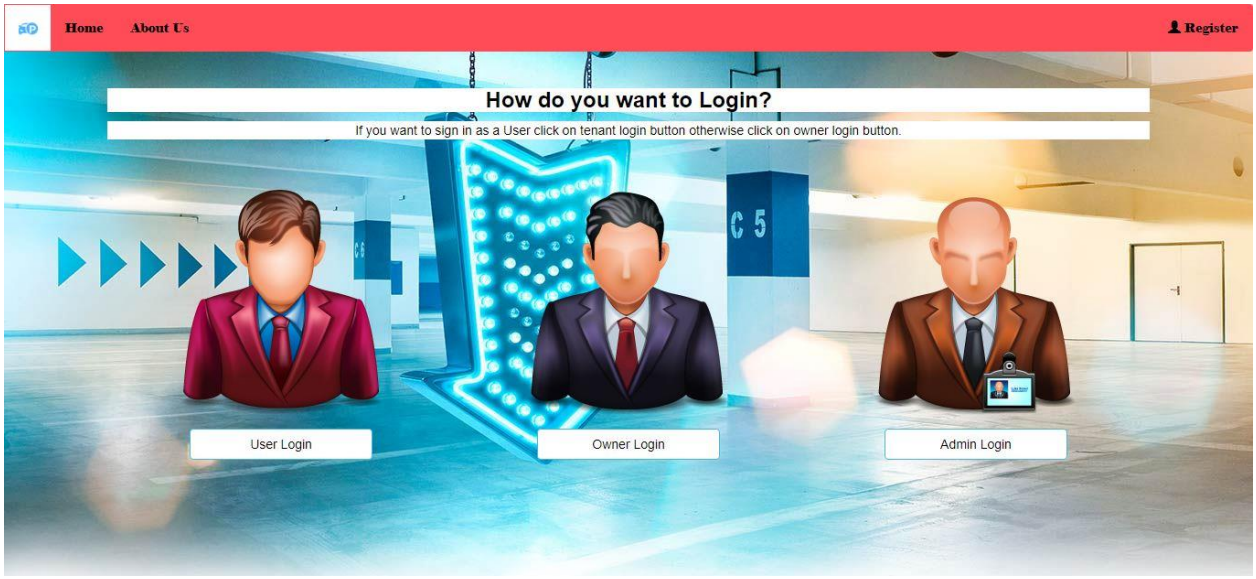
Future Enhancement:

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project. In this paper, we proposed a new solution for privacy-preserving user profile matching with holomorphic technique and multiple servers. Our solution allows a user to find out the matching users with the help of multiple servers without revealing the query and the user profiles. Security analyses have shown that the new protocol achieves user profile

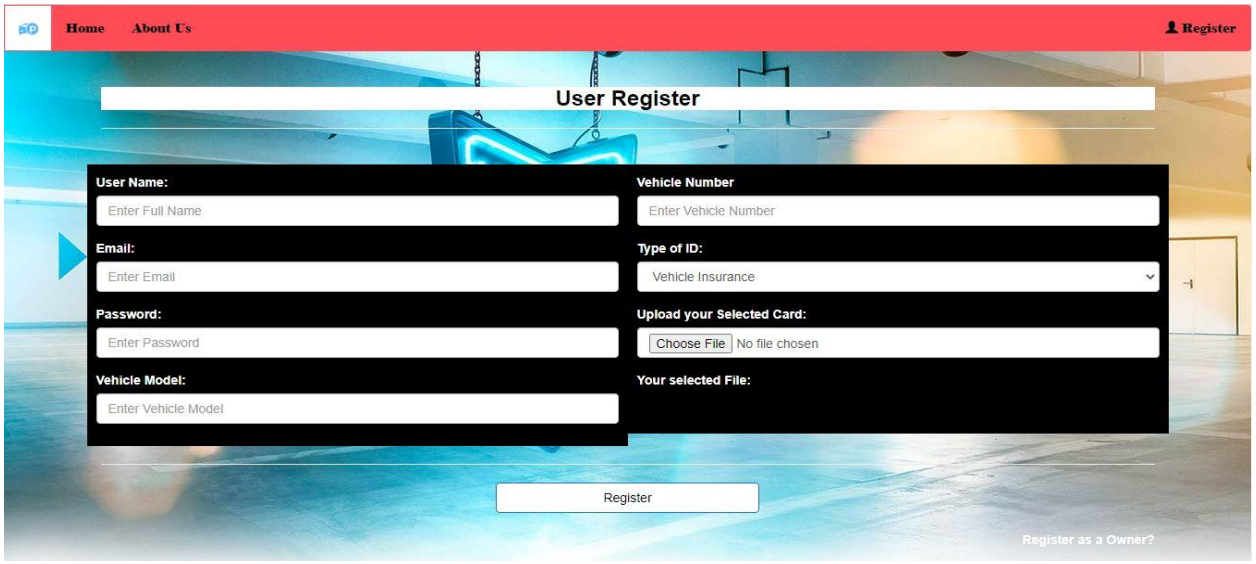
privacy and user query privacy. The experimental results have showed that the new protocol is practical and feasible. Our future work is to improve the performance of computing conditional gates by parallel computation.

APPENDICES

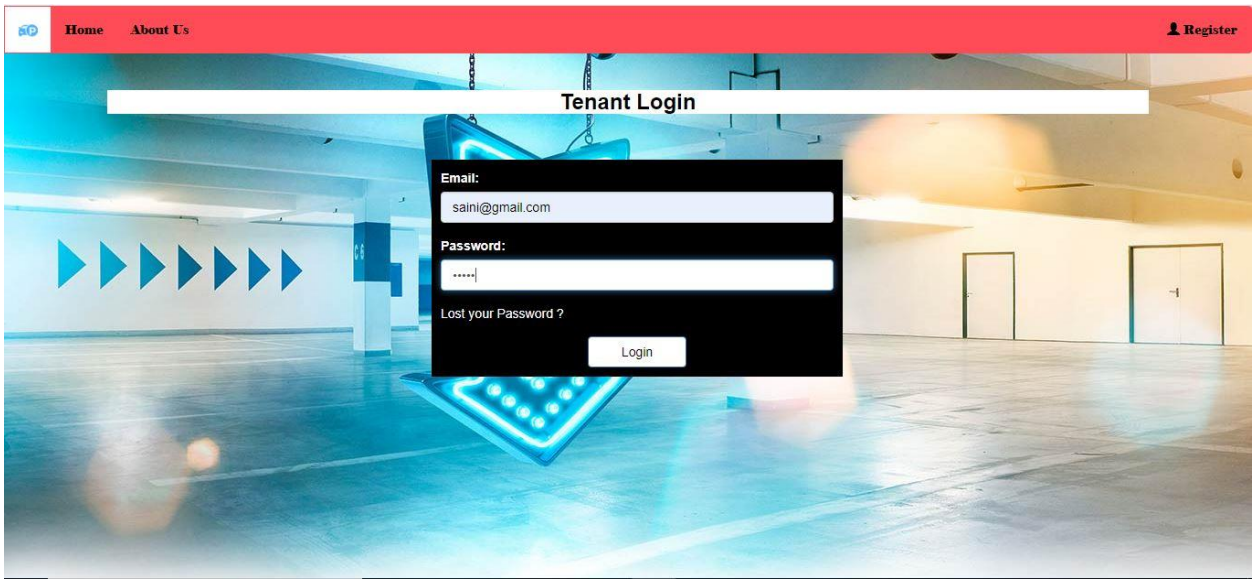
A1.SAMPLE SCREENSHOTS:



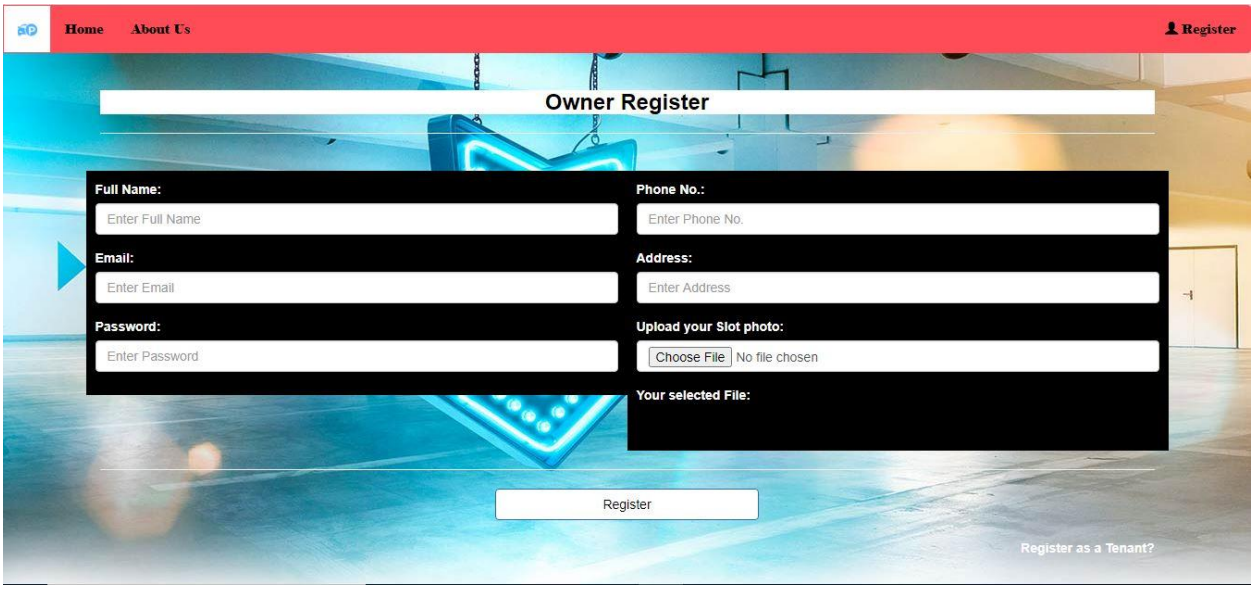
A.1 HOME PAGE



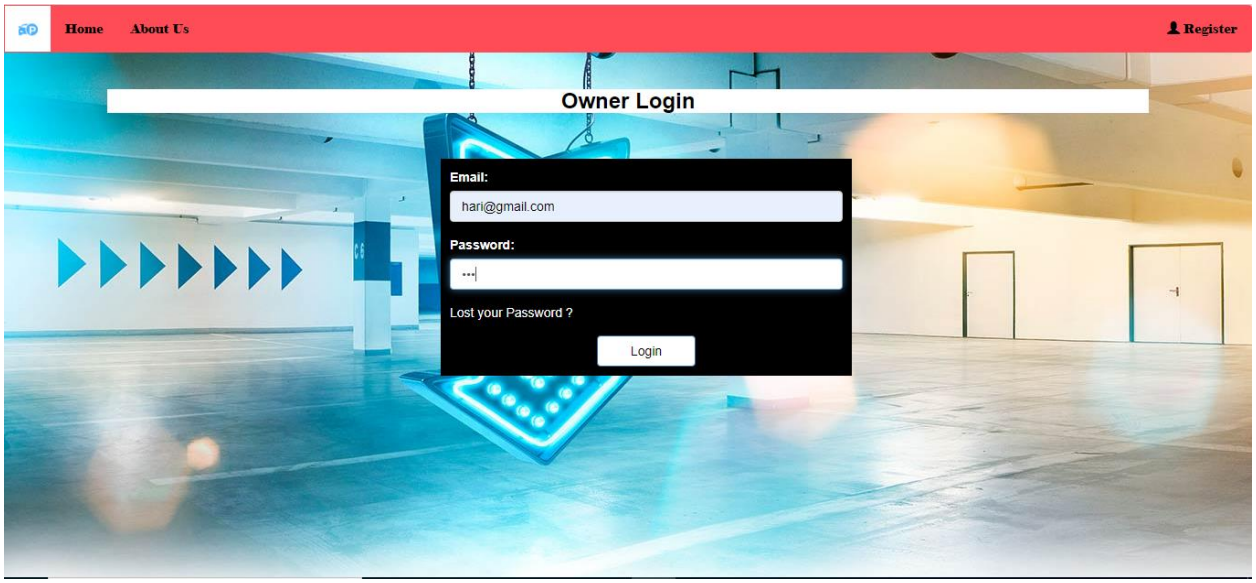
A.2 User Register



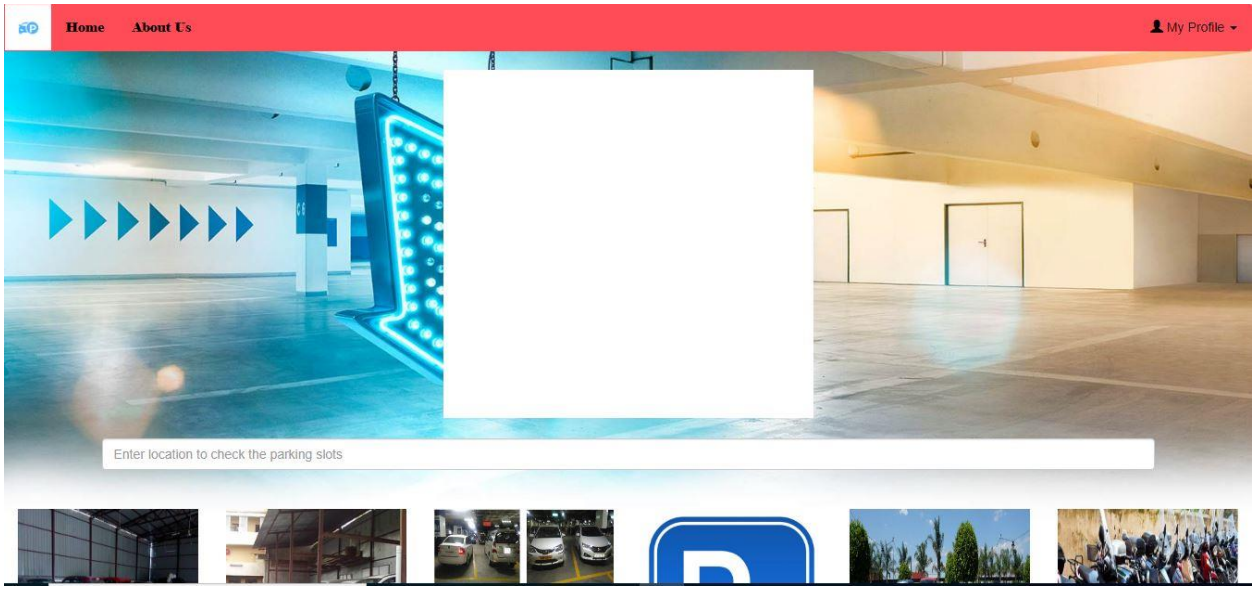
A.3 User Login



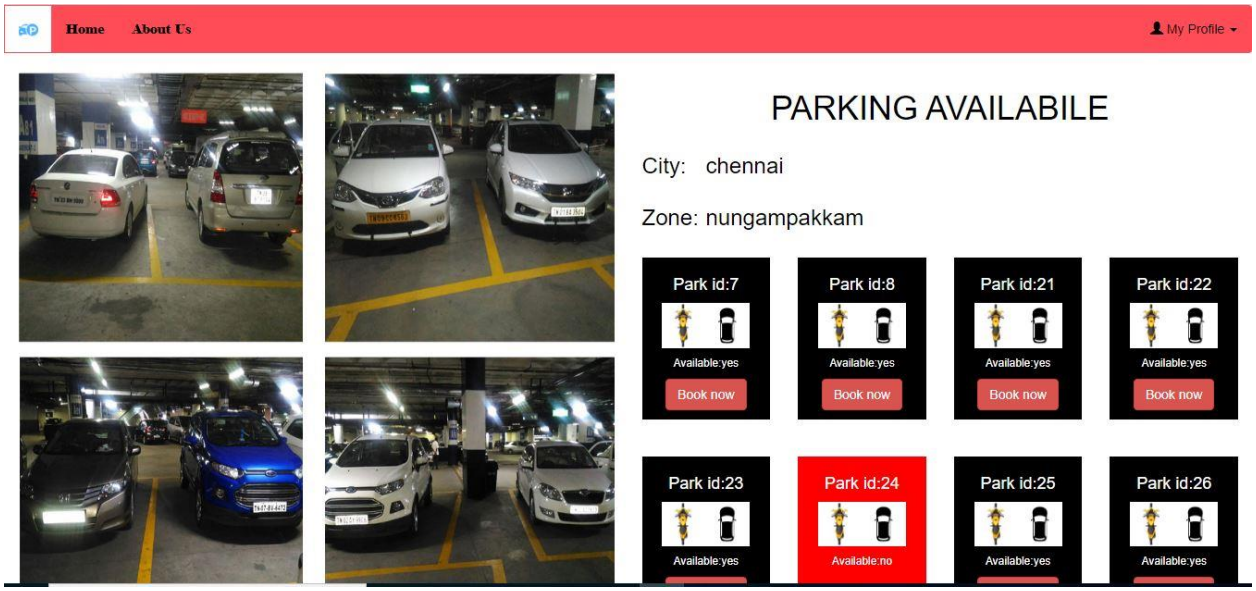
A.4 Owner Register



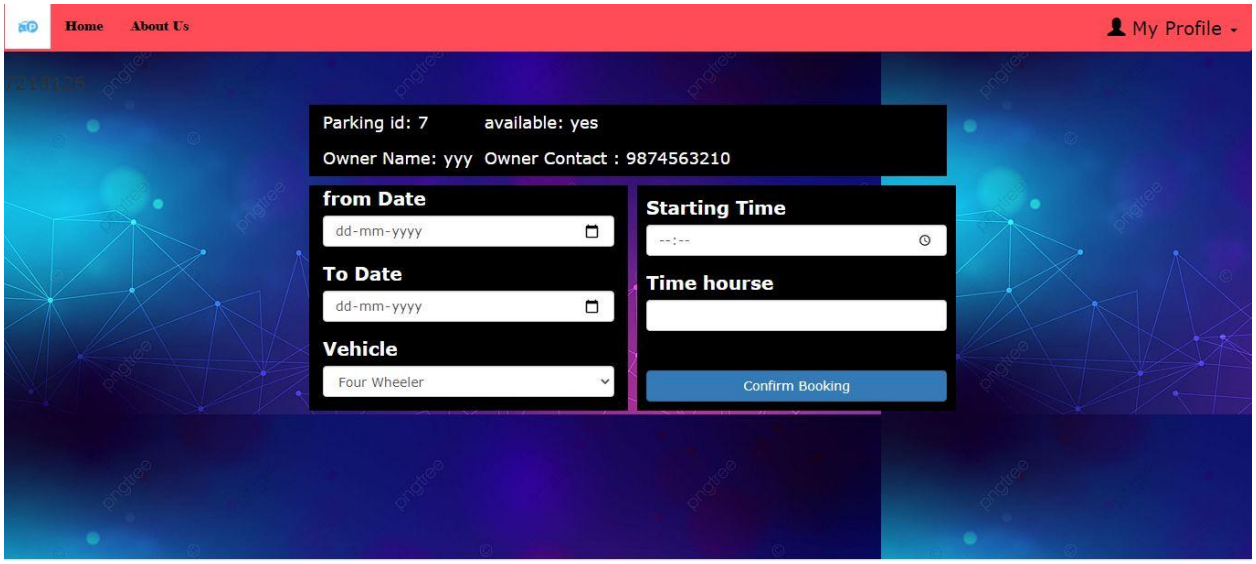
A.5 Owner Login



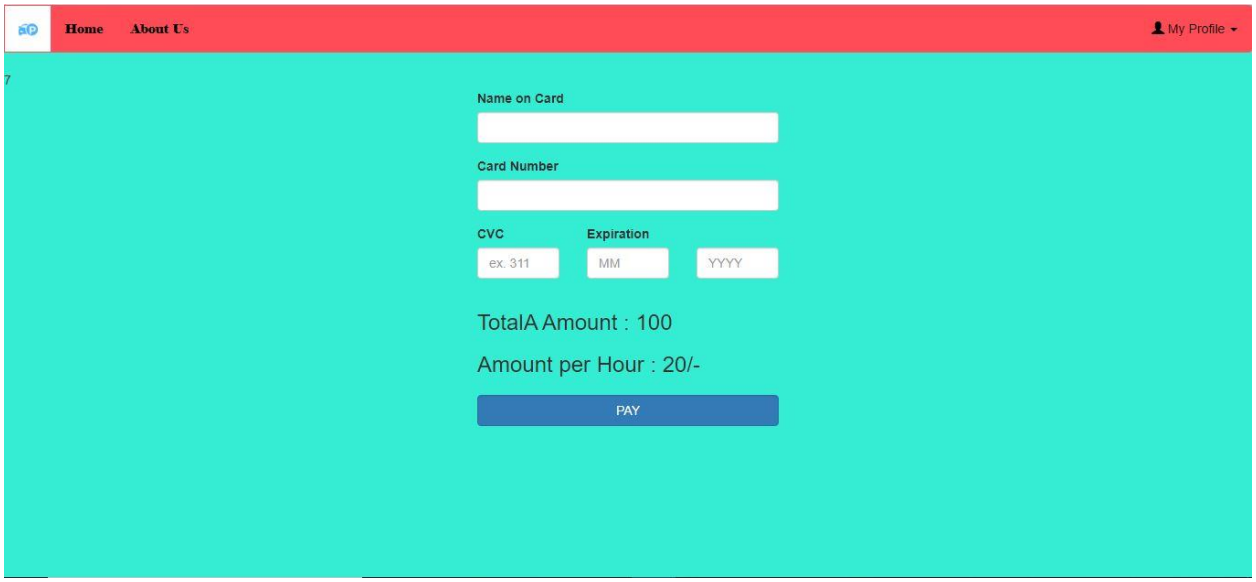
A.6 Home Page



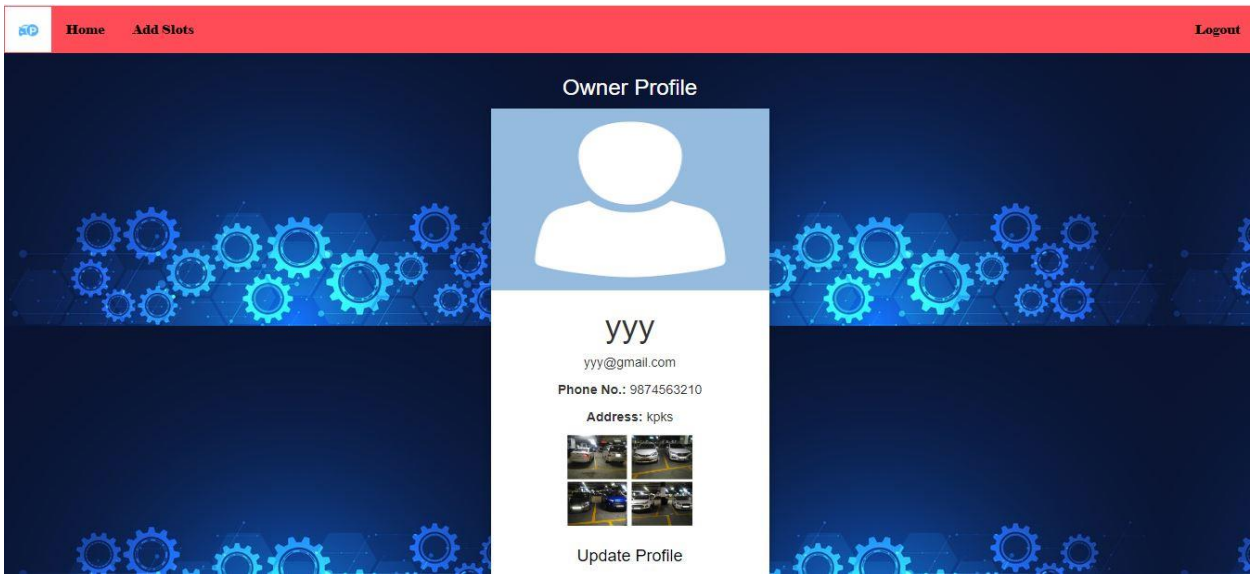
A.7: View Parking



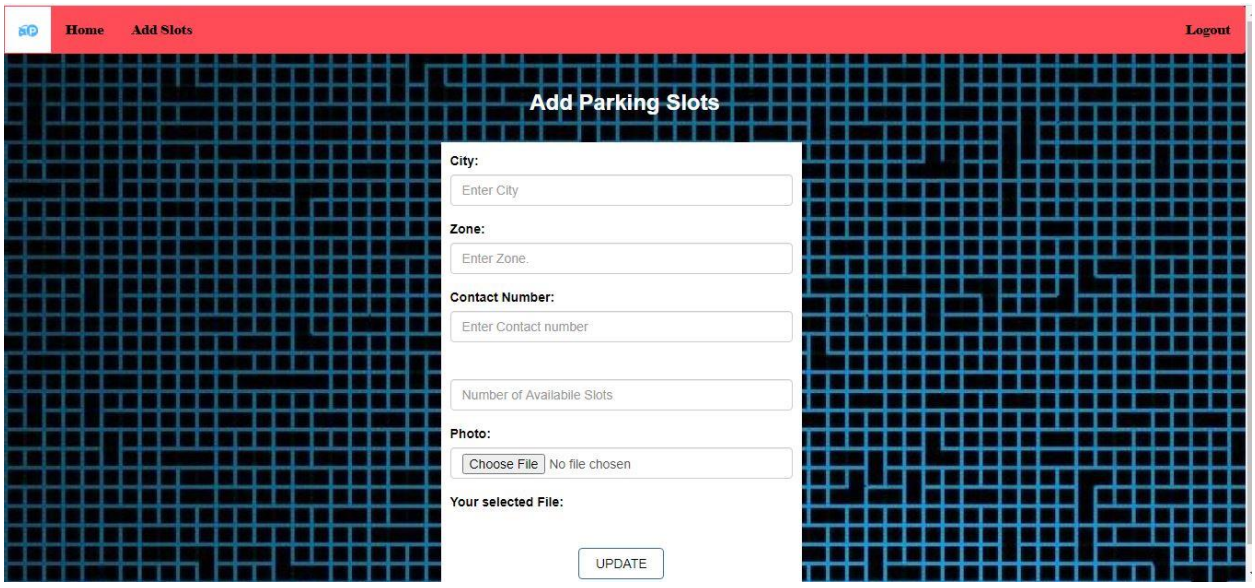
A.8: Booking Slot



A.9: Payment

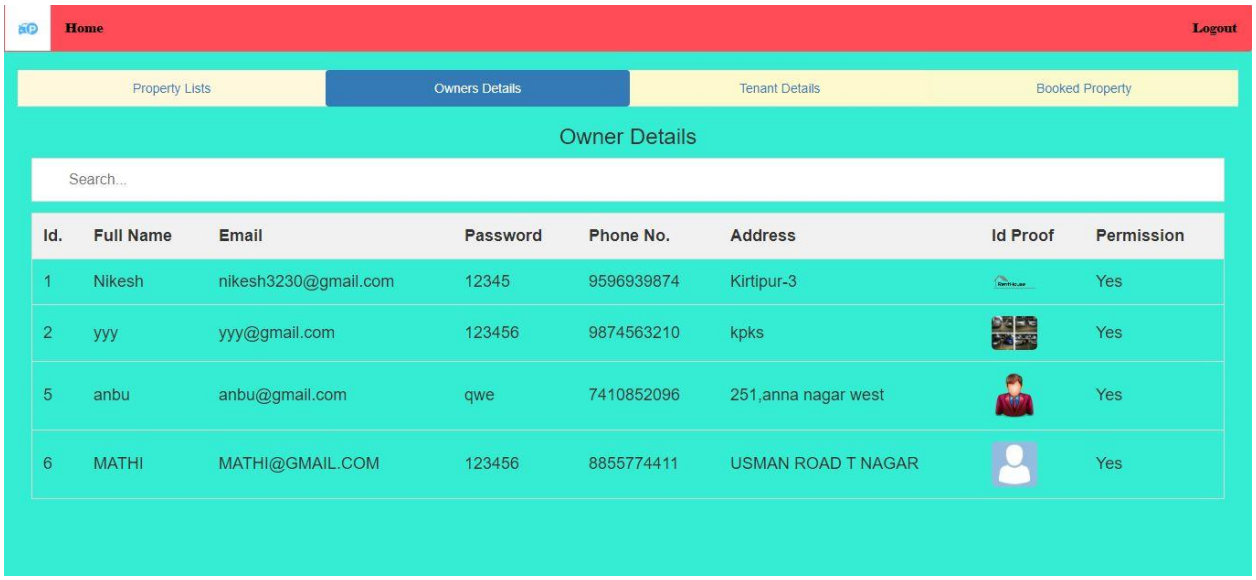


A.10 : Owner Profile

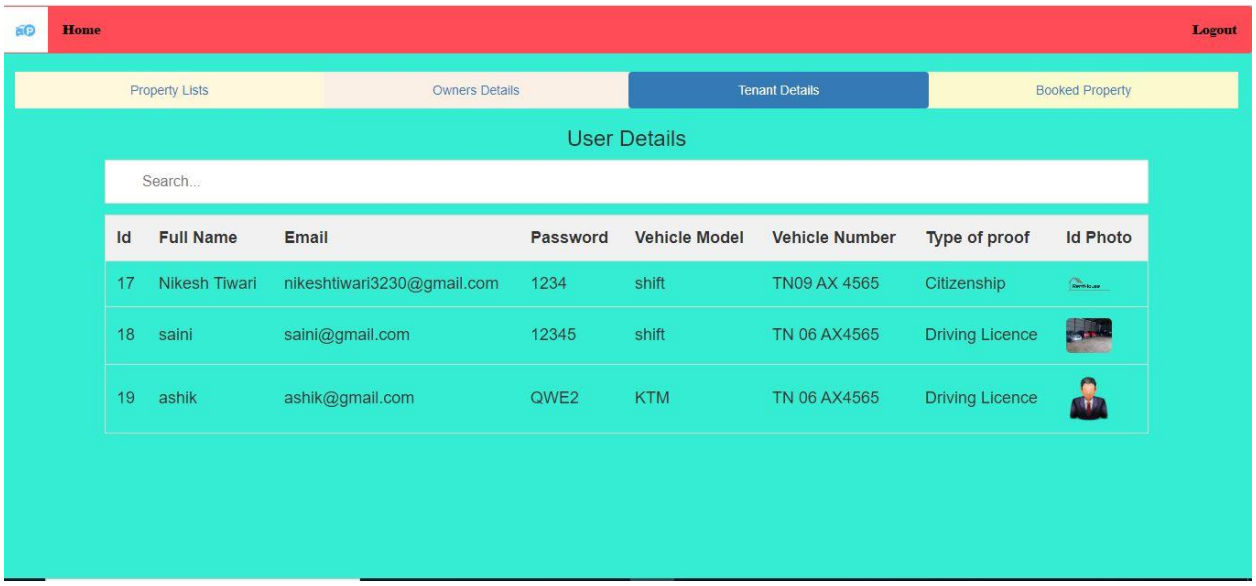


A.11: Add Slot

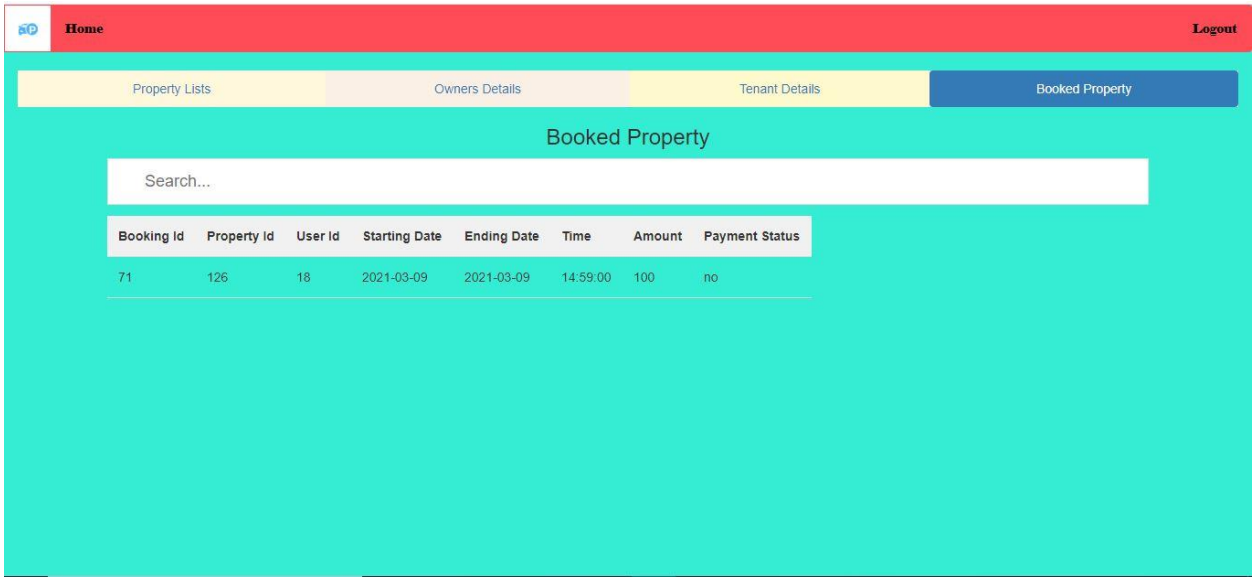
A.12: Admin Index



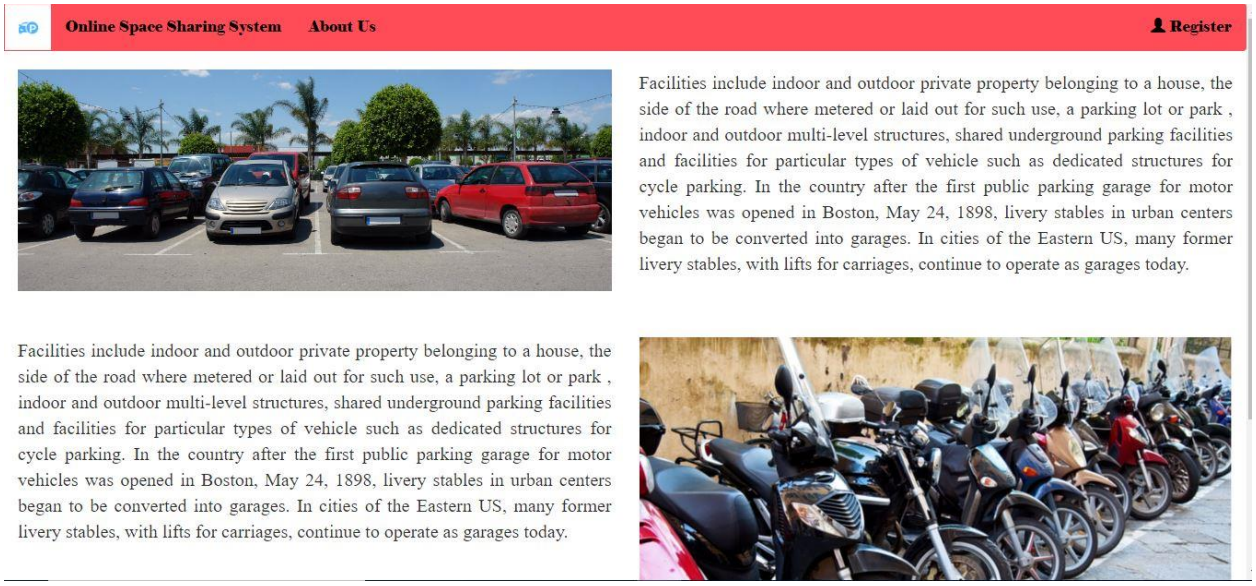
A.13: Owner Details



A.14: User Details



A.15: Booking Details



A.16: About Page

REFERENCES:

[1] R. Carli, M. Dotoli, and R. Pellegrino, “A hierarchical decision-making strategy for the energy management of smart cities,” IEEE Trans. Autom. Sci. Eng., vol. 14, no. 2, pp. 505–523, Apr. 2017.

[2] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, “User-defined privacy grid system for continuous location-based services,” IEEE Trans. Mobile Comput., vol. 14, no. 10, pp. 2158–2172, Oct. 2015.

[3] Finding Out All the Parking Spaces in Beijing. Accessed: Sep. 16, 2017. [Online]. Available: https://www.sohu.com/a/192479755_683886

[4] S. M. Nir, “Car-share companies get coveted parking in New York City,” New York Times. Accessed: May 31, 2018. [Online]. Available: https://www.nytimes.com/2018/05/31/nyregion/nyc-zipcar-parking.html

[5] X. T. R. Kong, S. X. Xu, M. Cheng, and G. Q. Huang, “IoT-enabled parking space sharing and allocation mechanisms,” IEEE Trans. Autom. Sci. Eng., vol. 15, no. 4, pp. 1654–1664, Oct. 2018.

[6] O. T. T. Kim, N. H. Tran, C. Pham, T. LeAnh, M. T. Thai, and C. S. Hong, “Parking assignment: Minimizing parking expenses and balancing parking demand among multiple parking lots,” IEEE Trans. Autom. Sci. Eng., vol. 17, no. 3, pp. 1320–1331, Jul. 2020.

[7] C. M. A. for Planning. Parking Strategies to Support Livable Communities. Accessed: May 2012. [Online]. Available: https://www.slideshare.net/lindsaybanks/parking-strategies-to-supportlivable-communities-12836141.

[8] R. Carli, M. Dotoli, and R. Pellegrino, ―A hierarchical decision-making strategy for the energy

management of smart cities,‖ IEEE Trans. Autom. Sci. Eng., vol. 14, no. 2, pp. 505–523,

Apr.2017.

[9] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, ―User-defined privacy grid system for

continuous location-based services,‖ IEEE Trans. Mobile Comput., vol. 14, no. 10, pp. 2158–

2172, Oct. 2015.

[10] Finding Out All the Parking Spaces in Beijing. Accessed: Sep. 16, 2017. [Online]. Available:

https://www.sohu.com/a/192479755 6 83886

[11] S. M. Nir, ―Car-share companies get coveted parking in New York City,‖ New York Times.

Accessed: May 31, 2018. [Online]. Available:

https://www.nytimes.com/2018/05/31/nyregion/nyc- zipcar-parking.html

[12] X. T. R. Kong, S. X. Xu, M. Cheng, and G. Q. Huang, ―IoT-enabled parking space sharing and

allocation mechanisms,‖ IEEE Trans. Autom. Sci. Eng., vol. 15, no. 4, pp. 1654–1664, Oct.

2018.

[13] O. T. T. Kim, N. H. Tran, C. Pham, T. LeAnh, M. T. Thai, and C. S. Hong, ―Parking assignment:

Minimizing parking expenses and balancing parking demand among multiple parking lots,‖

IEEE Trans. Autom. Sci. Eng., vol. 17, no. 3, pp. 1320–1331, Jul. 2020.

[14] C. M. A. for Planning. Parking Strategies to Support Livable Communities. Accessed: May 2012.

[Online].

[15] Lin, T.; Rivano, H.; Le Mouel, F. A Survey of Smart Parking Solutions. IEEE Trans. Intell.

Transp. Syst. 2017, 18, 3229–3253. [CrossRef]

[16] Vancluysen, K.; Boras, K. Smart Parking in the Thinking City; Technical Report; 2016.

Available online:

http://content.yudu.com/Library/A3zuy2/SmartParkingInTheThi/resources/index.htm

(accessed on 5 October 2019). 15. Geng, Y.; Cassandras, C