A Project Report on

ParkMania : The Parking Management System

Submitted in partial fulfillment of the requirements for the award
of the degree of

Bachelor of Engineering

in

Information Technology

by

Srinivas Vishwanath(16104010)
Saurabh Sharma(16104001)

Under the Guidance of

Prof. Yaminee Patil
Prof. Sneha Kanchan

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2019-2020

Approval Sheet

This Project Report entitled ***"ParkMania : The Parking Management System"*** Submitted by ***"Srinivas Vishwanath" (16104010), "Saurabh Sharma"(16104001)*** is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Sneha Kanchan Co-Guide	Prof. Yaminee Patil Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"ParkMania : The Parking Management System"*** submitted by ***"Srinivas Vishwanath" (16104010),"Saurabh Sharma" (16104001)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology*** to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020.

Prof. Sneha Kanchan Co-Guide	Prof. Yaminee Patil Guide
Prof. Kiran Deshpande Head Department of Information Technology	Dr. Uttam D.Kolekar Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Srinivas Vishwanath 16104010)

(Signature)

(Saurabh Sharma 16104001)

Date:

Abstract

Due to the increasing population in urban cities, there is an exponential rise in the number of vehicles which is leading to major problems leading to poor traffic management and congestion. Another major problem faced by the vehicle owners is the availability of parking space. The idea of Smart Cities is slowly gaining pace with the ever increasing technologies. Therefore, in the proposed parking system we are integrating the Wireless Sensor Technology with the Android Application so that the user can book or pre- book a slot. The vehicle owner will be able to reserve a slot for his/her vehicle from anywhere and will be provided with a QR code which will be scanned on the entry of the parking area. Another feature our system provides is providing information about the near-by parking areas which comes handy when the current parking area is full.

Contents

List of Figures

List of Tables

List of Abbreviations

IR :	Infrared Sensor
WSN :	Wireless Sensor Network
QR :	Quick Response
OTP :	One Time Password

Chapter 1

Introduction

In this system we focus to eradicate the problem of improper vehicle parking and unavailability of parking area. The basic technology our system uses is the WSN (Wireless Sensor Network) to detect if a vehicle is present in a specified slot. Wireless sensor networks (WSNs) organize and collect the data at a central location. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind, and so on. In our system, we will be making use of the IR sensors [1]. An infrared sensor is used to sense objects within its vicinity. IR sensors have the ability of measuring heat emissions and sensing motion. Infrared sensors are also capable of measuring the heat being emitted by an object and detecting motion. IRsensors prove essential in detecting an obstacle within its range. Therefore, in our system they are used to detect if the vehicle is present or not in a particular slot. Further, the vehicle will be guided to the parking slot with the help of the application. The details of the vehicle owners and that of the vehicles are stored in the backend database and thereby the identity is verified right at the entrance. Therefore, our system provides security. Payment will be calculated on the basis of time. Our system provides flexibility of booking a parking slot through the application so they will not have to wander in search of a vacant slot in the parking area. Assistance will be provided to guide the vehicle owner to the empty parking slot thereby avoiding chaos. And the real time updates will help the user know which slot is available for parking the vehicle.

Chapter 2

Literature Review

[1] D.Vakula and Yeshwanth Krishna Kolli, “Low Cost Smart Parking System for Smart Cities”, Department of Electronics and Communication Engineering National Institute of Technology, Warangal Telangana, India.

In the year 2017, D. Vakula and team proposed a vehicle parking system which was developed using the Internet of Things (IoT). The major technology used here is the Wireless Sensor Network Technology (WSNT). From this system we found out how ultrasonic sensors can be integrated with Raspberry Pi and Node MCU and how the information from Raspberry Pi can be transmitted to the front end GUI and back end database [1]. The information is fetched from the ultrasonic sensors that detect the presence or absence of a vehicle. This information is forwarded to the Node MCU. A single Node MCU takes care of 4-5 parking slots respectively. The Node MCU transmits this information to the Raspberry Pi that has an on- board Wi-Fi module which helps for internet connectivity. Hence the information of whether a parking slot is available or not is forwarded to the Front End (application/website) and the user/ administrator gets to know about the availability of the parking slot. Additionally, the vehicle entries are stored in the backend database hence keeping a record of the vehicles that have entered in the parking area, so that the details can be quickly and easily accessed. Database helps maintain all the vehicle and user records for future reference.

Advantages:

- It provides GUI so that you can see the available slots.

Disadvantages:

- No solution if parking slots are full. Does not display nearby parking area within the vicinity of the original parking area.

[2] Chaudhary, H., Bansal, P., Valarmathi, B. “Advanced CAR parking system using Arduino”. 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS).

In 2017, Hemant Chaudhary, Prateek Bansal, Dr.B. Valarmathi came up with a Vehicle Parking System which is developed using IR sensors [2]. An infrared sensor is an electronic instrument that is used to sense certain characteristics of its surroundings. It does this by either emitting or detecting infrared radiation. Infrared sensors are also capable of measuring the heat being emitted by an object and detecting motion. IR sensors prove essential in detecting an obstacle within its range. The major concern of their system was security. Therefore, they made use of RFID to authenticate a customer before he/she enters the parking area. Also, their system consisted of a gate at the entrance which would only open when the RFID scan was successful. For displaying if a slot is vacant or not, a LCD is used to display this information. If car is allowed to park, then mobile notification will be sent to the user about parking. It solves the parking issue in urban areas, also provides security to a vehicle and an unauthorized user is not allowed to enter into a parking place. It helps to park vehicle in multi-floored parking also as it will display which floor has free space.

Advantages:

- Every user gets an authorization RFID card which is verified and thereby only authentic car owners are allowed and only then will the gate open. Security is thereby taken care of.

Disadvantages:

- No GUI provided to see the available slots. Since Arduino was used external Wi-Fi module was required to transmit data to application.

[3] Mendiratta, S., Dey, D., Rani Sona, D. (2017). "Automatic car parking system with visual indicator along with IoT". 2017 International Conference on Microelectronic Devices, Circuits and Systems.

Sarthak Mendiratta, Debopam Dey, Deepika Rani Sona developed an Automatic car parking system with visual indicator along with IoT and published a paper which focuses on the concept of car parking detection mechanism using the ultrasonic sensor, in combination with the usage of Internet of Things i.e. sending the status of the parking slot to the Internet [3] Through which the user at any place in the world can see which parking slot is empty and where to park. This is done by sending the data of ultrasonic sensor through our Wi-Fi module that is ESP8266 to any open source easy to use IOT platform that uses HTTP to display our data (thingspeak.com in this case). LEDs were used to display the availability status of a particular slot. If a particular slot is vacant, then green LED would glow and if the slot is occupied a red LED would glow. To switch between Red Bulb and the Green Bulb a relay module is used which is triggered by the 5 Volt pin of the Arduino Board and to glow the bulb, an AC power supply is used, which will be connected to the relay module. The two Ultrasonic Sensors are used to eliminate and minimize any manual or human interference thus increasing the efficiency of the overall system. Since they had used Arduino, so Wi-Fi connectivity was a major problem, so to eradicate this problem they had to use external Wi-Fi module i.e. ESP8266 for connectivity purpose.

Advantages:

- Does not provide a solution when the parking area is full.
- GUI is provided which helps the user to know beforehand which parking slot is empty.
- This system is cost effective.

Disadvantages:

- No online payment feature included. Nearby parking areas are not displayed in case the parking is full.

[4] Desai, J., Bhanje, A., Biradar, S., Fernandes, D "IoT based Vehicle Parking Manager." 2017 7th International Conference on Cloud Computing, Data Science, Engineering Confluence.

IoT based Vehicle Parking Manager was developed by the team of Desai, J., Bhanje, A., Biradar, S., Fernandes in the year 2017 [4]. The main motive of this project was to provide a solution for resolving the parking issues that exist in public places such as malls, multiplexes, etc. especially on weekends. The objective was to achieve this by using the concept of Internet of Things (IoT), in which they developed an Android Application which provides brief details. This application provides certain functionalities that include identifying each vehicle uniquely, showing availablity of slots through the mobile application, and booking reservations for the same, maintenance of a database (for the management). Functionalities of the project for a user are: Representation of total parking slots in the parking lot, count of vacant parking slots, occupied parking slots and the reserved parking slots. Assigning QR code to customers/cars for identifying them uniquely. A user can check the balance, also check the transaction 'records of the previous parking done by them in the parking place, make reservations along with the timings. These transactional records and the QR code are displayed on the receipt that is generated. Features of the proposed project for managing the parking system are: A QR Code reader at the entry of the parking system to identify the unique ID associated with the vehicle and do the corresponding transactions (entry time, rate deduction etc.), An LCD at the gate to display the car number and the parking slot selected by the user, Parking slots status (fully reserved, totaltime slots), Earnings - Total/Per Customer/per week, View database of all the cars which entered the parking system along with the timings and the slots occupied., Recharge the customer's ID at the entry if required and Special reservations for VIP's (no time limit) Allow one time entries for vehicle without any unique ID.

Advantages:

- Provides a feature to view the availability of the parking slots through an application time.
- QR code scanning for authorization.

Disadvantages:

- Does not provide a solution when the parking area is full.

[5] Khanna, A., Anand, R. (2016). IoT based smart parking system. 2016 International Conference on Internet of Things and Applications (IOTA).

Khanna, A., Anand, R. In the year 2016 came up with the IoT based smart parking system [5]. The paper they published, mainly focuses implementing a vehicle parking system using Wireless Sensor Network. The main sensors that were utilised in their system were the IR and the Ultrasonic Sensors to detect the presence or absence of a vehicle in a particular slot. Additionally, an application was provided to the users for interactive experience and for the ease of booking the parking slot from any location. The application showed which slot was currently vacant and could be booked while the ones which are occupied as well. Also, the user could be able to select the time for which the particular slot he/she was going to occupy (2 hours, 3 hours...5 hours.). The major highlight of this project we learnt was that the server was deployed on cloud. Therefore, all the storage of user and vehicle information was done on the cloud database. In case of a system failure, since cloud was used, it provided quick recovery thereby increasing the uptime and also provided the backup for the data. Time out notifications were sent to user through the mobile application thereby keeping them updated. The cost calculation was done on time basis. The timer used to start as soon as the vehicle occupied a particular slot. Once the driver had parked its car in the selected slot he/she needed to confirm its occupancy. This very scenario in which the driver had to specify its presence. This feature was added so that only a genuine driver can park its car in a particular parking slot. If a driver failed to confirm his occupancy in the next 30 secs of parking its car, an alarm would start ringing causing the authorities to know that a car has been parked in the wrong place

Advantages:

• Provides cloud for the storage of vehicle and vehicle owner details.

• High detail view of system infrastructure.

• An application is provided through which customers are notified.

Disadvantages:

• Does not provide a solution if the parking area is full.

[6] Owayjan, M., Sleem, B., Saad, E., Maroun, A. (2017). Parking management system using mobile application. 2017 Sensors Networks Smart and Emerging Technologies (SENSET).

Michel Owayjan, Bahaa Sleem, Elio Saad Amer Maroun in the year 2017 [6] proposed a parking management system using a mobile application to address the parking problems in malls. The hardware part of this system consists of custom- made sensor units based on a phototransistor with an infrared transmitter that is responsible for determining if a cert ain parking space is occupied or free. The units were connect to a central controller wirelessly using Arduino microcontrollers with Ethernet shields. The central controller was a server hosting a database that is accessible through the developed mobile application. The software part of this system consisted of the mobile application which was developed using the Eclipse IDE and runs on the Android platform. It was connected to the database using JSON (JavaScript Object Notation) format. Using this mobile application, users were able to locate free parking spaces, check the parking fees, locate their cars, and can even perform the payment transactions.

Advantages:

• Users can easily and quickly pre-book or book a slot through the GUI provided i.e. Android Application

Disadvantages:

• On failure of main controller, the entire system fails.

Chapter 3

Design

3.1 Activity Diagram:

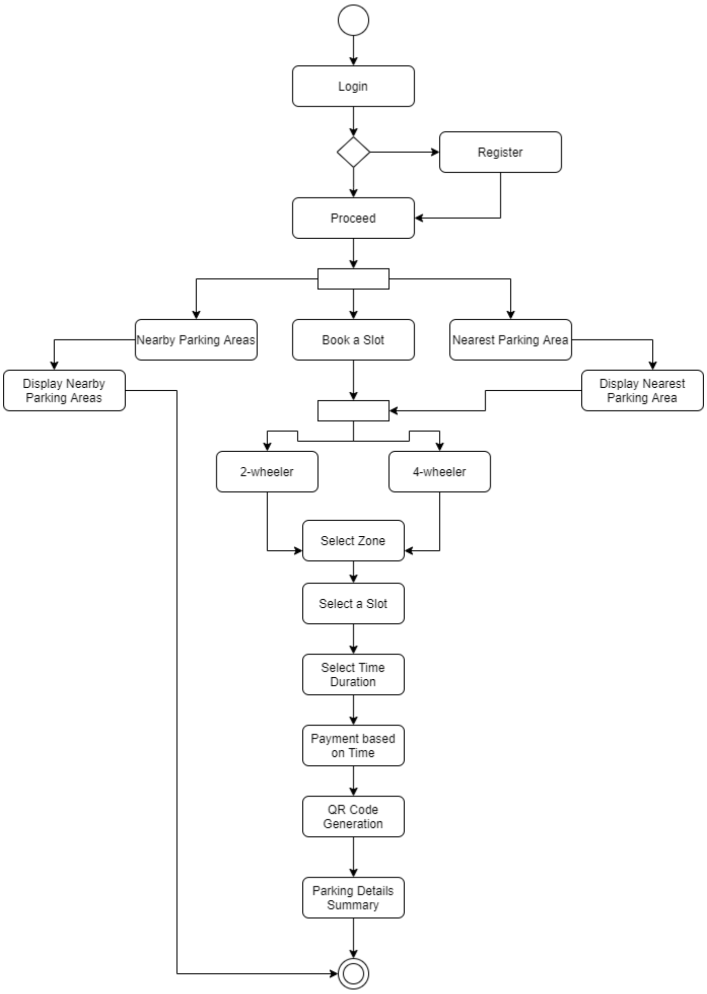


Figure 3.1: Activity Diagram

Activity Diagram begins with the login or signup page after which the user can either book a slot or check for nearby parking areas. Further there are zones for 2 wheeler and

4 wheeler parking which the user should select providing the necessary date and time for booking the slot. Accordingly the price will be calculated and displayed on to the user The user can generate a QR code which contains the booking details of that particular user and this QR code will be scanned at the entrance of the parking area.

3.2 Usecase Diagram:

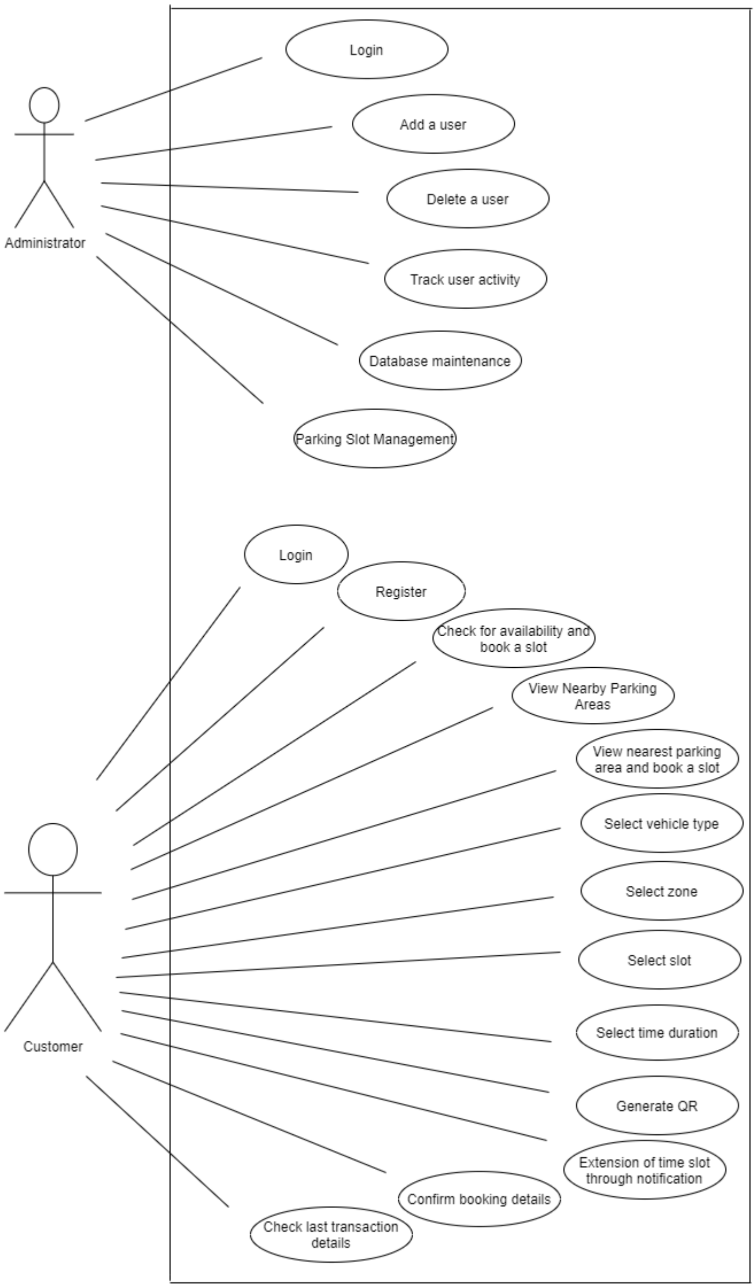


Figure 3.2: Usecase Diagram

Our use case consists of two major entities that interact with the major modules of the application. The first one is the customer who can sign up/login, book a slot check for nearby areas, book a slot in nearby parking area. The second is the administrator who has an access to access the backend database enter and manage slots, accounts and user activity.

3.3 Class Diagram:

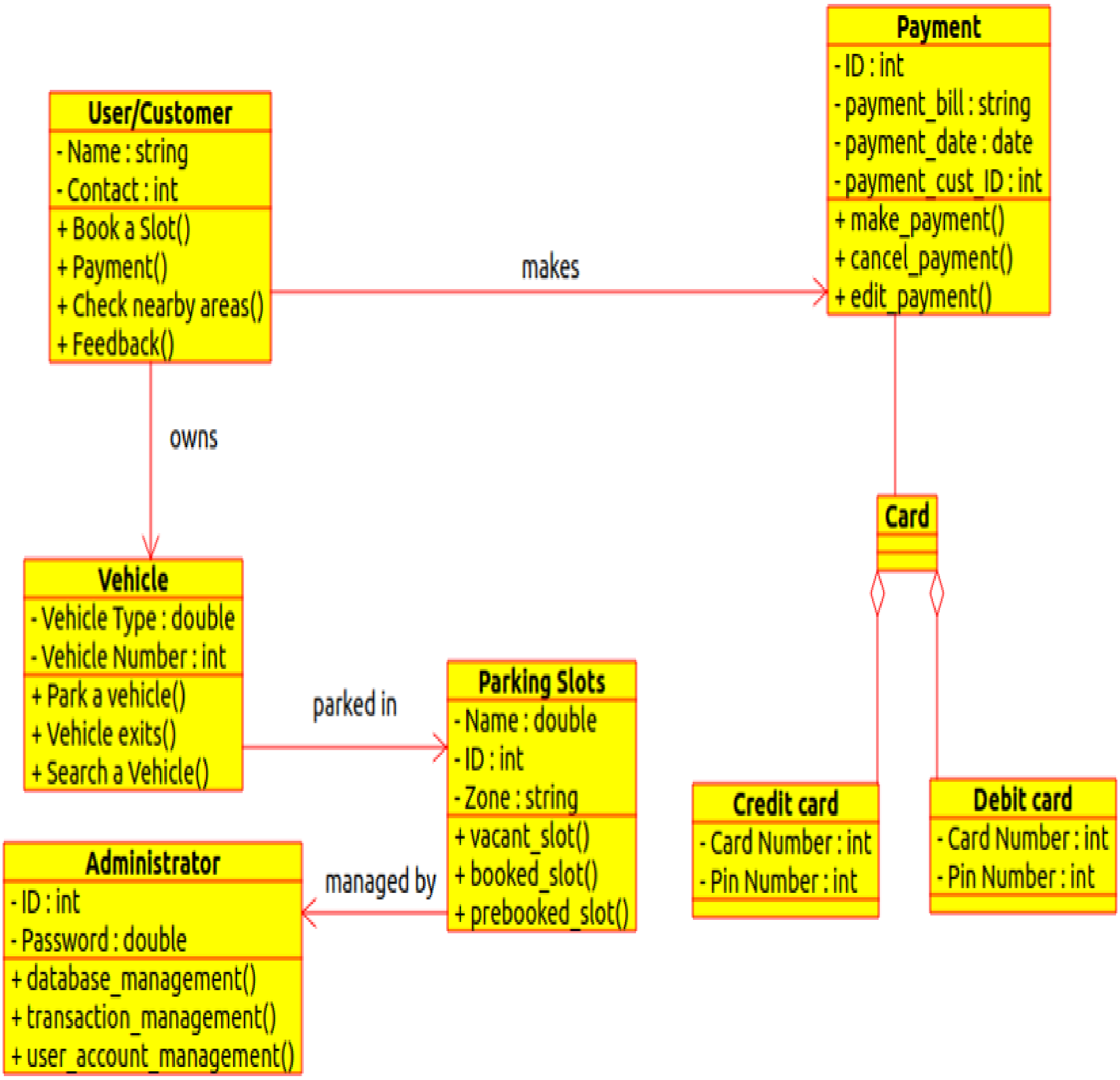


Figure 3.3: Class Diagram

The class diagram alongside displays various modules of our application interacting with each other. As it can be seen, the class customer is associated to payment class as customer makes the payment. Similarly, admin monitors and manages the parking slots, the vehicle is parked in these slots. Payment is further divided into Credit card and Debit card payment.

3.4 System Architecture:

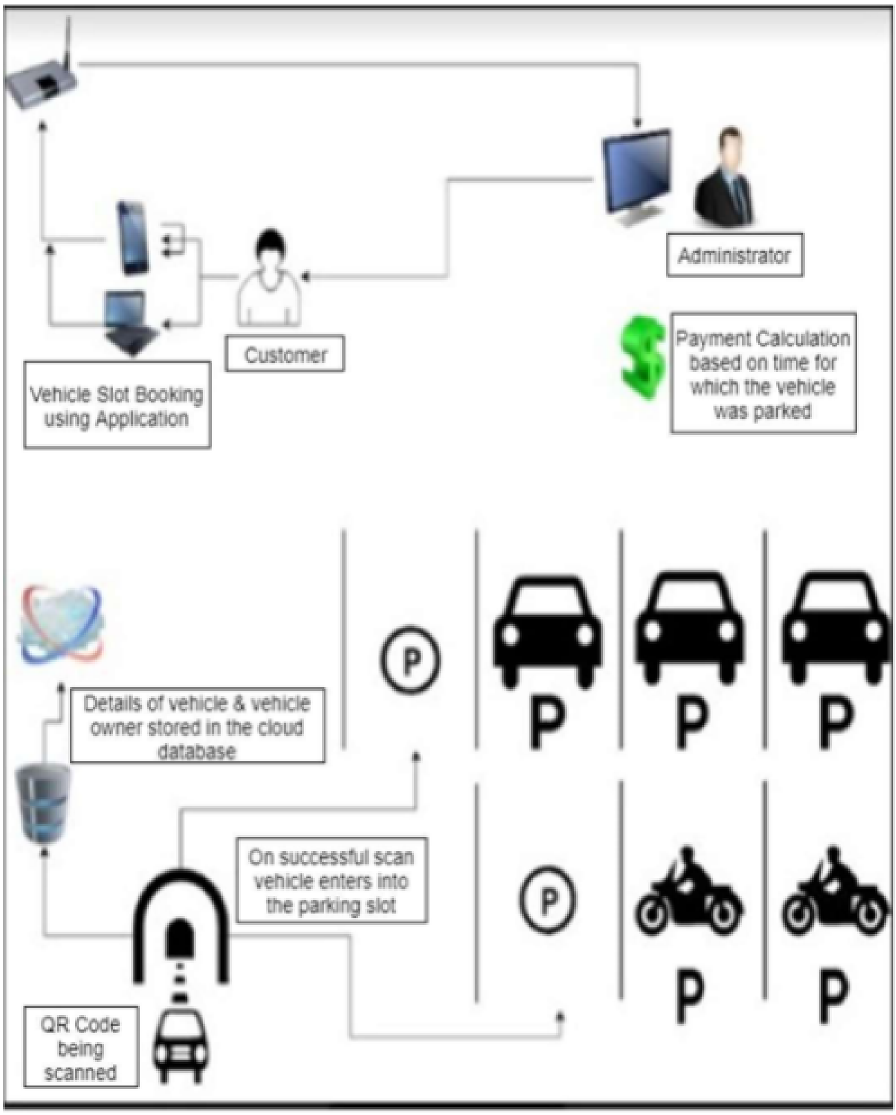

Administrator
Customer
Vehicle Slot Booking using Application
Payment Calculation based on time for which the vehicle was parked
Details of vehicle & vehicle owner stored in the cloud database
On successful scan vehicle enters into the parking slot
QR Code being scanned
P
P
P
P
P
P
P

Figure 3.4: System Architecture

The proposed system architecture depicts the actual working of the system in real time in collaboration with the WSNs and how data gets stored on backend database and cloud.

Chapter 4

Implementation

Our proposed system works in the following way:

1. Vehicle Owner books/reserves the parking slot using the mobile application by selecting the date and time.

2. In turn the user will get a QR code which will be used for verification later.

3. Our system provides multi-factor authentication which includes OTP and Email authentication.

4. At the entrance, the user verifies the identity using the QR code and is assigned a particular parking slot based on the type of vehicle (car/bike).

5. The details are stored at the back end when the QR code is scanned.

6. An additional feature our system provides is that it displays the nearby parking areas. This feature is included because if the current parking area is full, the user can check for the nearby area where the vehicle can be parked.

7. The application contains different zones for parking two wheeler and four wheeler vehicles.

8. The application displays booked, prebooked and occupied slots with the help of GUI.

9. Additional amount is imposed if a user extends the parking time.

10. A notification for extension of slot booking is provided in case user wishes to extend the time slot.

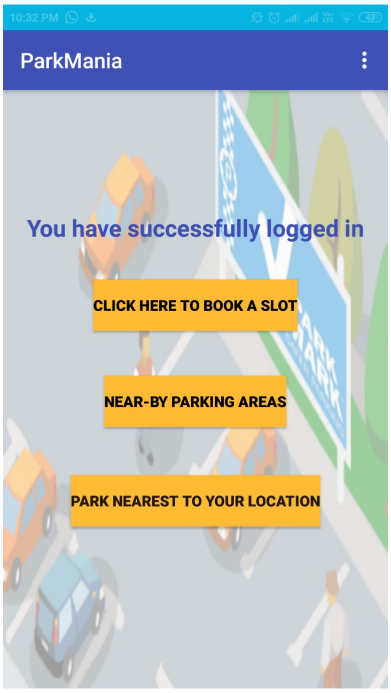
11. If the user does not have an application or internet connectivity, then booking for that particular user will be done right at entrance of the parking area by providing a receipt or ticket.

Therefore additional features our system is providing is:

- Prebooking a slot based on time.
- Real time updates of slot occupancy.
- Information about nearby vacant parking slots within the vicinity of the main parking slot.
- Booking the slots in the nearby parking area.
- Security, in terms of stored data and authentication means.

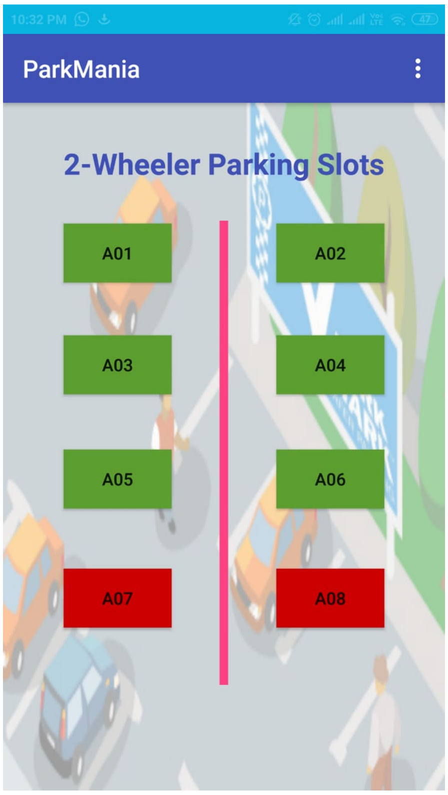
Following are some of the snapshots of our implemented system:

1.Home Page:



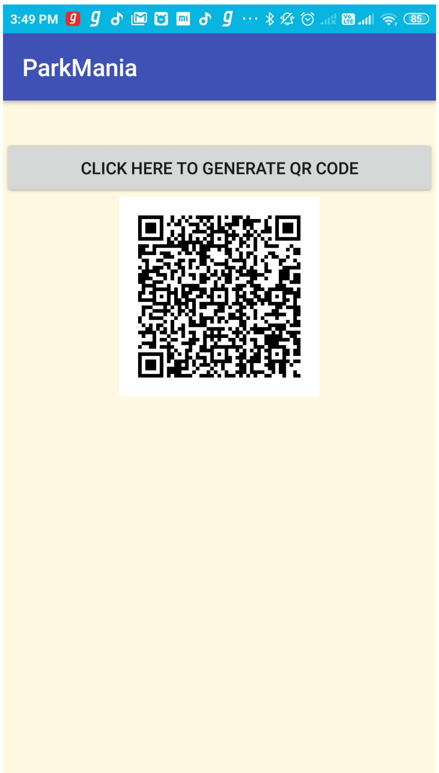
You have successfully logged in

2.Slots Layout:



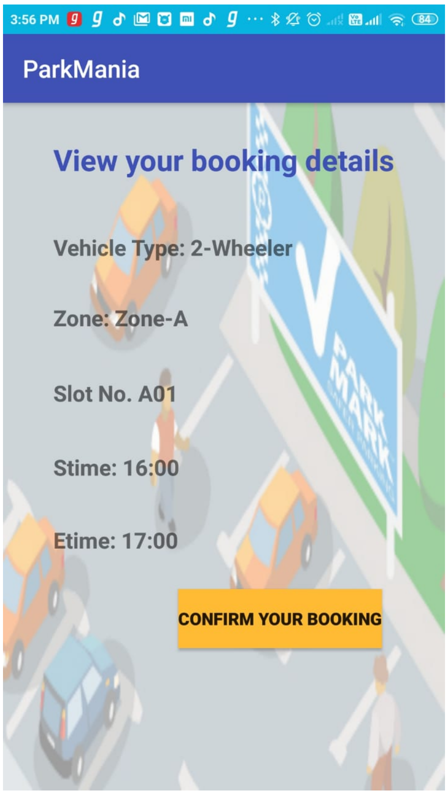
2-Wheeler Parking Slots

3.QR Code Generation:


3:49 PM

ParkMania

4.Parking Details:


3:56 PM

ParkMania

View your booking details

Vehicle Type: 2-Wheeler

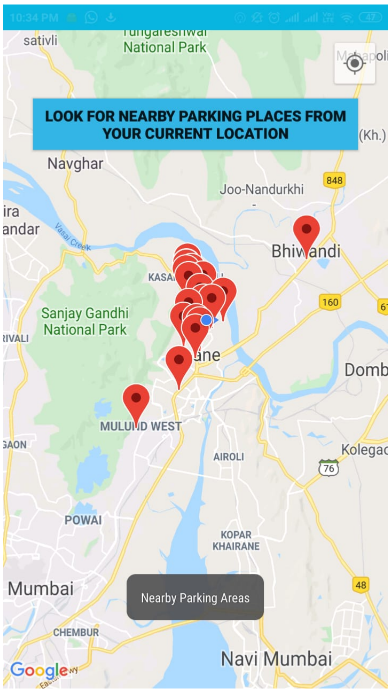
Zone: Zone-A

Slot No. A01

Stime: 16:00

Etime: 17:00

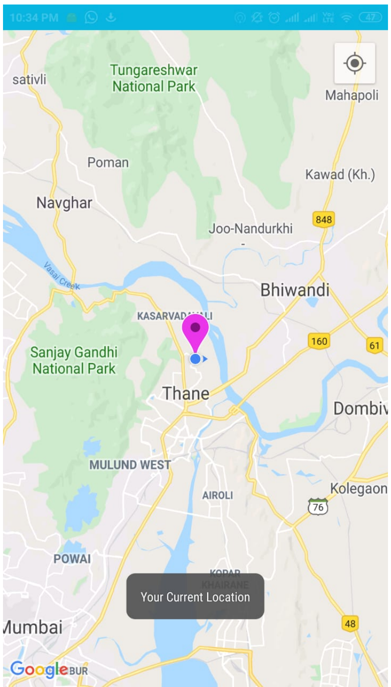
5.Nearby Parking Areas:



LOOK FOR NEARBY PARKING PLACES FROM YOUR CURRENT LOCATION

Nearby Parking Areas

6.Nearest Parking Areas:



Your Current Location

Chapter 5

Testing

For the testing purpose we opted to go for the functional testing methods. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. Function testing includes :

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing also makes debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.

Therefore, opting for the unit testing method in our project played a crucial role in assessing each module of the application separately. This testing method best suited our project as we had various modules at the start which were to be tested and verified. It made the testing process easier by helping us discover the minute errors in each module and therefore we could rectify them efficiently.

Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

Integration testing was a necessity to check whether each individual module/unit was working well in synchronisation with one another. There were multiple problems while integrating the various modules which were only discovered with the help of integration testing methodology. For example, we integrated the Google Maps API for the Nearby and Nearest Parking

Area feature, the QR Code generation using the Zxing library and the Cloud Database i.e. Firebase. The OTP and Email Authentication of Firebase were also used.

System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

Finally, the entire system was tested as a whole using the System testing Methodology. Here, the functional requirements of our applications that include booking slots, checking Nearby Parking Areas and checking the Nearest Parking Area were checked altogether. The applications was given to other professionals (faculty) who checked and verified the proper working of all the modules and the system as a whole.

Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs. This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users.

Acceptance Testing method was adopted for our application to get a final review of our application by the actual end users. The application was given to fellow peers to try hands on and get their opinions and recommendations which were worked upon. Also, it was seen that all the end users needs were met for their complete satisfaction after using the application. By this, an idea as to if our application is actually acceptable by the users was gained and the view obtained from them were considered and worked upon.

Chapter 6

Result

Our system provides an upper hand in the following ways as shown in (Table 5.1):

1. Gives information about the nearby parking areas which helps users in case the main parking area is full.

2. The availability of GUI makes our system more user friendly and interactive and makes the overall process easier.

3. Dedicated application which helps to book slots from any time anywhere.

4. User can check the availability of slots in real- time.

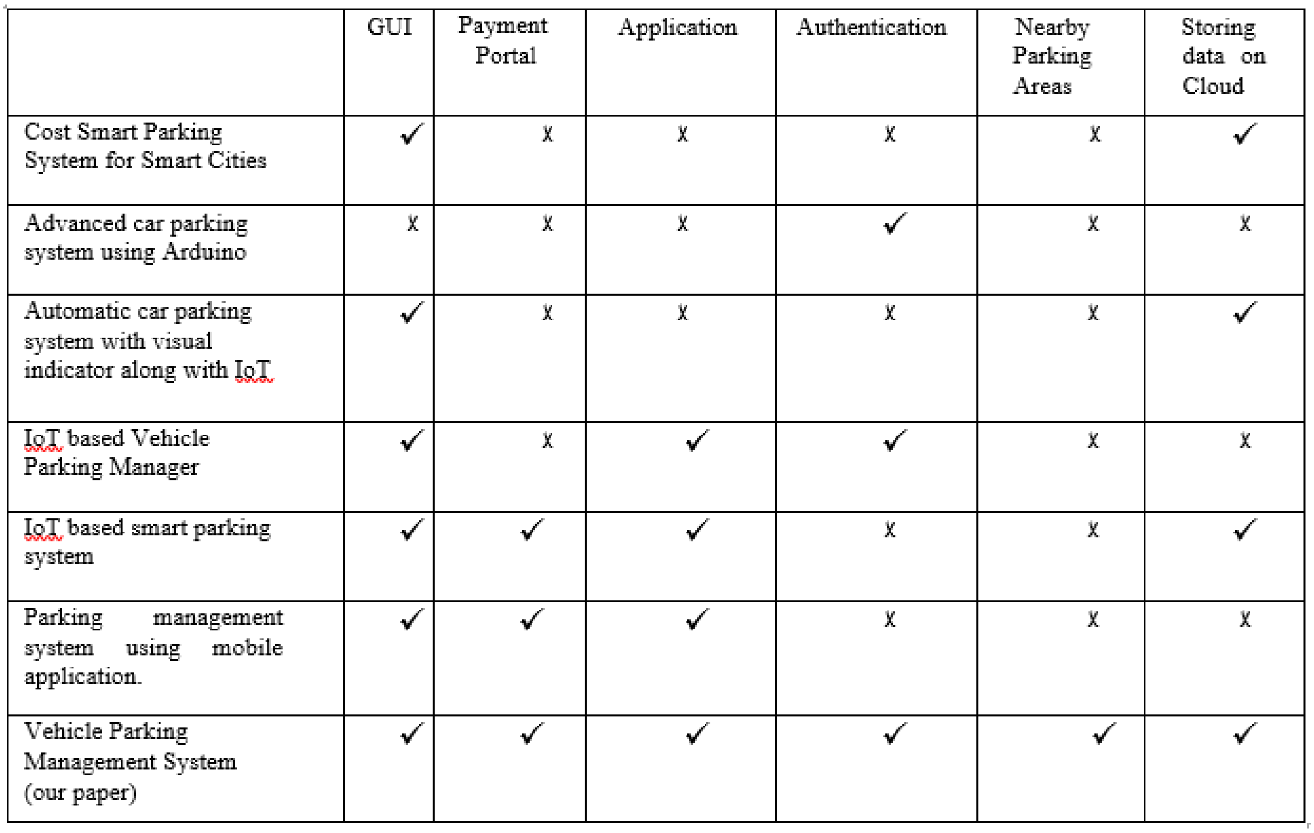
5. Pricing will be based on the time duration and it will be dynamic as per the number of hours.

6. Online payment option is available based on time for which the vehicle was parked.

7. Real Time updates for displaying availability of slots.

8. QR (Quick Response) code on the E-bill for user authentication.

9. Feedback mechanism for gaining user suggestions.



	GUI	Payment Portal	Application	Authentication	Nearby Parking Areas	Storing data on Cloud
Cost Smart Parking System for Smart Cities	✓	X	X	X	X	✓
Advanced car parking system using Arduino	X	X	X	✓	X	X
Automatic car parking system with visual indicator along with IoT	✓	X	X	X	X	✓
IoT based Vehicle Parking Manager	✓	X	✓	✓	X	X
IoT based smart parking system	✓	✓	✓	X	X	✓
Parking management system using mobile application.	✓	✓	✓	X	X	X
Vehicle Parking Management System (our paper)	✓	✓	✓	✓	✓	✓

Table 6.1: Comparison with existing systems

Chapter 7

Conclusions and Future Scope

7.1 Conclusion

By implementing this system, we are making the work of finding parking slots for private vehicles much more easier. The interface provided by the system would be user friendly and interactive which will attract more customer base. This thereby will solve the problem of traffic jams causing inconvenience to the people on road. Finding parking slots will no more be a difficult task for the vehicle owners. Availability of these parking areas will be known before hand. A proper bill and amount calculation based on the time for which the vehicle was parked is provided to the customers using this system.

7.2 Scope

- Our application can be scaled up for further use by residential societies.
- Although this project can be scaled up to bigger parking areas for eg. Malls, Airports, etc.

Bibliography

[1] D.Vakula and Yeshwanth Krishna Kolli, “Low Cost Smart Parking System for Smart Cities”, Department of Electronics and Communication Engineering National Institute of Technology, Warangal Telangana, India.

[2] Chaudhary, H., Bansal, P., Valarmathi, B. “Advanced CAR parking system using Arduino”. 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS).

[3] Mendiratta, S., Dey, D., Rani Sona, D. (2017). “Automatic car parking system with visual indicator along with IoT”. 2017 International Conference on Microelectronic Devices, Circuits and Systems.

[4] Desai, J., Bhanje, A., Biradar, S., Fernandes, D “IoT based Vehicle Parking Manager.” 2017 7th International Conference on Cloud Computing, Data Science, Engineering Confluence.

[5] Khanna, A., Anand, R. (2016). IoT based smart parking system. 2016 International Conference on Internet of Things and Applications (IOTA).

[6] Owayjan, M., Sleem, B., Saad, E., Maroun, A. (2017). Parking management system using mobile application. 2017 Sensors Networks Smart and Emerging Technologies (SENSET).

[7] Devi, R. S. S., Kumar, V. R. V., Sridevi, S. (2017). Application development for reservation based parking slot allotment and management system using Android. 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS).

[8] Praveen, M., Harini, V. (2019). NB-IOT based smart car parking system. 2019 International Conference on Smart Structures and Systems (ICSSS).

[9] Lee, C., Han, Y., Jeon, S., Seo, D., Jung, I. (2016). Smart parking system for Internet of Things. 2016 IEEE International Conference on Consumer Electronics (ICCE).

[10] Hakim, I. M., Christover, D., Jaya Marindra, A.M. (2019). Implementation of an Image Processing based Smart Parking System using Haar-Cascade Method. 2019 IEEE 9th Symposium on Computer Applications Industrial Electronics (ISCAIE)

Appendices

Appendix-A: Installation and Configuration of Android Studio

1. If you downloaded an .exe file (recommended), double-click to launch it.

If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio ¿ bin folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines)
2. Follow the setup wizard in Android Studio and install any SDK packages that it recommends.

Android Studio provides wizards and templates that verify your system requirements, such as the Java Development Kit (JDK) and available RAM, and configure default settings, such as an optimized default Android Virtual Device (AVD) emulation and updated system images. This document describes additional configuration settings you may want to use to customize your use of Android Studio.

Android Studio provides access to two configuration files through the Help menu:

studio.vmoptions: Customize options for Studio's Java Virtual Machine (JVM), such as heap size and cache size. Note that on Linux machines this file may be named studio64.vmoptions, depending on your version of Android Studio. idea.properties: Customize Android Studio properties, such as the plugins folder path or maximum supported file size.

For specific documentation about emulator and device setup and use, see the following topics:

Managing virtual devices

Using hardware devices

OEM USB drivers

This category encompasses those Java-based libraries that are specific to Android development.

Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows

- **android.app** Provides access to the application model and is the cornerstone of all Android applications.

- **android.content** Facilitates content access, publishing and messaging between applications and application components.
- **android.database** Used to access data published by content providers and includes SQLite database management classes.
- **android.opengl** A Java interface to the OpenGL ES 3D graphics rendering API.
- **android.os** Provides applications with access to standard operating system services including messages, system services and inter-process communication.
- **android.text** Used to render and manipulate text on a device display.
- **android.view** The fundamental building blocks of application user interfaces.
- **android.widget** A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- **android.webkit** A set of classes intended to allow web-browsing capabilities to be built into applications.

If you expand all of the folders in the project explorer you will see a vast array of files and folders. Most of them are managed by Android Studio.

1. Manifests folder

This is where you would put your manifest files. Most Android apps have single manifest file. But an app may have several manifest files due to application versioning, or for supporting specific hardware.

2. Java folder

Let's take a look at what is in the java folder. This is the folder in your project where you will be storing all of the source code files written in Java programming language.

3. Res folder

It contains folders that help you separate and sort the resources of your application. Resources basically mean all the needed files except the source code. For example, while developing an app, you need to include resource files such as the app-logo, photos, sounds, videos or animations. Each file type should be added to its own folder to comply with the Android development standards.

Acknowledgement

We have great pleasure in presenting the report on **ParkMania : The Parking Management System**. We take this opportunity to express our sincere thanks towards our guide **Prof. Yaminee Patil** & Co-Guide **Prof. Sneha Kanchan** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1: Srinivas Vishwanath
Student ID1: 16104010

Student Name2: Saurabh Sharma
Student ID2: 16104001

Publication

Paper entitled **“Vehicle Parking Management System”** is presented at **“2020 IEEE International Conference on Convergence to Digital World – Quo Vadis (ICCDW 2020) ”** by **“Saurabh Sharma and Srinivas Vishwanath”.**