**STUDENT FEE PAYMENT AND EMPLOYEE PAYROLL MANAGEMENT SYSTEM PROJECT**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

HARSHITH – 1602-19-737-073

VIVEK RAJ – 1602-19-737-122



**Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31 202**

**Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University)**

**2020**

**Hyderabad-500 031 Department of Information Technology**

**DECLARATION BY THE CANDIDATE**

We, HARSHITH and VIVEK RAJ bearing hall ticket numbers, 1602-19-737-073 and 1602-19-737-122 respectively, hereby declare that the project report entitled “VEHICLE PARKING MANAGEMENT SYSTEM” is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

HARSHITH – 1602-19-737-073

VIVEK RAJ – 1602-19-737-122

(Faculty In-Charge) (Head, Dept. of IT)

**AKNOWLEDGEMENTS**

Our Mini Project would not have been successful without the help of several people. We are extremely thankful to our college, Vasavi College of Engineering, for providing the opportunity to implement our project, “VEHICLE PARKING MANAGEMENT SYSTEM”.

We would like to express our gratitude to Ms. Prasanna, Assistant Professor, Department of Information Technology, Vasavi College of Engineering, for their esteemed guidance, moral support and invaluable advice provided by them for the success of the Mini Project.

Sincerely,

**HARSHITH- 1602-19-737-073**

**VIVEK RAJ- 1602-19-737-122**

**ABSTRACT**

As our parking system is quite improper, we selected this project in order to improve this system, to avoid all manual errors and to save time for the people.

Our project parking management system includes two types of users, new user and regular user. There will be unique features like parking cash deposit, less parking charges benefits for regular user. This project consists of type of vehicle, slot availability, generating ticket, generating price, login, arrival and departure of the vehicle.

**TABLE OF CONTENTS**

1. Introduction………………...............................................7
2. Technology………………………………………………9
3. Proposed Work…………………………………………10
4. Design………………………………………………..10
5. Implementation…………………………………….12
6. Testing……………………………………………..16
7. Results…………………………………………………20
8. Additional Knowledge Acquired………………………34
9. Conclusion and Future Work………………………….36
10. References…………………………………………….37

# INTRODUCTION

## ABOUT THE PROJECT

“VEHICLE PARKING MANAGEMENT SYSTEM” is a console-based C Project which

keeps the record of vehicles and users info. This project consists of type of vehicle, slot

availability, generating ticket, generating price, login, arrival and departure of the

vehicle.

## 1.1 PROJECT DOMAIN

A parking lot (or) a parking space is delegated for parking of vehicles. It mainly saves cost, time, energy and manual errors. Today parking lot can give us upgraded security, safety and privacy. Overall, it requires lower maintenance and services and ensures efficient functioning and coordination within the provided parking space. Our parking management allows both the new users and regular users to park a vehicle.

**1.2 SPECIFIC TO OUR PROJECT**

**REGULAR USER:**

1. Login using User name and Password.
2. The following are the facilities provided for the regular user :
3. parking cash deposit
4. Type of vehicle
5. Slot availability
6. Generating ticket
7. Arrival of the vehicle
8. Departure of the vehicle

**NEW USER:**

The following are the facilities provided for the new user :

1. Type of vehicle
2. Slot availability
3. Generating ticket
4. Arrival of the vehicle
5. Departure of the vehicle

**Regular user has the following benefits**

1. They can deposit parking cash
2. Payment can be done through parking cash
3. The regular user will be benifitted with this parking management system as they can pay through their parking cash with less parking charges when compared to the new user.

Vehicle Parking Management updates and maintains the records of the customers of the parking lot.

# TECHNOLOGY

All computer software needs certain hardware components or other software resources to be present, in order for computers to be used efficiently. These prerequisites are known as System Requirements. Within this, we have two types – Software Requirements and Hardware Requirements.

## SOFTWARE REQUIREMENTS

Software Requirements deal with defining the software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These preconditions are generally not included in the software installation package and need to be installed separately.

In order to use CODIAC, one should have the following:

* **Operating System:** Windows 7 and above
* **C Compiler:** GNU Compiler Collection (GCC)
* **Editor:** Any text editor . (Example: Notepadd++)

## HARDWARE REQUIREMENTS

Hardware requirements refer to the common set requirements defined by any operating system or software application and are usually the physical computer resources. In this, we look into the architecture, processing power, memory, secondary memory, display adapter and peripherals.

In order to use this project, one should have the following:

* + - **Processor:** Intel Pentium processor and above
    - **Memory:** 4 GB RAM and above

1. **PROPOSED WORK**
   1. **DESIGN:**
2. **USE CASES DIAGRAM**

**Diagram

Description automatically generated**

1. **FLOW CHART**

**Diagram

Description automatically generated**

**b. IMPLEMENTATION:**

1. **DESCRIPTION OF MAIN MODULES**

**LOGIN:**

The Regular user have to login with the predefined username and password, where as for new user no need to login. If the password and username are attempted incorrectly, then it prompts the user to re-enter the credentials. If either user name or password is invalid, then display error message and prompts for valid user name or password​.

**SLOT AVAILABLITY:**

The admin checks the available slots, if the slots are empty it will be displayed as “there are no slots available for parking” else, if the slots are available then the slots will be allocated to the user. According to the slots displayed by the admin it parks the vehicle.

**ARRIVAL OF THE VEHICLE:**

The user chooses whether they are regular user or new user.

According to their choice (new or regular), the user can park their own vehicle by selecting the type of their vehicle and provides their details given by the user.

**GENERATING TICKET:**

The admin generates the ticket based on the details provided by the user at the time of slot booking. If the details provided are not complete then it ensures the user to complete it. If all the details are valid then the ticket will be generated.

**DEPARTURE OF THE VEHICLE:**

Vehicles get departure from the parking slot whenever the user opts for departure. Based on the exit time of the departure vehicle the price will be generated, where regular users amount will be automatically deducted from their parking cash amount, on the other hand new user will pay cash directly after departure of the vehicle.

**GENERATING PRICE:**

Whenever the vehicle gets departured, the price will be generated according to the exit time of that particular departure vehicle.

Price = (exit time – entry time) / hour.

The regular user will be benifitted with this parking management system as they can pay through their parking cash with less parking charges when compared to the new user

1. **SPECIFIC ALGORITHMS / LOGIC TO BE HIGHLITED:**
   * 1. *The following code stores the previous transactions like the parking cash amount of the regular user*

if(ar[number]==0)

{

printf("\nenter the amount to be deposited\n");

scanf("%d",&s->pc);

ar[number]=s->pc;

}

else

{

s->pc=ar[number];

printf("\nYou are a regular user and your parking cash after the past transaction is %d\n",ar[number]);

}

Algorithm:

Step 1- Start.

Step 2- Checks whether the array of that vehicle number is zero or not.

Step 3- If it is zero (the user’s parking cash amount = 0) then, it prompts the regular user to deposit amount.

Step 4- else if it non-zero, then he already has amount in his account, Directly that amount will be accssed by that regular user and displays the current parking cash amount.

Step 5- Stop.

* + 1. *if the regular user’s parking cash is not enough for making transactions for the departure of the parked vehicle then it prompts the user to pay the remaining amount through cash .If the same regular user comes again it will ask to deposit amount since he don’t have any parking cash amount left in his account*

if(v->pc<=0)

{

printf("\nplease pay the remaning amount of %d\n",-(v->pc));

v->pc=0;

}

1. **GITHUB LINKS AND FOLDER STRUCTURE:**

We have segregated the files we have used based on the usage. We also created a README which has a brief description of our project. The Main C file is not part of any folder. We have both our final presentation and documentation of the project which are not part of any folder.

Name: Harshith

Roll no:1602-19-737-073

GITHUB LINK:

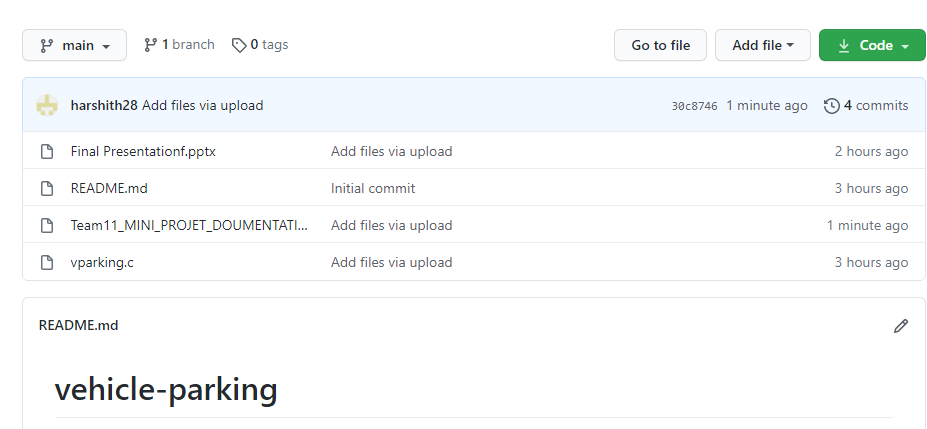
<https://github.com/harshith28/vehicle-parking>

Name: Vivek

Roll no :1602-19-737-122

GITHUB LINK:

<https://github.com/Vivek939/vehicle-parking-management-system>



## c. TESTING.

**USER TEST CASES**

The user has 6 major functionalities: Login, Slot availability, Generating ticket, Generating price, Arrival of the vehicle and Departure of the vehicle. Below are the testcase description:

A picture containing text

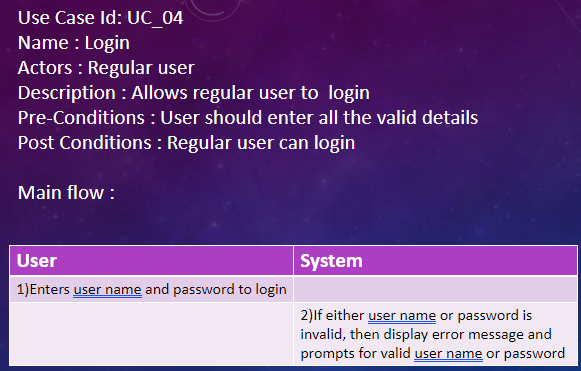
Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated



Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

# 4.RESULTS

* **Arrival of a regular user, if the vehicle no is invalid it shows invalid vehicle no. and prompts to enter again.**

Text

Description automatically generated

* **Ticket will be generated and it shows no of vehicle parked and slots for each vehicle**

Text

Description automatically generated

* **If the entry time is invalid it shows invalid entry time and prompts to enter again.**

Text

Description automatically generated

* **If the regular user enters invalid user name or password it shows invalid user name or password and prompts to enter again.**

Text

Description automatically generated

A picture containing text

Description automatically generated

Text

Description automatically generated

A picture containing text

Description automatically generated

Text

Description automatically generated

* **If the regular user deposited only 5 rupees and amount to be paid is 30 rupees then the remaining 25 rupees has to be paid directly to the admin**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

* **The same regular user who has 0 rupees in his account parks a vehicle then it asks that user to redeposit the parking cash again**

Text

Description automatically generated

A picture containing text

Description automatically generated

* **Here we departure vehicle no 888. On that slot 888 will be replaced with 0**

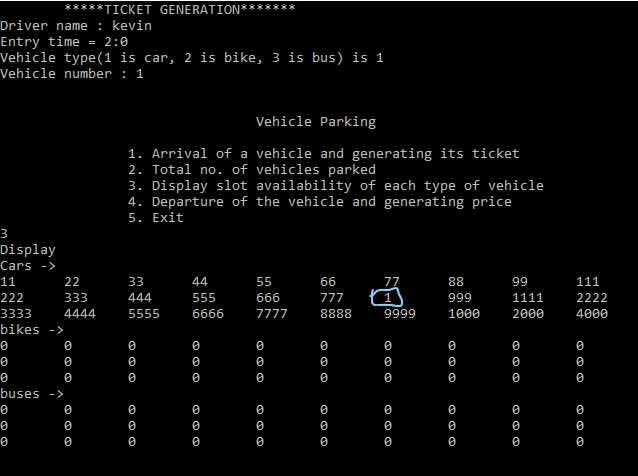
Text

Description automatically generated

Text

Description automatically generated

* **Here another new vehicle will be parked in the slot of 0(which was departure earlier)**



# 5.ADDITIONAL KNOWLEDGE ACQUIRED

Implementing this project in C Language has introduced us to different libraries such as: ‘conio.h’, ‘time.h’ and ‘windows.h’. We were able to use the knowledge we have on the Linked List Data Structure and execute it as a real-time application. We used the ‘windows.h’ library for controlling the display colours in a controlled manner. We explored the ‘time.h’ and ‘conio.h’ libraries for achieving a look-and-feel of an actual window application by constructing our own time delay function.

Also, we have further improved in our knowledge in file-handling because of the vast amount of data manipulation we have done using text files.

Other than this, we have learnt the value of team spirit and have understood the intention behind working in teams. We have learnt to be team players.

# 6.CONCLUSION AND FUTURE WORK

To conclude, this application is useful not only for educational institution but

Also for any other organization who are keen to utilize this kind of application. It can be operated very easily.

Future work is to we develop the application in PHP Or HTML to make the real payment, make this console application as a web application.

**7. REFERENCES**

1. C Language Documentation: [https://docs.microsoft.com/en-us/cpp/c- language/?view=msvc-160](https://docs.microsoft.com/en-us/cpp/c-language/?view=msvc-160)
2. Stack Overflow (for debugging errors): <https://stackoverflow.com/>