**OPTIMISATION OF PARKING SLOT**

By

ABHISHEK KUMAR (1609114001)

PRAKHAR SRIVASTAVA (1609114025)

SANYA BANSAL (1609114030)

SHOBHIT SAXENA (1609114032)

Submitted to the Department of Computer Applications

In partial fulfillment of the requirements

For the degree of

Master of Computer Application



JSS Academy of Technical Education

A.K. Technical University

May, 2018

Submitted to

**Prof. VIKAS SINGHAL**

**INDEX**

|  |  |
| --- | --- |
| ***Topic*** | ***Page No.*** |
| 1. INTRODUCTION    * 1. Objective 2. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT 3. ANALYSIS 4. SOFTWARE &HARDWARE REQUIREMENT SPECIFICATION 5. SNAPSHOTS 6. CODING 7. FUTURE SCOPE 8. REFERENCES | **4**  **5**  **6-7**  **8**  **8-10**  **10-16**  **11-17**  **18** |

**ACKNOWLEDGEMENT**

First of all we thank The Almighty God for blessing us and Mentors for supporting us throughout this endeavour.

We take this opportunity to express our profound and sincere gratitude to our guide **Mr. Vikas Singhal** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project.

We gratefully acknowledge the support extended by Head of the Department, **Mr. Vikas Singhal** and also thank him for letting us use the lab facilities.

We are also obliged to the staff members of the division for their cooperation during the period of the project.

Lastly, we thank all our friends without whom this project would not be possible.

ABHISHEK KUMAR (1609114001)

PRAKHAR SRIVASTAVA (1609114025)

SANYA BANSAL (1609114030)

SHOBHIT SAXENA (1609114032)

**INTRODUCTION**

Parking is considered a major land use challenge in campus planning. The problem can be in terms of scarcity (few available spaces compared to demand) or management (inefficient usage of available facilities). Many studies have looked at the parking problem from the administrative and management points of view. However, it is believed that mathematical models and optimization can provide substantial solution to the parking problem. This project investigates a model for allocating car parking spaces in the university environment and improves on the constraints to address the reserved parking policy on campus.

In this project, we will identify the type of vehicle (i.e. regular or visitor) that enters to college campus and will keep the track of time duration it resided within the college campus. This project will also provide necessary information about available parking slots in order to optimally use the parking area of the college and will give an alarm if visitors vehicle resides in college premises after a time limit being provided.

* + 1. **OBJECTIVE**

This project’s main objective is:-

1. Identifying type of vehicle entering in the college premises i.e. regular visitor like faculty or student or guest visitor like taxi etc.
2. Providing a parking slot to the vehicle entering in college premises in an optimised way.
3. If any vehicle stays for more than a specific time the software will give a beep sound alarming the guards to have a check in order to prevent any of the suspicious activity and to provide security.

**FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT**

FUNCTIONAL

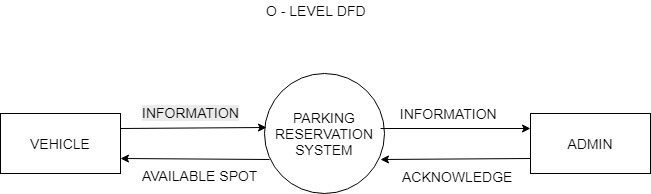
* As soon as the vehicles enters the college campus, a manual entry is done by the guard about the vehicle details and where it needs to be parked.
* After keeping track, vehicle will be provided a parking slot based on their need.
* While entering of the vehicle, we will extract the information whether vehicle is a regular or visitor and accordingly timer will be set.
* If the vehicle surpasses this time duration an alarm with beep sound come into action and security in-charge is informed.

NON-FUNCTIONAL

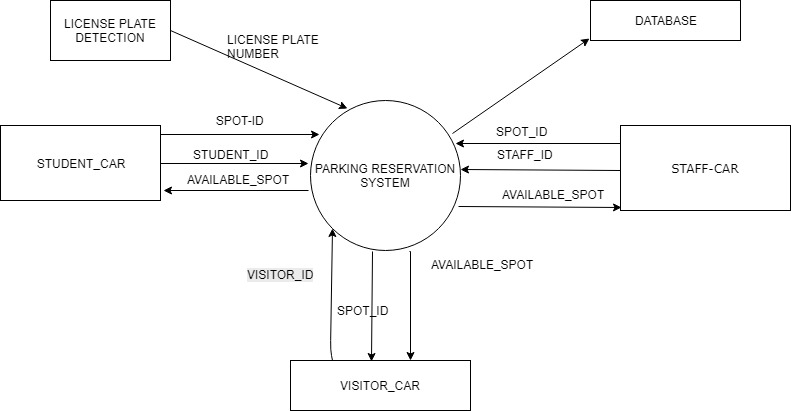
* **Efficiency :-** The extent to which the software system handles capacity, throughput, and response time.
* **Integrity :-** The degree to which the data maintained by the software system are accurate, authentic, and without corruption.
* **Reliability :-** The extent to which the software system consistently performs the specified functions without failure.
* **Flexibility :-** The ease with which the software can be modified to adapt to different environments, configurations, and user expectations.
* **Maintainability :-** The ease with which faults in a software system can be found and fixed.
* **Modifiability :-** The degree to which changes to a software system can be developed and deployed efficiently and cost effectively.
* **Portability :-** The ease with which a software system can be transferred from its current hardware or software environment to another.
* **Reusability :-** The extent to which a portion of the software system can be converted for use in another system.

**ANALYSIS ( DFD)**

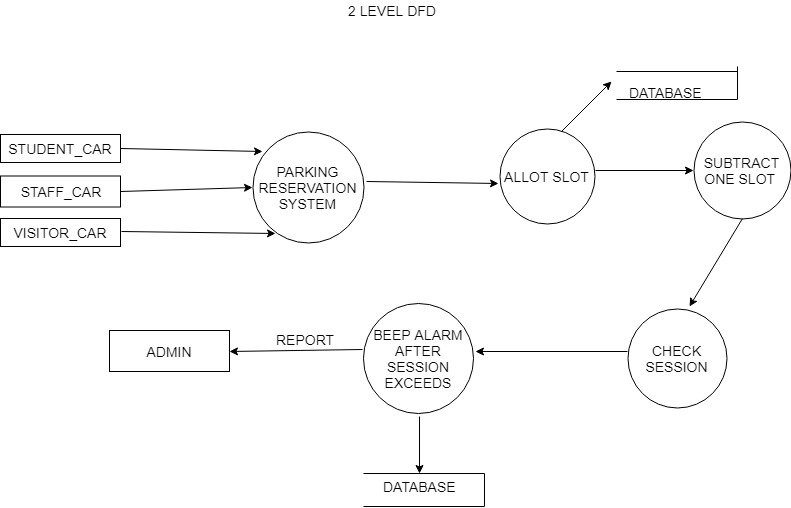
**0 level DFD**



**1 level DFD**



**2 LEVEL DFD**



**6) HARDWARE & SOFTWARE REQUIREMENTS:**

The following are the minimal hardware and software requirements for this project:-

**HARDWARE REQUIREMENTS**

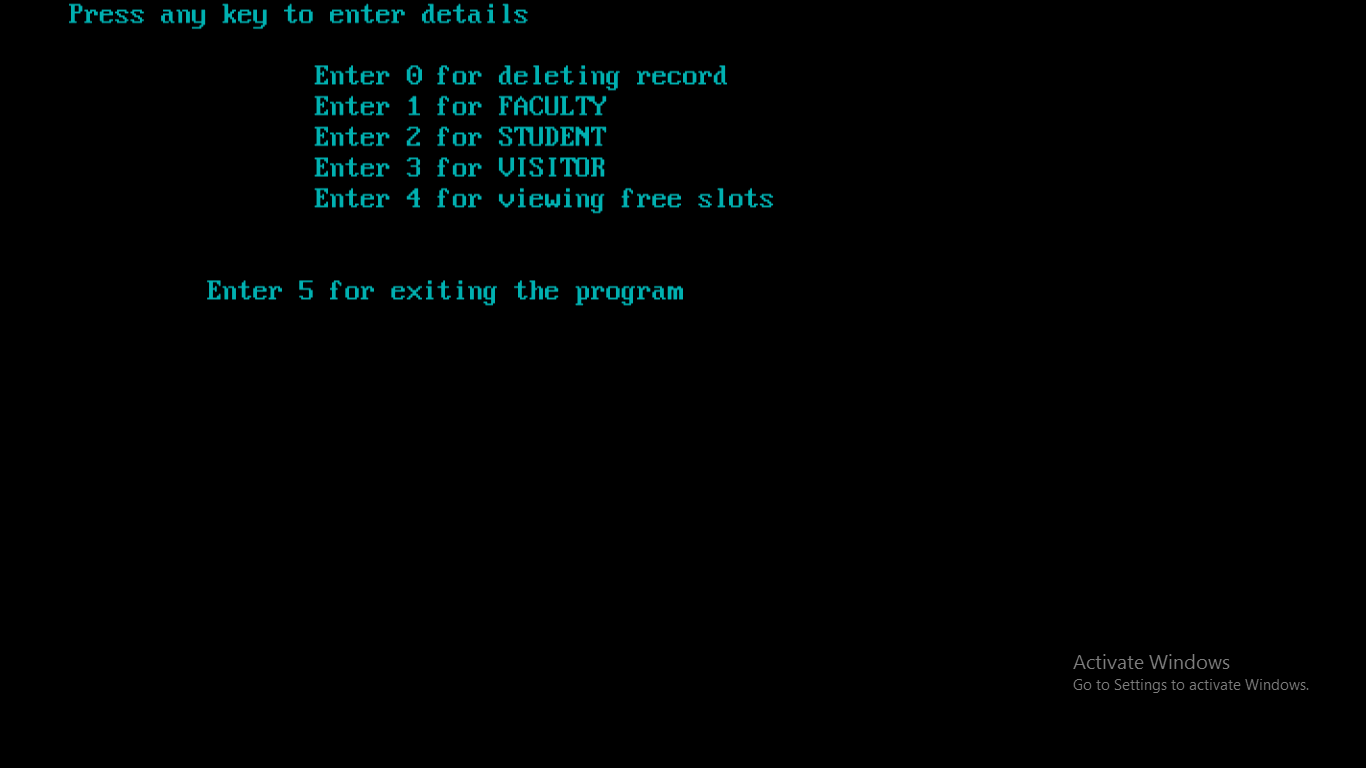
* + A Computer system.
  + An audio output device such as speaker.

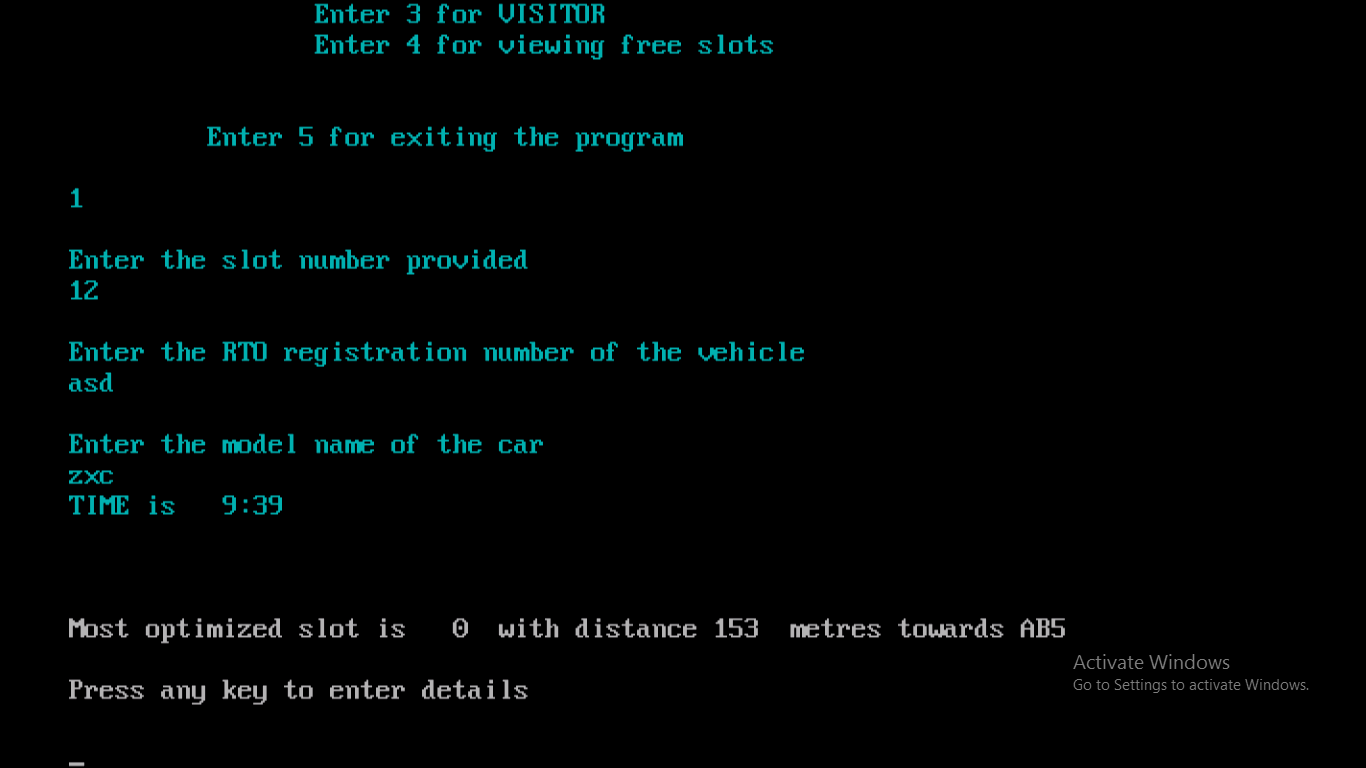
**SOFTWARE REQUIREMENTS**

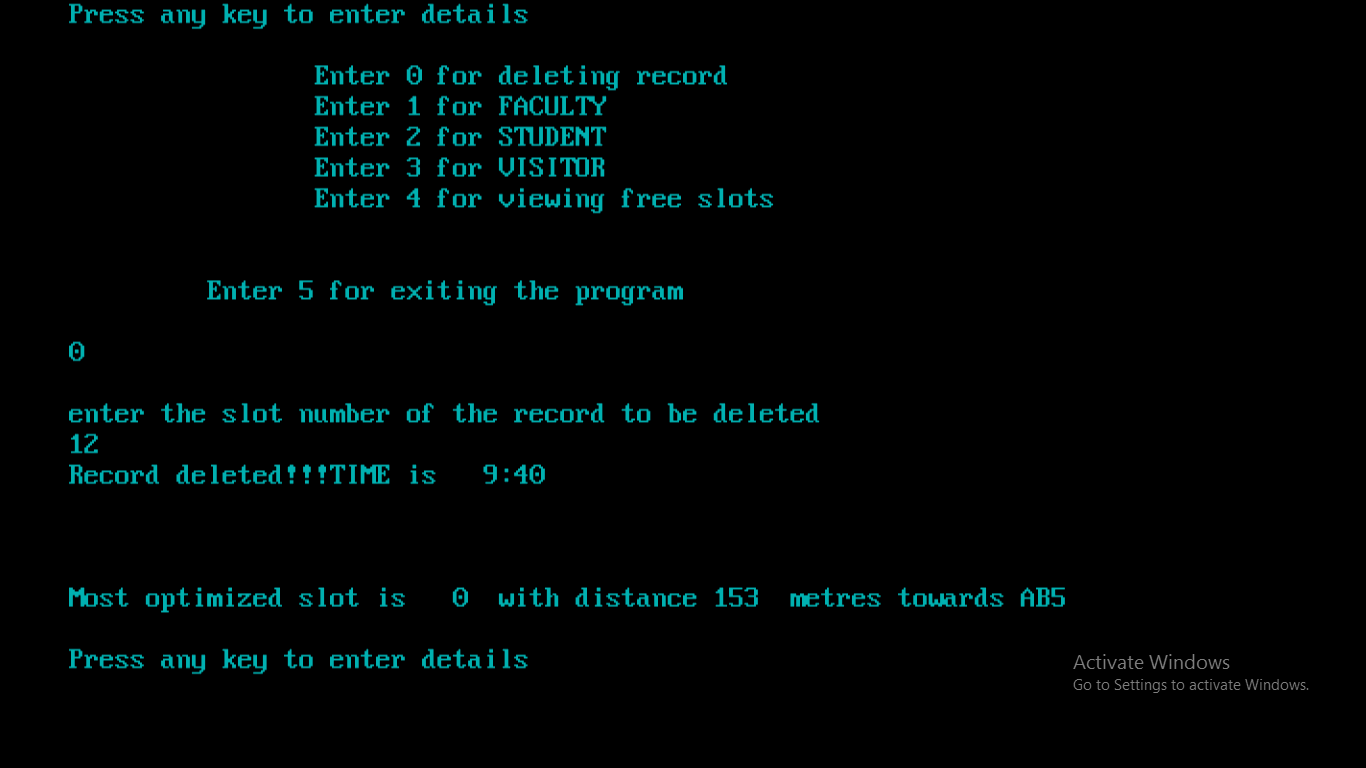
* + Borland C++ Builder/ Microsoft Visual C++ (Licensed Version)
  + Microsoft Windows 7 or above.(32 bit or above).

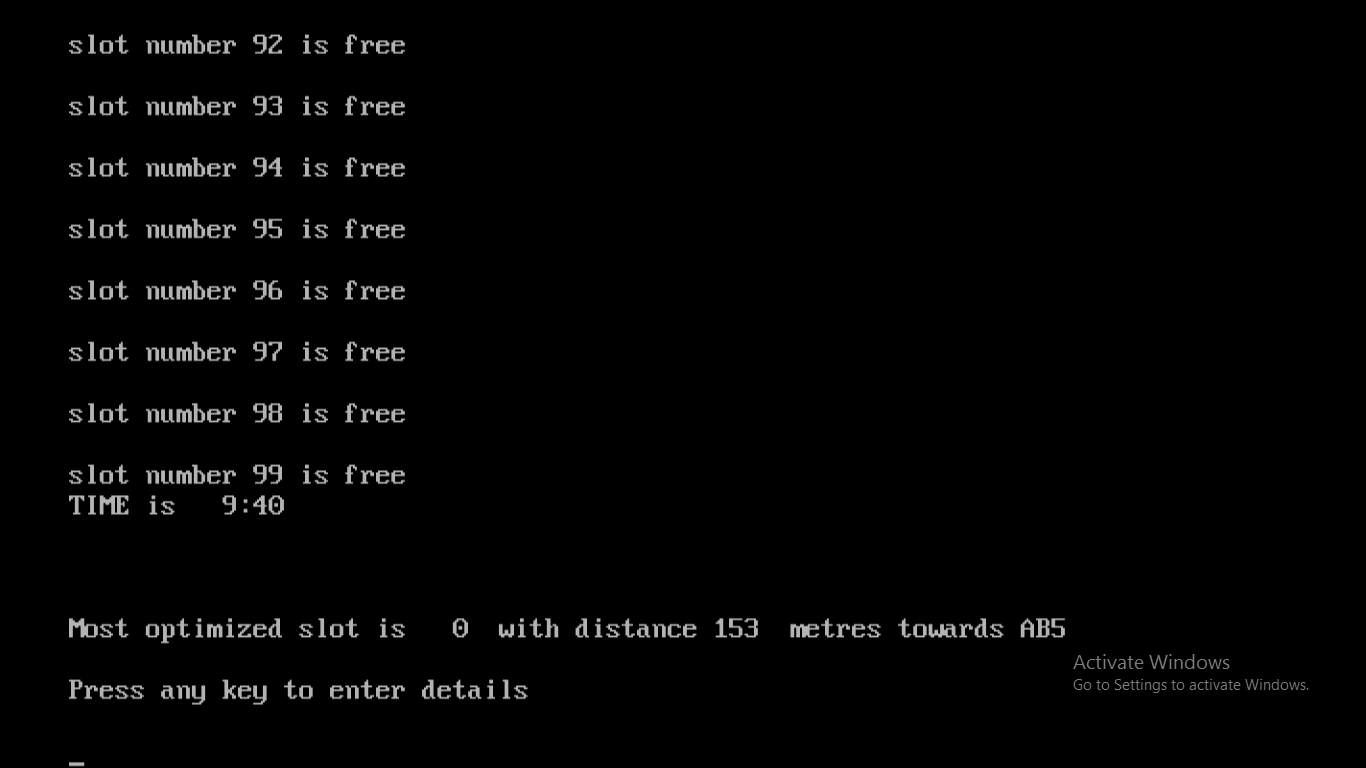
**SCREENSHOTS**

****

****

****

****

****

**CODING**

#include <iostream.h>

#include <conio.h>

#include <dos.h>

#include <stdio.h>

int count[100];

int size=0;

int m,h,m1,h1;

struct time t;

struct car

{

char rto\_reg[30];

char model[30];

int slot;

};

car obj[100];

void faculty();

void del();

void student();

void visitor();

void view\_slot();

void check();

void main()

{

clrscr();

flushall();

int input,i,c;

for(i=0;i<=99;i++)

count[i]=1000;

while(!kbhit())

{ ref:

check();

gettime(&t);

m1=t.ti\_min,h1=t.ti\_hour;

textcolor(3);

cout<<"TIME is "<<h1<<":"<<m1<<endl<<endl<<endl<<endl;

//clrscr();

//delay(1000);

i=0;

for(i=0;i<100;i++)

{

c=count[i];

if(c==1000)

{

cout<<"Most optimized slot is "<<i<<" with distance "<<150+i+3<<" metres towards AB5"<<endl<<endl;

break;

}

}

cout<<"Press any key to enter details "<<endl<<endl;

/\*cout<<"\t\tEnter 0 for deleting record"<<endl;

cout<<"\t\tEnter 1 for FACULTY"<<endl;

cout<<"\t\tEnter 2 for STUDENT"<<endl;

cout<<"\t\tEnter 3 for VISITOR"<<endl;

cout<<"\t\tEnter 4 for viewing free slots"<<endl;

cout<<"\t\tEnter 6 for viewing this menu"<<endl;

cout<<"\n\n\t Enter 5 for exiting the program"<<endl<<endl;\*/

delay(1000);

clrscr();

}getch();

cout<<"Press any key to enter details "<<endl<<endl;

cout<<"\t\tEnter 0 for deleting record"<<endl;

cout<<"\t\tEnter 1 for FACULTY"<<endl;

cout<<"\t\tEnter 2 for STUDENT"<<endl;

cout<<"\t\tEnter 3 for VISITOR"<<endl;

cout<<"\t\tEnter 4 for viewing free slots"<<endl;

//cout<<"\t\tEnter 6 for viewing this menu"<<endl;

cout<<"\n\n\t Enter 5 for exiting the program"<<endl<<endl;

cin>>input;

// else if(input==6)

// goto ref;

switch(input)

{

case 0:

del();

goto ref;

case 1:

faculty();

goto ref;

//break;

case 2:

student();

//break;

case 3:

visitor();

goto ref;

//break;

case 4:

view\_slot();

goto ref;

//break;

case 5:

goto cls;

case 6:

goto ref;

}

cls:

getch();

}

void faculty()

{

int a;

cout<<"\nEnter the slot number provided \n";

cin>>a;

obj[count[a]].slot=a;

cout<<"\nEnter the RTO registration number of the vehicle"<<endl;

cin>>obj[count[a]].rto\_reg;

cout<<"\nEnter the model name of the car"<<endl;

cin>>obj[count[a]].model;

}

void student()

{

int a;

cout<<"\nEnter the slot number provided \n";

cin>>a;

obj[count[a]].slot=a;

cout<<"\nEnter the RTO registration number of the vehicle"<<endl;

cin>>obj[count[a]].rto\_reg;

cout<<"\nEnter the model name of the car"<<endl;

cin>>obj[count[a]].model;

}

void view\_slot()

{

for(int i=0;i<=99;i++)

{

if(count[i]==1000)

cout<<"\nslot number "<<i<<" is free\n";

}

}

void del()

{

int a;

cout<<"\nenter the slot number of the record to be deleted\n";

cin>>a;

count[a]=1000;

cout<<"Record deleted!!!";

}

void visitor()

{

int a;1

cout<<"\nEnter the slot number provided \n";

cin>>a;

obj[count[a]].slot=a;

count[a]=a;

cout<<"\nEnter the RTO registration number of the vehicle"<<endl;

cin>>obj[count[a]].rto\_reg;

cout<<"\nEnter the model name of the car"<<endl;

cin>>obj[count[a]].model;

flushall();

cout<<"\n set the time u want to be beeped\n";

cout<<"\n\nEnter hour\n";

cin>>h;

cout<<"\n\n Enter minute";

cin>>m;

}

void check()

{

gettime(&t);

m1=t.ti\_min,h1=t.ti\_hour;

if((m==m1)&&(h==h1))

{

sound(1000);

cout<<"ALERT!!!!";

delay(5000);

nosound();

}

}

**FUTURE SCOPE**

In the future a digital camera can be installed to identify the vehicles which will processs the image and will automatically provide the essential details about the vehicles. This will make this software fully automated. Since the software is now capable of identifying the model of vehicles this helps in identifying the dominant or leading zones of a specific Brand. Equipped with the knowledge of the leading zones various vehicle manufacturing companies can make strategies related to their service stations, target region for new launches etc.

**REFERENCES**

1. https://innovate.mygov.in/sih2017/

2. <https://requirementsquest.com/wp.../Nonfunctional-Requirement-EXAMPLES.pdf>

3. http://www.tutorialdost.com/Cpp-Programming-Tutorial/27-Cpp-Structure-Array.aspx