JAVA PROJECT

Group 17

K. Sushma Rachel

Steven Muvva

Siva Sai Varma

Parking Management

The aim of this project is to reduce traffic in the parking area. It helps in knowing where a vehicle can be parked by checking the parking area through a computer. It can detect every free and filled parking slot and shows a suitable parking slot for the vehicle. It helps to park the vehicle without any ambiguity and also works as an easy way to detect parking slots and park the vehicle.

In this software the computer asks the user for some basic details like name, mobile number, type of vehicle, vehicle number and then gives the user a parking slot. It stores all the entered data that can be reviewed for later.

CODE -

```
package project;

import java.util.*;

import java.io.*;

import java.text.SimpleDateFormat;

import java.time.*;



class Slot

{

int x=0;

int y=0;

}



interface intfmat

{

 Slot allocateLocation(char type);

 boolean remove(Slot locate);

 void display();

 char mat[][]=new char [32][32];
```

```
}


class grid implements intfmat
{
public char parkArea[][];
grid() {
parkArea=new char[32][32];
for(int i=0;i<32;i++)
        for(int j=0;j<32;j++)
    {
     parkArea[i][j]=' ';
    }

for(int i=0;i<32;i++)
        for(int j=0;j<32;j++)
        {
                if(i==0||(i==31&&(j!=31&&j!=30&&j!=29)))
                        parkArea[i][j]='-';
                if(j==0||j==31)
                        parkArea[i][j]='|';
```

```
            }


for(int i=2;i<31;i+=4)
            for(int j=1;j<29;j+=2)
            {
              parkArea[i][j]=' ';
            }


for(int i=1;i<31;i+=2)
            for(int j=1;j<29;j+=1)
            {
                        parkArea[i][j]='-';
            }


  for(int i=2;i<31;i+=4)
            for(int j=2;j<29;j+=2)
            {
                        parkArea[i][j]='|';}
            }
```

```
public Slot allocateLocation(char type) {
        int m=1;
        Slot loc=new Slot();
        loc.x=-1;
        loc.y=-1;
        for(int i=2;i<31;i+=4)
        {
                for(int j=1;j<29;j+=2)
                {
                        if(parkArea[i][j]==' '&&m==1)
                        {
                                parkArea[i][j]=type;
                                m=0;
                                loc.x=((i-2)/4)+1;
                                loc.y=((j-1)/2)+1;
                                return loc;
                        }
                }
        }
        return loc;
```

```
}

public boolean remove(Slot loc)
{
	if(parkArea[loc.x][loc.y]==' ')
	{
		return false;
	}
	parkArea[loc.x][loc.y]=' ';
	return true;
}

public void display()
{
	for(int i=0;i<32;i++)
	{
		System.out.print("\t");
		for(int j=0;j<32;j++)
		{
			System.out.print(parkArea[i][j]+" ");
```

```
                }
                System.out.println("");
        }
}
}

class PM{
  public static void main(String[] args)throws IOException, InterruptedException  {
  Slot loc=new Slot();
  grid grid=new grid();
  Scanner scan=new Scanner(System.in);
  System.out.println("\n\t-------Parking Management System-------\n");
  BufferedReader br = null;
  try
  {
        br=new BufferedReader(new InputStreamReader(new FileInputStream("Matrix.txt")));
        for(int i=0; i<32; i++)
        {
                for(int j=0;j<32; j++)
                {
```

```
                        grid.parkArea[i][j]=(char)br.read();
                    }
            }
            if(br!=null)
                    br.close();
    }
    catch(IOException e)
    {
            System.out.println("\nDatabase not present. A new empty file will be created.");
            File f1 = new File("Matrix.txt");
    }
    stop:
            while(true)
            {
                    System.out.println("\n1. Entering a vehicle \n2. Removing a vehicle\n3. Display parking lot\n4. Display parking logs\n5. Exit\nEnter your choice: ");
                    int cho=0;
                    while(true)
                    {
                            try
```

```
                {
                        cho=scan.nextInt();
                        if(cho<=5&&cho>=1)
                                break;
                        else
                                throw new InputMismatchException();
                }
                catch(InputMismatchException e)
                {
                        System.out.print("\nInvalid \nEnter choice : \n");
                        scan.next();
                }
        }
        switch(cho)
        {
                case 1:
                        System.out.println("\nEnter type of vehicle ( 'c' for car and 't' for
two-wheeler)");
                        char type=' ';
                        while(true)
```

```
{
        try
        {
                type=scan.next().charAt(0);
                if(type!='c'&&type!='t')
                        throw new InputMismatchException();
                else
                        break;
        }
        catch(InputMismatchException e)
        {
                System.out.println("\nInvalid \nEnter choice : ");
        }
}
loc =grid.allocateLocation(type);
if(loc.x!=-1)
{
        System.out.print("Name : ");
        scan.nextLine();
        String name=scan.nextLine();
```

```
String vehicleNo;

while(true)

{

System.out.print("Vehicle Number (Ex: TS09-ER-4561)
: ");

vehicleNo=scan.next();

if(vehicleNo.charAt(0)>=65&&vehicleNo.charAt(0)<=90&&vehicleNo.charAt(1)>=65&&vehicleNo.cha
rAt(1)<=90&&vehicleNo.charAt(2)>=48&&vehicleNo.charAt(2)<=90&&vehicleNo.charAt(3)>=48&&v
ehicleNo.charAt(3)<=57&&vehicleNo.charAt(5)>=65&&vehicleNo.charAt(5)<=90&&vehicleNo.charAt
(6)>=65&&vehicleNo.charAt(6)<=90&&vehicleNo.charAt(8)>=48&&vehicleNo.charAt(8)<=57&&vehi
cleNo.charAt(9)>=48&&vehicleNo.charAt(9)<=57&&vehicleNo.charAt(10)>=48&&vehicleNo.charAt(1
0)<=57&&vehicleNo.charAt(11)>=48&&vehicleNo.charAt(11)<=57)

break;

else

System.out.println("\nInvalid \nEnter Vehicle
Number : \n");

}

String mobileNo;

while(true)

{

System.out.print("Contact number : ");

mobileNo=scan.next();

if(mobileNo.length()!=10)
```

```
                    System.out.println("\nInvalid \nEnter Contact
Number : \n");
                else
                {
                    System.out.println();
                    break;
                }
            }

            File f;
            try
            {
                f=new File("log.txt");
                if(!f.exists())
                {
                    f.createNewFile();
                }
                PrintWriter pw=new PrintWriter(new FileWriter(f,
true));
                pw.println("Entry:");
                pw.println("Name: "+name);
```

```
if(type=='c')
        pw.println("Vehicle type: Car");
else
        pw.println("Vehicle type: Two-wheeler");
pw.println("Vehicle Number: "+vehicleNo);
pw.println("Contact Number: "+ mobileNo);
pw.println("Location of parking(x y): "+loc.x+"
","+loc.y);

int day, month, year;
int second, minute, hour;
GregorianCalendar date = new GregorianCalendar();

day = date.get(Calendar.DAY_OF_MONTH);
month = date.get(Calendar.MONTH);
year = date.get(Calendar.YEAR);

second = date.get(Calendar.SECOND);
minute = date.get(Calendar.MINUTE);
hour = date.get(Calendar.HOUR);
```

```
                    pw.println("Date: "+day+"/"+(month+1)+"/"+year);
                    pw.println("Time: "+hour+":"+minute+":"+second);
                    pw.println();
                    pw.println();
                    pw.close();

                }
                catch(IOException e)
                {
                    System.out.println("\nProblem found in creating logs
/nEnter logs manually");
                }

                System.out.println("\nYour parking location is row "+loc.x+"
and column "+loc.y+" .\n");

            }
            else
                System.out.println("\nSorry! all parking slots are full\n");
            break;
```

```
                case 2:

                        System.out.println("\n");

                        grid.display();

                        System.out.println("\tEnter the coordinates to remove (Example: x y) :
");

                        loc.x=scan.nextInt();

                        loc.x=4*(loc.x-1)+2;

                        loc.y=scan.nextInt();

                        loc.y=2*(loc.y-1)+1;

                        boolean check=false;

                        check=grid.remove(loc);

                        String vehicleName;

                        int cost_car=400;

                        int cost_bike=300;

                        Scanner sc = new Scanner(System.in);

                        vehicleName = sc.nextLine();

                        if(vehicleName=="Car")

                        {

                                System.out.println("Cost= "+cost_car);

                        }
```

```
else if(vehicleName=="Bike")
{
        System.out.println("Cost= "+cost_bike);
}
if(check==true)
{
        System.out.print("Enter vehicle number: ");
        scan.nextLine();
        String exiting=scan.nextLine();
        try
        {
                File f=new File("log.txt");
                if(!f.exists())
                        f.createNewFile();

                PrintWriter pw=new PrintWriter(new FileWriter(f,
true));
                pw.println("Exit:");
                pw.println("Vehicle Number: "+exiting);
                int day, month, year;
```

```
            int second, minute, hour;
            GregorianCalendar date = new GregorianCalendar();

            day = date.get(Calendar.DAY_OF_MONTH);

            month = date.get(Calendar.MONTH);
            year = date.get(Calendar.YEAR);

            second = date.get(Calendar.SECOND);
            minute = date.get(Calendar.MINUTE);
            hour = date.get(Calendar.HOUR);
            pw.println("Date: "+day+"/"+(month+1)+"/"+year);
            pw.println("Time: "+hour+":"+minute+":"+second);
            pw.println();
            pw.println();
            pw.close();
        }
        catch(IOException e)
        {
```

```
                                        System.out.println("/nExit log entry failed/nPlease enter
to file manually");
                                }


                        }
                        else
                                System.out.println("\nThere is no vehicle in the parking lot\n");
                        break;

                case 3:
                        grid.display();
                        break;

                case 4:
                try
                {
                        File f=new File("log.txt");
                        BufferedReader bri=new BufferedReader(new FileReader(f));
                        String buffer;
                        System.out.println("\nDisplaying previous Logs ");
```

```
                    System.out.println();
                    System.out.println();
                    while((buffer=bri.readLine())!=null)
                    {
                        System.out.println(buffer);
                    }
                    bri.close();

                }
                catch(IOException e)
                {
                    System.out.println("Error in displaying Logs on the console /nPlease refer to file manually");
                }
                break;

                default:
                break stop;
            }
```

```
}
BufferedWriter bo = null;
try
{
        bo = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("Matrix.txt")));
        for(int i=0; i<32;  i++)
        {
                for(int j=0; j<32; j++)
                {
                        bo.write(grid.parkArea[i][j]);
                }
        }
        if(bo!=null)
                bo.close();
}
catch(IOException e)
{
        System.out.println("IO error.");
}
System.out.println("\n-------Exiting from software-------\n");
```

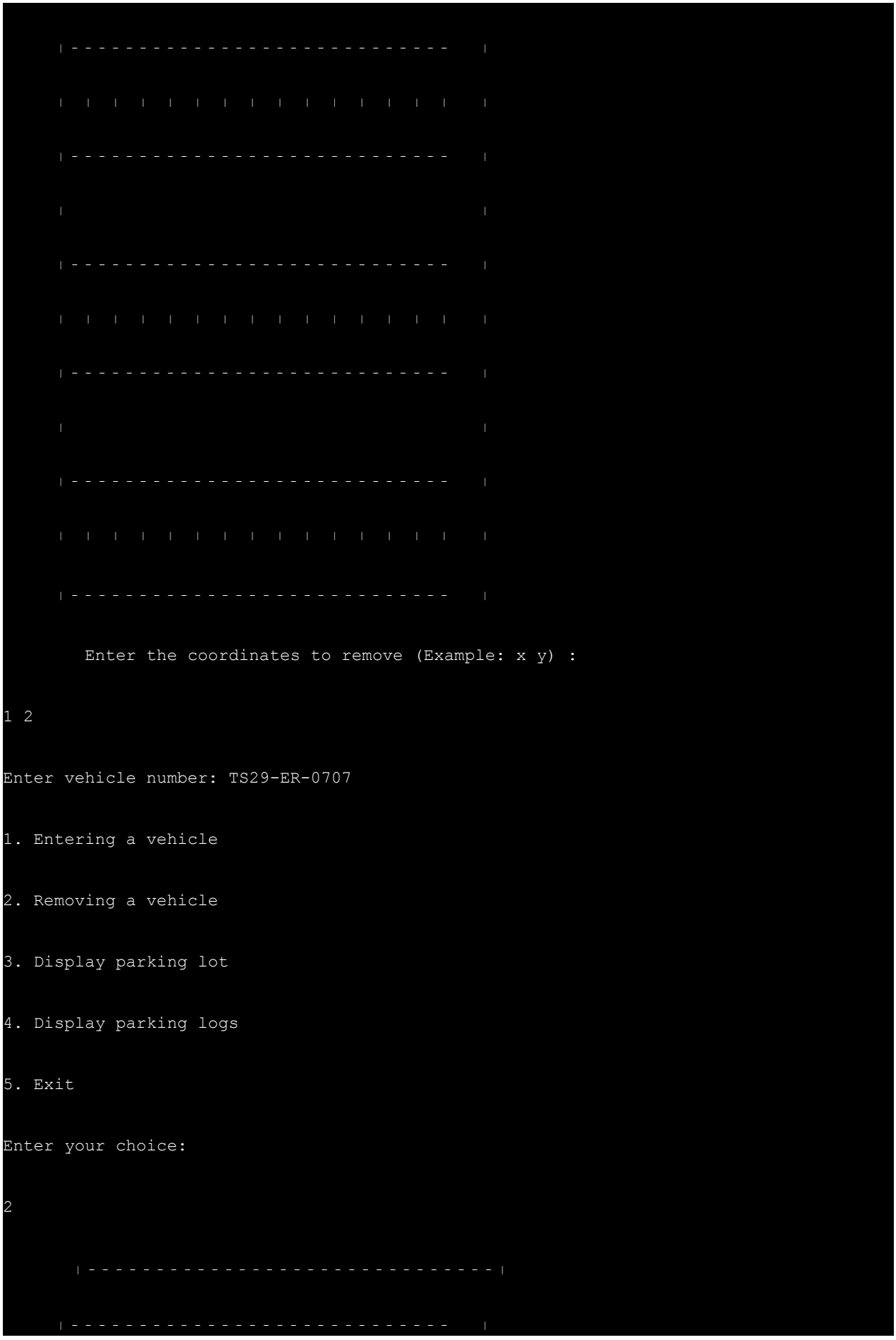
```
}

}
```

INPUT and OUTPUT -

```
       -------Parking Management System-------



Database not present. A new empty file will be created.


1. Entering a vehicle

2. Removing a vehicle

3. Display parking lot

4. Display parking logs

5. Exit

Enter your choice:

1

Enter type of vehicle ( 'c' for car and 't' for two-wheeler)

c
```

```
Name : ars

Vehicle Number (Ex: TS09-ER-4561) : TS29-ER-1470

Contact number : 9876543210

Your parking location is row 1 and column 1 .

1. Entering a vehicle

2. Removing a vehicle

3. Display parking lot

4. Display parking logs

5. Exit

Enter your choice:

1

Enter type of vehicle ( 'c' for car and 't' for two-wheeler)

t

Name : def

Vehicle Number (Ex: TS09-ER-4561) : TS29-ER-0707

Contact number : 9988776655

Your parking location is row 1 and column 2 .

1. Entering a vehicle

2. Removing a vehicle

3. Display parking lot

4. Display parking logs
```

```
5. Exit

Enter your choice:

2

      |-----------------------------------|

     |-------------------------------    |

     |c|t|  |  |  |  |  |  |  |  |  |  |  |    |

     |-------------------------------    |

     |                                   |

     |-------------------------------    |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |    |

     |-------------------------------    |

     |                                   |

     |-------------------------------    |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |    |

     |-------------------------------    |

     |                                   |

     |-------------------------------    |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |    |

     |-------------------------------    |

     |                                   |

     |-------------------------------    |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |    |

     |-------------------------------    |

     |                                   |
```



```
     |-------------------------------   |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |

     |-------------------------------   |

     |                                  |

     |-------------------------------   |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |

     |-------------------------------   |

     |                                  |

     |-------------------------------   |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |

     |-------------------------------   |

        Enter the coordinates to remove (Example: x y) :

1 2

Enter vehicle number: TS29-ER-0707

1. Entering a vehicle

2. Removing a vehicle

3. Display parking lot

4. Display parking logs

5. Exit

Enter your choice:

2

       |--------------------------------|

     |-------------------------------   |
```

```
     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |

     |- - - - - - - - - - - - - - - - - - - - - - -   |

     |                                               |

     |- - - - - - - - - - - - - - - - - - - - - - -   |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |

     |- - - - - - - - - - - - - - - - - - - - - - -   |

        Enter the coordinates to remove (Example: x y) :

1 1

Enter vehicle number: TS29-HR1470

1. Entering a vehicle

2. Removing a vehicle

3. Display parking lot

4. Display parking logs

5. Exit

Enter your choice:

3

       |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - |

     |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |

     |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   |

     |                                                               |

     |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   |

     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |
```

```
     |  |  |  |  |  |  |  |  |  |  |  |  |  |   |

     |-------------------------------   |



1. Entering a vehicle

2. Removing a vehicle

3. Display parking lot

4. Display parking logs

5. Exit

Enter your choice:

4



Displaying previous Logs

Entry:

Name: ars

Vehicle type: Car

Vehicle Number: TS29-ER-1470

Contact Number: 9876543210

Location of parking(x y): 1 ,1

Date: 29/11/2020

Time: 8:45:32
```

```
Entry:

Name: def

Vehicle type: Two-wheeler

Vehicle Number: TS29-ER-0707

Contact Number: 9988776655

Location of parking(x y): 1 ,2

Date: 29/11/2020

Time: 8:46:28




Exit:

Vehicle Number: TS29-ER-0707

Date: 29/11/2020

Time: 8:46:53



Exit:

Vehicle Number: TS29-HR1470

Date: 29/11/2020

Time: 8:47:49
```

```
1. Entering a vehicle

2. Removing a vehicle

3. Display parking lot

4. Display parking logs

5. Exit

Enter your choice:

5



-------Exiting from software-------









...Program finished with exit code 0

Press ENTER to exit console.
```

ALGORITHM -

- Create a class name Slot and declare two variables x and y as coordinates.
- Declare an interface named intfmat and declare methods to allocate location, remove and display the parking lot.
- Create a class named grid that implements intfmat. In it declare a matrix named parkArea
- Create a class named grid, that prints the parking area using for loops.
- In grid class, create public Slot allocateLocation
    - Memory allocation for slot
    - location. x and location. y is equal to -1
    - Using for loops return the location
- Public boolean remove is created to remove a vehicle from the slot
    - If (x y) == ‘ ‘ , returns false
    - If (x y) = ‘ ‘, return true
- Public void display is created to display the grid
- In the main class Parking Management
    - allocate memory for the slot, grid, and file
    - In while loop display all the choices.
    - Using switch cases to read the choices from the user
        - Case 1 :
            - ask the user to enter the type of vehicle (2 or 4 wheeler)
            - Next, ask the user to enter details of the user and vehicle.
            - In file f, create an object for a calendar that can automatically update the date and time and printing it to the file
            - Return the location (x y) for the vehicle to park
            - If all slots are filled ‘print all are filled’
        - Case 2 :

- In this, if the user selects an option to remove the vehicle, it prints the parking lot with grid.display()
- It asks for the user to enter coordinates and vehicle number
- It prints the amount to be paid, for 4-wheeler =400 and 2-wheeler = 300.
- In file f using the calendar, it updates the exit date and time.
- It there is no vehicle found it prints not found

- Case 3 :
  - It displays the parking area
- Case 4 :
  - In file f the details of all the vehicle entries and exits are stored.
  - If the user chooses 5 as his/her choice then this software is exited.