11/2/2021

CAR POOLING

DATA-PIRATES

JAVA MINI PROJECT

CAR POOLING

1)TEAM NAME: Data Pirates.

2)Date of start of Project:9/14/2021.

3)Team Members:

S.no	Name	Registration Number
1	Bharath Kumar Uppala	9920004404
2	Narayanam Prudhvish	9920004412
3	Vutthunoori Yagnesh	9920004451
4	LV.Tharun Kumar Reddy	9920004386

4)Responsibilities:

S.no	Name	Responsibility
1	Bharath Kumar Uppala	Developed connection b/w program and SQL Data base
2	Narayanam Prudhvish	Developed the Graphic user interface
3	Vutthunoori Yagnesh	Developed ideas and worked in fixing bugs
4	LV.Tharun Kumar Reddy	Monitoring the Team and Developing the Ideas

5)Main Categories identified:

a) Instruction to the Customer.

b) User account creation and login.

c)Vehicle management.

d)Price for the Pool.

e) Pool History for the user.

f) Identification of Duplicate User.

g) Provides date of pool.

h) Data Base connection of login details and pool details.

Car Pooling System

Abstract

The online carpooling system is a User-based application is to provide us with a simple riding platform between the car owner and car user. This project enables users to access mobility assets own by others exactly when they need. It shows a medium for available cars to pick up them on the interest of car owner with time and capacity.

Introduction

Personal ride booking and sharing services allow customers to arrange transportation quickly. user can create account and create a pool for his journey.User can select the places for his pool.

Modules

Admin:
Owner accepting nearby user, the owner is going to get the request from the user considering his starting point.

- Step1: Start
- Step2: Admin login with username and password
- Step3: We need to register by giving all the details
- Step4: Start the ride
- Step5: User searching for this ride related post
- Step6: Selecting the ride
- Step7: Viewing user details and ride details
- Step8: Stop

User:
After getting the registration request to the admin, he personally verifies whether the user or the car owner is genuine or not .if genuine then there request is accepted.

- Step 1: Start
- Step 2: Admin will make a post of his ride
- Step 3: User login with username and password
- Step 4: User needs to register the details
- Step 5: Viewing user details
- Step 6: Admin accepts the user details
- Step 7: If the admin does not accept the requesting user needs to choose another driver
- Step 8: Stop

DEVELOPED IN PROGRAM

1)Basic idea and Structural program for carpooling.

2)Developed SQL data base connection.

3)Developed GUI.

4)k-Nearest neighbour algorithm.

Modules in Source code:

```
import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import java.awt.Window.Type;

import java.awt.Dialog.ModalExclusionType;

import java.awt.Color;

import javax.swing.JTextPane;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.sql.SQLException;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;
```

import java.awt.event.ActionEvent;

import javax.swing.JPasswordField;

import javax.swing.JLabel;

import javax.swing.ImageIcon;

Major Agenda of Project

1)Developing the Skills in the connection of Data base.

2)Major Advantages of Car pooling.

3)Developing skills in Major modules of car pooling.

4)Developing the skills in the basic Programming.

5)Building up the Awareness in the industrial level projects.

6)Developing Skills in Graphical user interface.

Sub Domains of Source code

Created 7 classes:

1)Created module with car_pooling:

In this we have imported Java.sql and java.desktop for data base and GUI(Swing and AWT).

2)Created 7 classes

In this project we have created 7 classes for various elements for carpooling.,like：

Car management: In this class we have defined the cars and connection with the data base.

Login Details: In this class we have defined User login credentials and Admin can know the user duplication/creating multiple accounts with same name.

Pool details: In this class car details, pool history, destination all the related pool details are imported to data base.

Source code

Source code with class names:

1)Module:

```
module car_pooling {
	requires java.sql;
	requires java.desktop;
}
```

2)First:

```
package ABC;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.ImageIcon;
import javax.swing.JTextPane;
import java.awt.Color;
import java.awt.Font;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class first extends JFrame {

	private JPanel contentPane;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					first frame = new first();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public first() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 962, 540);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		contentPane.setLayout(null);
```

```
		JTextPane txtpnWelcomeToData = new JTextPane();
		txtpnWelcomeToData.setEditable(false);
		txtpnWelcomeToData.setBackground(Color.WHITE);
		txtpnWelcomeToData.setFont(new Font("Arial", Font.BOLD, 33));
		txtpnWelcomeToData.setForeground(Color.BLACK);
		txtpnWelcomeToData.setText("WELCOME TO DATA PIRATES  CAR-POOLING");
		txtpnWelcomeToData.setBounds(96, 28, 753, 51);
		contentPane.add(txtpnWelcomeToData);

		JTextPane txtpnyouCanFind = new JTextPane();
		txtpnyouCanFind.setEditable(false);
		txtpnyouCanFind.setForeground(Color.BLACK);
		txtpnyouCanFind.setFont(new Font("Arial", Font.BOLD, 13));
		txtpnyouCanFind.setText("1.YOU CAN FIND CAR-POOL\r\n2.VERIFIED PROFILES
ONLY\r\n3.YOU CAN KNOW YOUR POOL HISTORY \r\n4.CALCULATED PRICE WILL BE GIVEN\r\n");
		txtpnyouCanFind.setBounds(25, 419, 282, 74);
		contentPane.add(txtpnyouCanFind);

		JButton btnNewButton = new JButton("PRESS THIS TO CONTINUE");
		btnNewButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				GUI_1 i=new GUI_1();
				i.setVisible(true);
			}
		});
		btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 16));
		btnNewButton.setBounds(652, 423, 270, 51);
		contentPane.add(btnNewButton);

		JLabel lblNewLabel = new JLabel("");
		lblNewLabel.setIcon(new ImageIcon("D:\\3rd sem\\java\\PROJECT\\WhatsApp Image
2021-11-03 at 9.59.32 AM.jpeg"));
		lblNewLabel.setBounds(0, 0, 948, 503);
		contentPane.add(lblNewLabel);
	}

}
```

3)ADD:

```
package ABC;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Toolkit;
import javax.swing.JTextField;
import java.awt.Font;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.ImageIcon;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class ADD extends JFrame {

	private JPanel contentPane;
	private JTextField txtTravellingOutOf;

	/**
	 * Launch the application.
```

```
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					ADD frame = new ADD();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public ADD() {
		setTitle("ADD");
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 700, 540);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		contentPane.setLayout(null);

		txtTravellingOutOf = new JTextField();
		txtTravellingOutOf.setEditable(false);
		txtTravellingOutOf.setFont(new Font("Arial", Font.PLAIN, 34));
		txtTravellingOutOf.setText("Travelling out of town?");
		txtTravellingOutOf.setBounds(29, 111, 358, 40);
		contentPane.add(txtTravellingOutOf);
		txtTravellingOutOf.setColumns(10);

		JButton btnNewButton = new JButton("THEN USE DATA-PIRATES CAR-POOLING\r\n");
		btnNewButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				first f=new first();
				f.setVisible(true);
			}
		});
		btnNewButton.setFont(new Font("Arial", Font.PLAIN, 18));
		btnNewButton.setBounds(219, 429, 417, 51);
		contentPane.add(btnNewButton);

		JLabel lblNewLabel = new JLabel("");
		lblNewLabel.setIcon(new ImageIcon("D:\\3rd sem\\java\\PROJECT\\first.jpeg"));
		lblNewLabel.setBounds(10, 10, 642, 462);
		contentPane.add(lblNewLabel);
	}

}
```

GUI_1:

```
package ABC;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
```

```java
import java.awt.Window.Type;
import java.awt.Dialog.ModalExclusionType;
import java.awt.Color;
import javax.swing.JTextPane;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;
import java.awt.event.ActionEvent;
import javax.swing.JPasswordField;
import javax.swing.JLabel;
import javax.swing.ImageIcon;

public class GUI_1 extends JFrame {

	private JPanel contentPane;
	private JTextField textField;
	private JPasswordField passwordField;
	private JTextField txtGogreen;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					GUI_1 frame = new GUI_1();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public GUI_1() {
		setBackground(Color.BLACK);
		setTitle("                          DATA_PIRATES CAR-POOLING");
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 550, 450);
		contentPane = new JPanel();
		contentPane.setBackground(Color.LIGHT_GRAY);
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		contentPane.setLayout(null);

		JTextPane txtpnUsername = new JTextPane();
		txtpnUsername.setEditable(false);
		txtpnUsername.setFont(new Font("Arial", Font.BOLD, 14));
		txtpnUsername.setForeground(Color.BLACK);
		txtpnUsername.setBackground(Color.LIGHT_GRAY);
		txtpnUsername.setText("USERNAME");
		txtpnUsername.setBounds(36, 135, 89, 23);
		contentPane.add(txtpnUsername);
```

```
		textField = new JTextField();
		textField.setBounds(188, 135, 251, 23);
		contentPane.add(textField);
		textField.setColumns(10);

		JTextPane txtpnPassword = new JTextPane();
		txtpnPassword.setEditable(false);
		txtpnPassword.setText("PASSWORD");
		txtpnPassword.setFont(new Font("Arial", Font.BOLD, 14));
		txtpnPassword.setBackground(Color.LIGHT_GRAY);
		txtpnPassword.setBounds(36, 197, 89, 23);
		contentPane.add(txtpnPassword);

		JButton btnNewButton = new JButton("CREATE A ACCOUNT");
		btnNewButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {

				String username=textField.getText();
				String password=textField.getText();
				LoginDetails lg=new LoginDetails(username,password,"user");
				car_mangement carmg=new car_mangement();
				boolean deltr=carmg.getuserdetails(lg);
				if(!deltr)
				{
				boolean abc = false;
				try {
					abc = carmg.addUser(lg);
				} catch (ClassNotFoundException | SQLException e1) {
					// TODO Auto-generated catch block
					e1.printStackTrace();
				}
				if(abc==true)
				{
					System.out.println("added success");
					JOptionPane.showMessageDialog(null, "successfully created");
				}
				else
				{
				System.out.println("error in  adding");
				JOptionPane.showMessageDialog(null, "error in  adding");
				}
				}
				else
				{
					System.out.println("User duplicate/Already found");
					JOptionPane.showMessageDialog(null, "User duplicate/Already
found");
				}
			}
		});
		btnNewButton.setBounds(22, 304, 159, 39);
		contentPane.add(btnNewButton);

		JButton btnNewButton_1 = new JButton("BOOK A POOL\r\n\r\n");
		btnNewButton_1.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				JOptionPane.showMessageDialog(null, "Login Successfully");
				List<String> loc=new
ArrayList<>(Arrays.asList("hyderabad","kamareddy","nizambad","vizag","vijayavada"));
				List<String> carno=new ArrayList<>(Arrays.asList("ap 1234","ts
1234","re 12","yu 12","kmr 12"));
				//System.out.println("Enter the user name");
				//String username=sc.next();
				//String uuu=username;
```

```
//                          System.out.println(uuu);
                            //System.out.println("Enter password");
//                          sc.nextLine();
                            //String password=sc.next();
                            String username=textField.getText();
                            String password=textField.getText();
                            LoginDetails lg=new LoginDetails(username,password,"user");
                            car_mangement carmg=new car_mangement();
                            boolean abc=carmg.getuserdetails(lg);
                            if(abc==true)
                            {
                                    System.out.println("User already exsists");

      System.out.println("hyderabad\n"+"kamareddy\n"+"nizambad\n"+"vizag\n"+"vijayavada");
                                    System.out.println("Enter pickup");
                                    Scanner sc=new Scanner(System.in);
                                    String pickup=sc.next();

                                    System.out.println("Enter destination");
                                    String desti=sc.next();

                                    System.out.println("Enter the vechile");
                                    for(int i=0;i<carno.size();i++)
                                    {
                                            System.out.println(carno.get(i));
                                    }
                                    String vecno=sc.next();

                                    int total=0;
                                    if(loc.indexOf(pickup)>=0 && loc.indexOf(desti)>=0 &&
loc.indexOf(pickup)!=loc.indexOf(desti))
                                    {
                                            if((loc.indexOf(pickup)-loc.indexOf(desti))<0)
                                            {
                                              total=total+(-(loc.indexOf(pickup)-
loc.indexOf(desti)))*300;

                                            }
                                            else
                                            {
                                            System.out.println((loc.indexOf(pickup)-
loc.indexOf(desti)));
                                            total=total+(loc.indexOf(pickup)-
loc.indexOf(desti))*300;


                                            }LocalDate ld=LocalDate.now();
                                            System.out.println("lg.getUsername"+lg.getUsername()+"
ASDFG+"+total);

                                            PoolDetails pp=new PoolDetails(lg.getUsername(), ld,
vecno, total, pickup, desti);
//                                          boolean confirm;
                                    System.out.println("enter yes to save");
                                    String confirm=sc.next();
                                    car_mangement carma=new car_mangement();
                                    if(confirm.equalsIgnoreCase("yes"))
                                    {

                                            boolean bb=carma.detailssave(pp);
                                            if(bb)
                                            {
                                            System.out.println("sucessssss");

                                            }else
                                            {
                                                  System.out.println("pool drive failed");
                                            }
```

```
                                }
                                else
                                {       System.out.println("order cancelled");}
                                }
                                else
                                {       System.out.println("picup or desti is wrong");}
                          }
                          else
                          {System.out.println("User id not found in  data bas");}
                    }
              });
              btnNewButton_1.setBounds(205, 304, 127, 39);
              contentPane.add(btnNewButton_1);

              JButton btnNewButton_2 = new JButton("POOL HISTORY");
              btnNewButton_2.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                          JOptionPane.showMessageDialog(null, "Login Successfully");
                          String username=textField.getText();
                          String password=textField.getText();
                          LoginDetails lg=new LoginDetails(username,password,"user");
                          car_mangement carmg=new car_mangement();
                          ArrayList<PoolDetails> ppd=carmg.viewalldetails(username);
                          System.out.println("use name  "+"Date   "+"  car no"+"   amount"+"
pickup"+" destination");
                          for(int i=0;i<ppd.size();i++)
                          {
                                System.out.println(ppd.get(i).getName()+"
"+ppd.get(i).getDate()+" "+ppd.get(i).getCarNo()+" "+ppd.get(i).getAmount()+"
"+ppd.get(i).getPickup()+" "+ppd.get(i).getDestination());
                          }
                          System.out.println("These are all transactions done");
                    }
              });
              btnNewButton_2.setBounds(376, 304, 125, 39);
              contentPane.add(btnNewButton_2);

              passwordField = new JPasswordField();
              passwordField.setBounds(188, 197, 251, 23);
              contentPane.add(passwordField);

              JLabel lblNewLabel = new JLabel("");
              lblNewLabel.setIcon(new ImageIcon("D:\\3rd sem\\java\\PROJECT\\WhatsApp Image
2021-11-03 at 9.59.32 AM (1).jpeg"));
              lblNewLabel.setBounds(10, 46, 526, 318);
              contentPane.add(lblNewLabel);

              txtGogreen = new JTextField();
              txtGogreen.setForeground(Color.GREEN);
              txtGogreen.setFont(new Font("Arial", Font.BOLD, 17));
              txtGogreen.setText("GO-GREEN");
              txtGogreen.setEditable(false);
              txtGogreen.setBounds(206, 22, 104, 29);
              contentPane.add(txtGogreen);
              txtGogreen.setColumns(10);

              JTextPane txtpnGoCarpooling = new JTextPane();
              txtpnGoCarpooling.setForeground(Color.GREEN);
              txtpnGoCarpooling.setFont(new Font("Arial", Font.BOLD, 17));
              txtpnGoCarpooling.setText("GO CAR-POOLING");
              txtpnGoCarpooling.setBounds(188, 363, 168, 27);
              contentPane.add(txtpnGoCarpooling);
        }
}
```

5)DB :Database connection

```
package ABC;

import java.io.*;
import java.util.*;
import java.sql.*;

public class DB {

	private static Connection con=null;
	private static Properties props=new Properties();
//	static String url = "jdbc:mysql://localhost:3306/puddi";
//	static String	user = "root";
//	static String password = "";
	static String url = "jdbc:mysql://localhost:3306/testing";
	static String user = "root";
	static String password = "Bharath@123";
	public static Connection getConnection() throws ClassNotFoundException, SQLException
	{
		try
		{
//		FileInputStream fis=null;
//			fis=new FileInputStream()

			Class.forName("com.mysql.cj.jdbc.Driver");
			con=DriverManager.getConnection(url,user,password);
			return con;
	}catch(Exception e)
		{
		System.out.println("db error"+e);
		}
		finally
		{

		}
		return con;

}
}
```

6)Car management:

```
package ABC;

import java.sql.*;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.*;

public class car_mangement {

	public boolean addUser(LoginDetails loginde) throws SQLException, ClassNotFoundException
	{
		boolean ab=false;
		try
		{
			Connection con=DB.getConnection();
			PreparedStatement ptstmt=con.prepareStatement("insert into
LoginDetails(username,password,role) values(?,?,?)");
		ptstmt.setString(1,loginde.getUsername());
		ptstmt.setString(2,loginde.getPassword());
		ptstmt.setString(3,loginde.getRole());
		int count=ptstmt.executeUpdate();
		if(count>0)
```

```
		{
			ab=true;
		}

		ptstmt.close();
		}
		catch(Exception e)
		{
		System.out.println("asdfghj"+e);
		}

		finally
		{

		}
		return ab;

	}

	public boolean getuserdetails(LoginDetails loginde)
	{
		boolean ab=false;
		Statement sts=null;
		ResultSet rs=null;
		try
		{
			Connection con=DB.getConnection();
//			PreparedStatement ptstmt=con.prepareStatement("insert into
LoginDetails(username,password,role) values(?,?,?)");
//		ptstmt.setString(1,loginde.getUsername());
//		ptstmt.setString(2,loginde.getPassword());
//		ptstmt.setString(3,loginde.getRole());
//		int count=ptstmt.executeUpdate();
			sts=con.createStatement();
			String query="select * from logindetails where
username='"+loginde.getUsername()+"' AND password='"+loginde.getPassword()+"' Order by
username";
			rs=sts.executeQuery(query);

		if(rs.next())
		{
			String uss=rs.getString("username");
			System.out.println("user alraedy found");
			return true;
		}
		rs.close();
		sts.close();
//		ptstmt.close();
		}
		catch(Exception e)
		{
		System.out.println("getdetails "+e);
		}

		finally
		{

		}
		return ab;
	}

	public boolean detailssave(PoolDetails pd)
	{
		boolean ab=false;
		try
```

```
		{
			Connection con=DB.getConnection();

		PreparedStatement ptstmt=con.prepareStatement("insert into
PoolDetails(name,date,carno,amount,pickup,destination) values(?,?,?,?,?,?)");
		ptstmt.setString(1,pd.getName());
		System.out.println("pd.getname"+pd.getName());
		ptstmt.setDate(2,java.sql.Date.valueOf(pd.getDate()));
		ptstmt.setString(3,pd.getCarNo());
		ptstmt.setInt(4,pd.getAmount());
		ptstmt.setString(5,pd.getPickup());
		ptstmt.setString(6,pd.getDestination());
		int count=ptstmt.executeUpdate();
		if(count>0)
		{
			ab=true;
		}

		ptstmt.close();
		}
		catch(Exception e)
		{
		System.out.println("details save "+e);
		}

		finally
		{

		}
		return ab;
	}
	public ArrayList<PoolDetails> viewalldetails(String userm)
	{
		Statement sts=null;
		ResultSet rs=null;
		ArrayList<PoolDetails> pdl=new ArrayList<>();

		try
		{
			System.out.println(userm);
			Connection con=DB.getConnection();
//			PreparedStatement ptstmt=con.prepareStatement("insert into
LoginDetails(username,password,role) values(?,?,?)");
//		ptstmt.setString(1,loginde.getUsername());
//		ptstmt.setString(2,loginde.getPassword());
//		ptstmt.setString(3,loginde.getRole());
//		int count=ptstmt.executeUpdate();
			sts=con.createStatement();
			String query="select * from pooldetails where name='"+userm+"'";
			rs=sts.executeQuery(query);
			System.out.println(userm);
			while(rs.next())
			{
				System.out.println("vales are presnet");
			pdl.add(new
PoolDetails(rs.getString("name"),rs.getDate("date").toLocalDate(),rs.getString("carno"),rs.getI
nt("amount"),rs.getString("pickup"),rs.getString("destination")));
			}
		rs.close();
		sts.close();
//		ptstmt.close();
		}
		catch(Exception e)
		{
		System.out.println("getdetails "+e);
```

```
			}
			System.out.println(userm);
			return pdl;
		}
}
```

7)Login details:

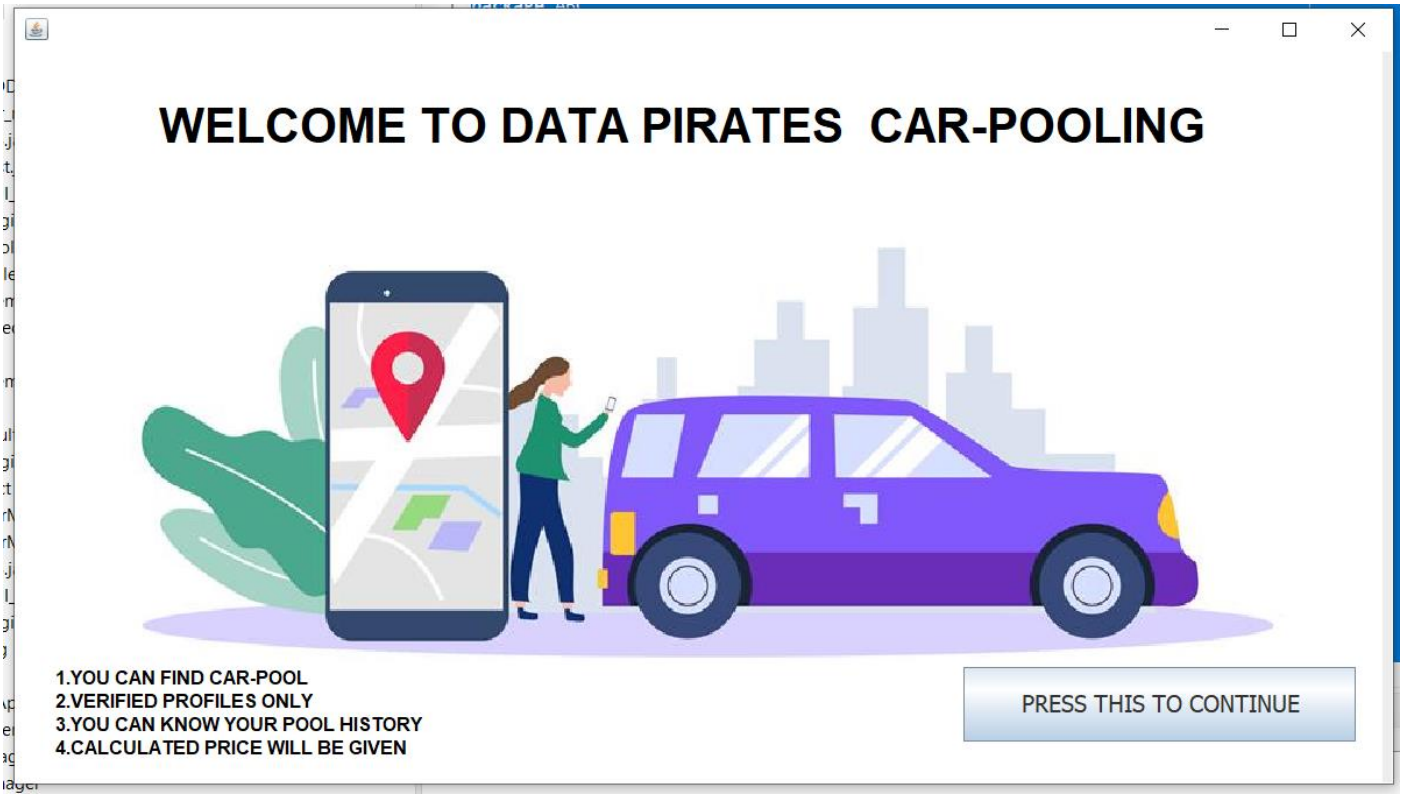
```
package ABC;
public class LoginDetails {


	private String username;
	private String password;
	private String role;

	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public String getRole() {
		return role;
	}
	public void setRole(String role) {
		this.role = role;
	}
	public LoginDetails(String username, String password, String role) {
		super();
		this.username = username;
		this.password = password;
		this.role = role;
	}

}
```

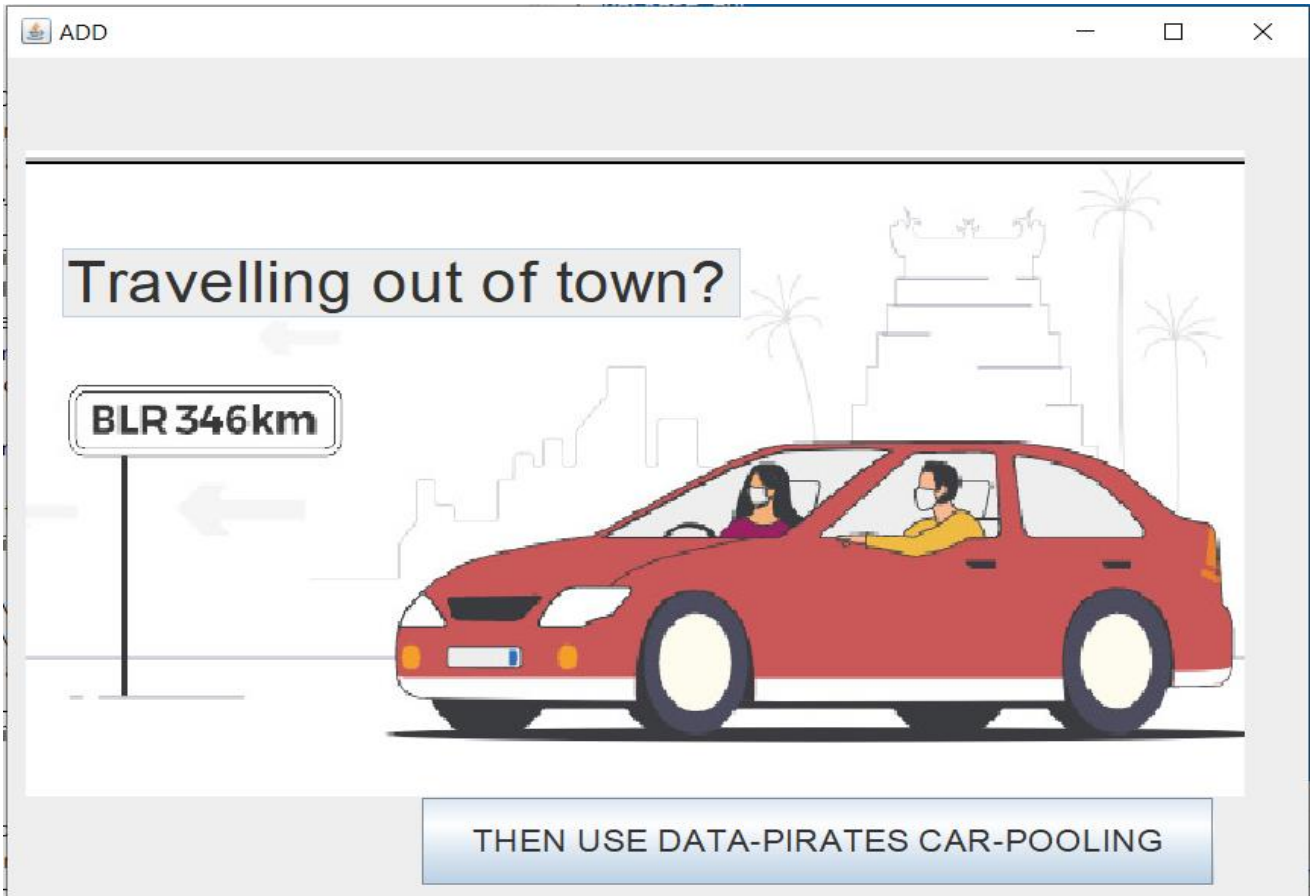
8)Pool details:

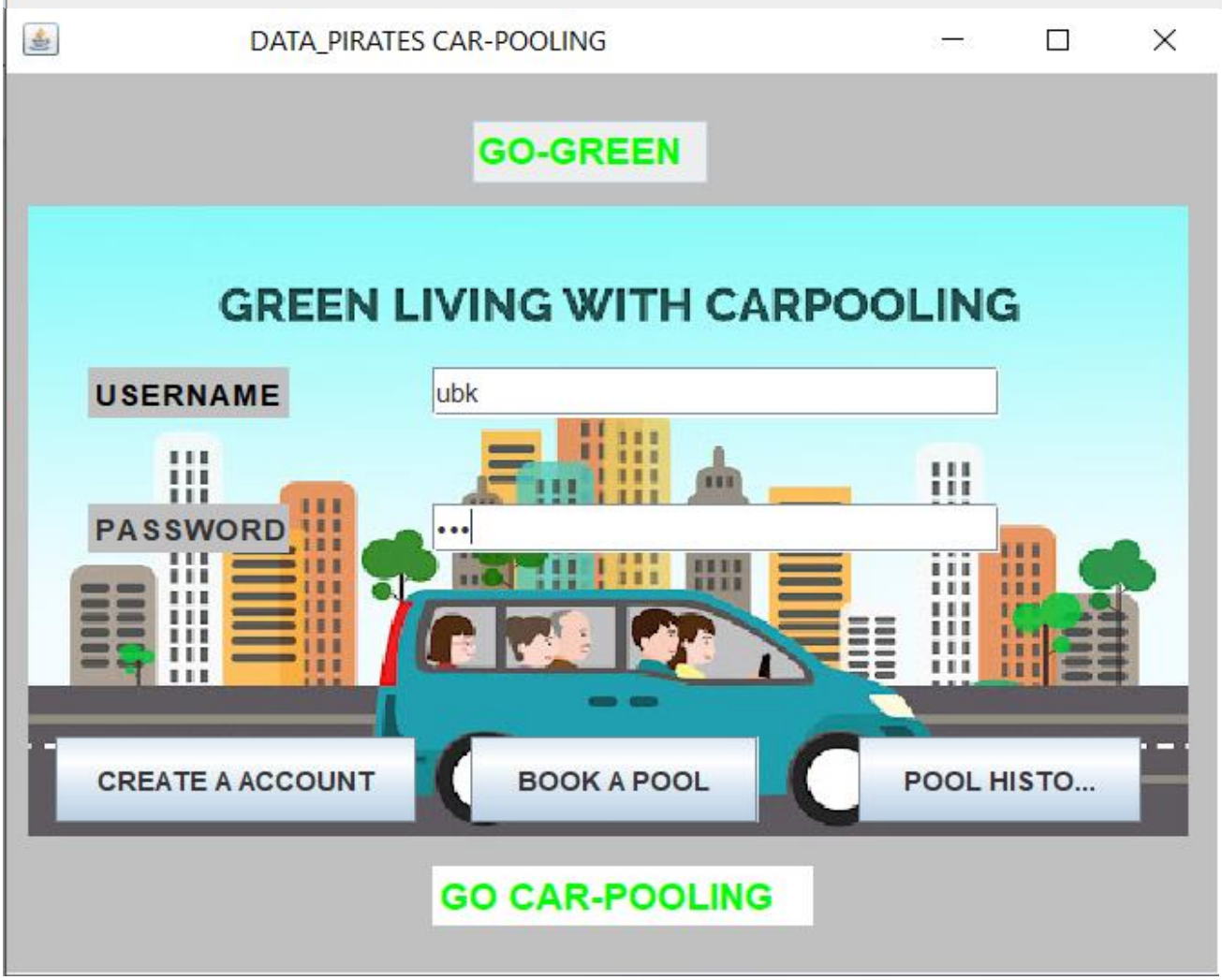
```
package ABC;
import java.util.*;
import java.time.*;
public class PoolDetails {

	String name;
	LocalDate date;
	String carNo;
	int Amount;
	String pickup;
	String destination;
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public LocalDate getDate() {
```

```
		return date;
	}
	public void setDate(LocalDate date) {
		this.date = date;
	}
	public String getCarNo() {
		return carNo;
	}
	public void setCarNo(String carNO) {
		this.carNo = carNO;
	}
	public int getAmount() {
		return Amount;
	}
	public void setAmount(int amount) {
		Amount = amount;
	}
	public String getPickup() {
		return pickup;
	}
	public void setPickup(String pickup) {
		this.pickup = pickup;
	}
	public String getDestination() {
		return destination;
	}
	public void setDestination(String destination) {
		this.destination = destination;
	}
	public PoolDetails(String name, LocalDate date, String carNo, int amount, String pickup,
String destination) {
		super();
		this.name = name;
		this.date = date;
		this.carNo = carNo;
		Amount = amount;
		this.pickup = pickup;
		this.destination = destination;
	}



}
```

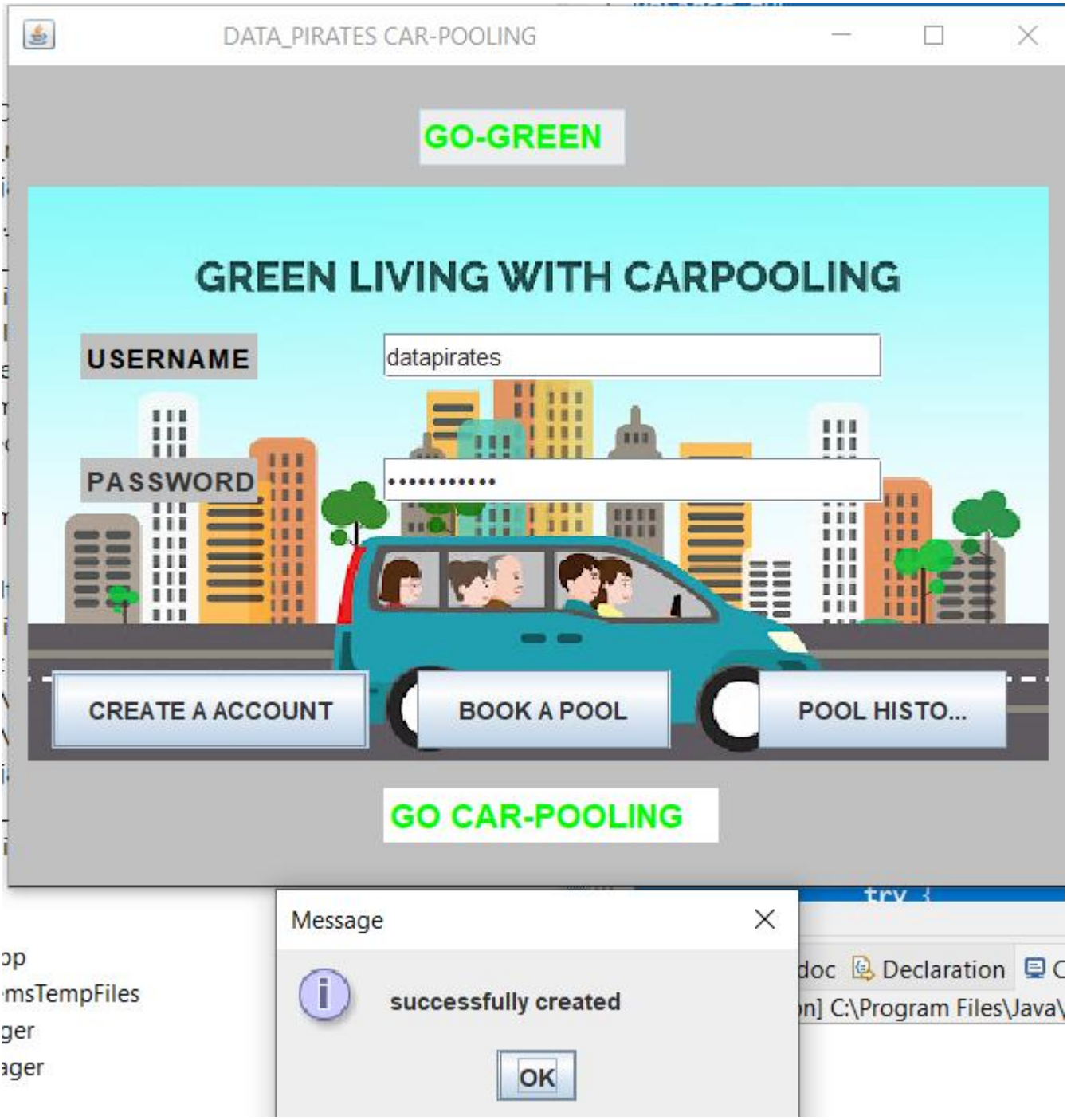
OUTPUT


WELCOME TO DATA PIRATES CAR-POOLING
1.YOU CAN FIND CAR-POOL
2.VERIFIED PROFILES ONLY
3.YOU CAN KNOW YOUR POOL HISTORY
4.CALCULATED PRICE WILL BE GIVEN
PRESS THIS TO CONTINUE

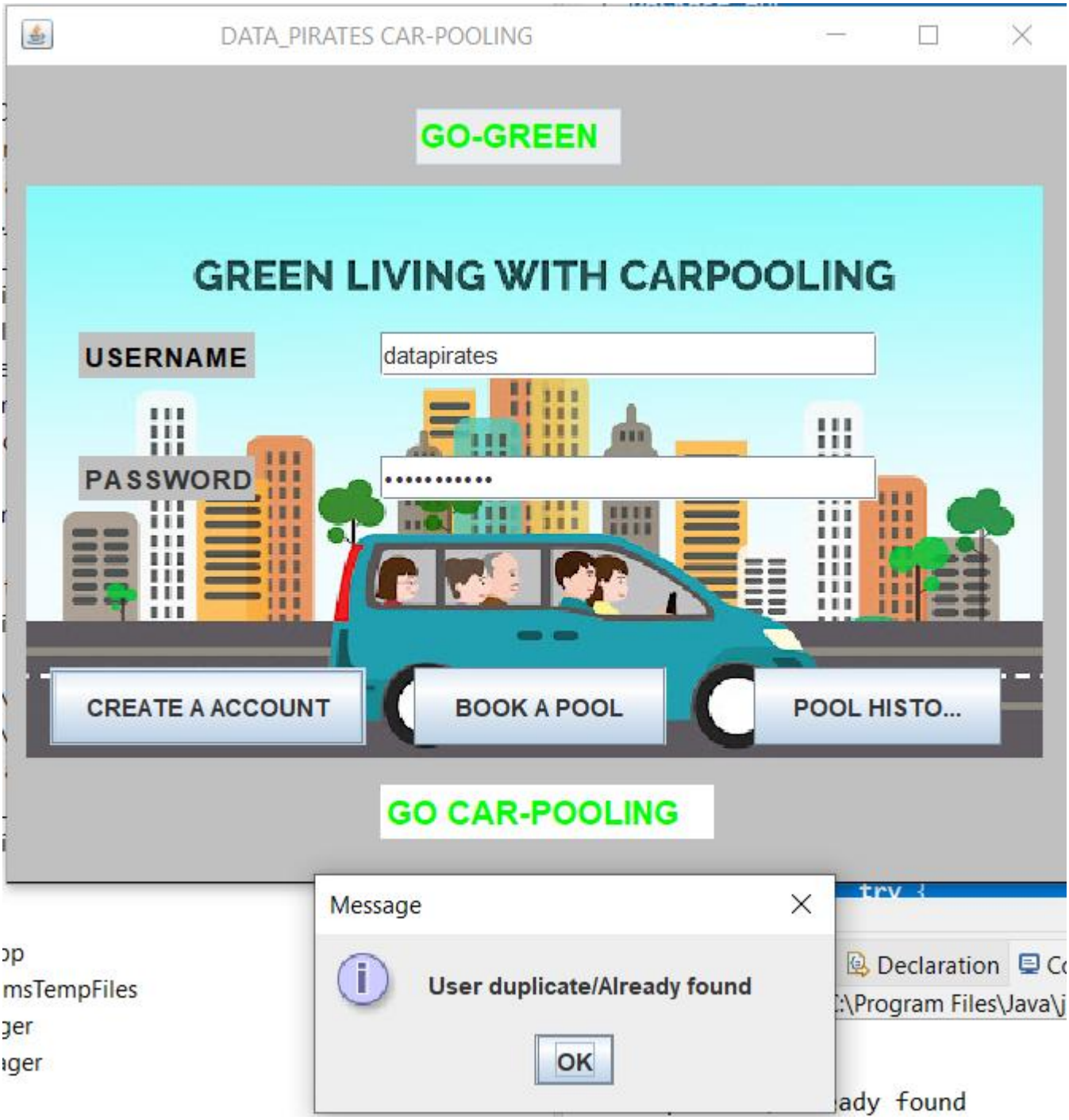




From here we have 3 output with create account,pool history,creatating the pool.

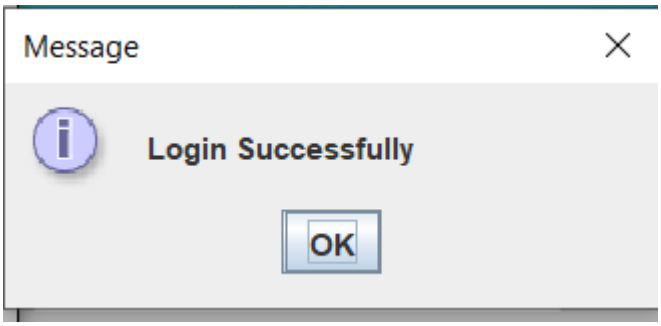


It can find the user duplication if we create with same username:



Here I tried to create the account with same user name.

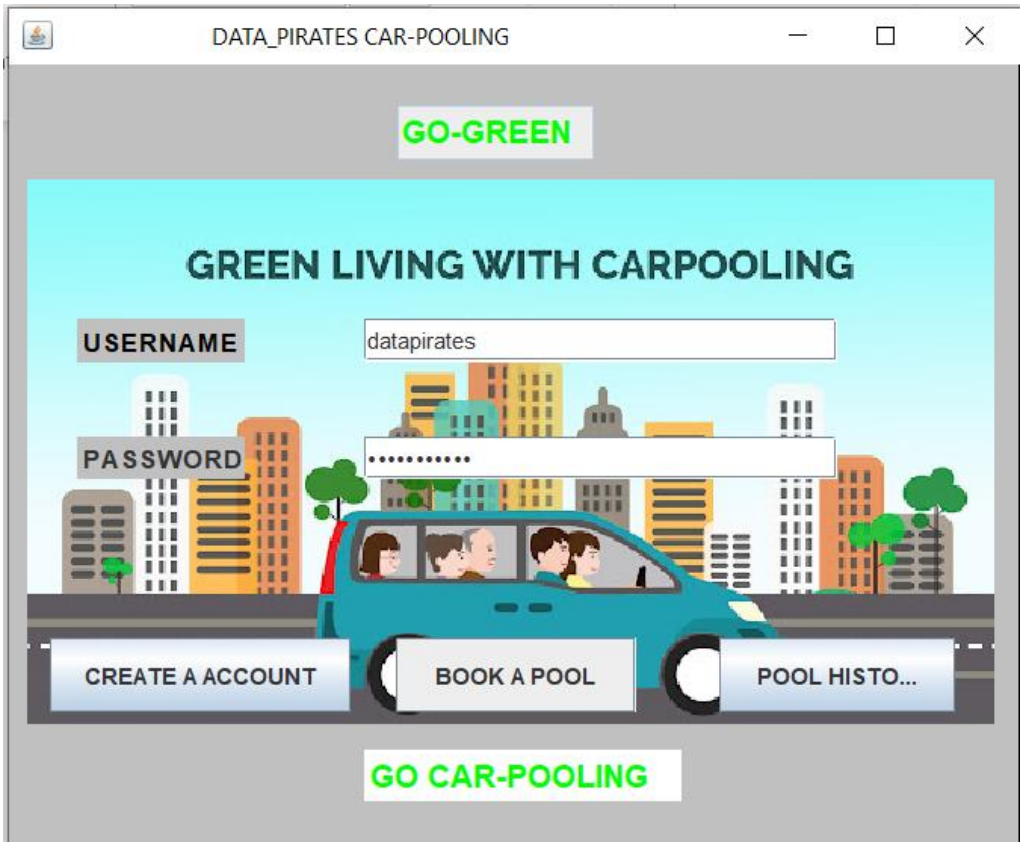
Now booking the pool:



After clicking on book a pool it shows message

```
user alraedy found
User already exsists
hyderabad
kamareddy
nizambad
vizag
vijayavada
Enter pickup
vizag
Enter destination
nizambad
Enter the vechile
ap 1234
ts 1234
re 12
yu 12
kmr 12
kmr12
1
lg.getUsernamedatapirates ASDFG+300
enter yes to save
yes
pd.getnamedatapirates
sucesssssss
```

Now we can check for the User pool history



```
----------
datapirates
datapirates
vales are presnet
datapirates
use name  Date     car no   amount  pickup destination
datapirates 2021-11-03 kmr12 300 vizag nizambad
These are all transactions done
```

Contents in the Data base:

Source code in Data Base:

show databases;

create database testing;

use testing;

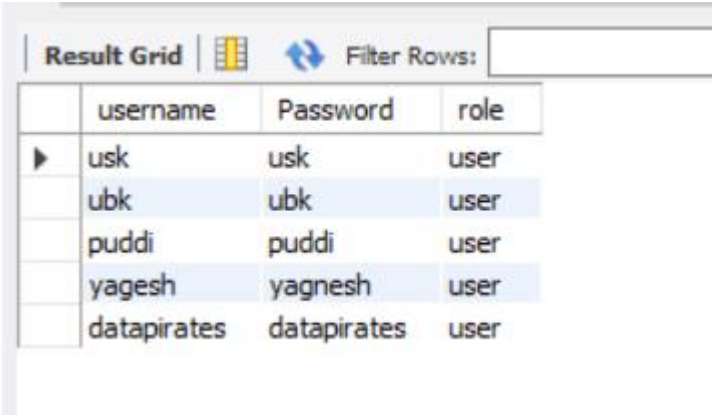
create table logindetails(username varchar(20),Password varchar(20),role varchar(20));

create table pooldetails(name varchar(20),date Date, carno varchar(20), amount int, pickup varchar(20),destination varchar(20));

select * from logindetails;

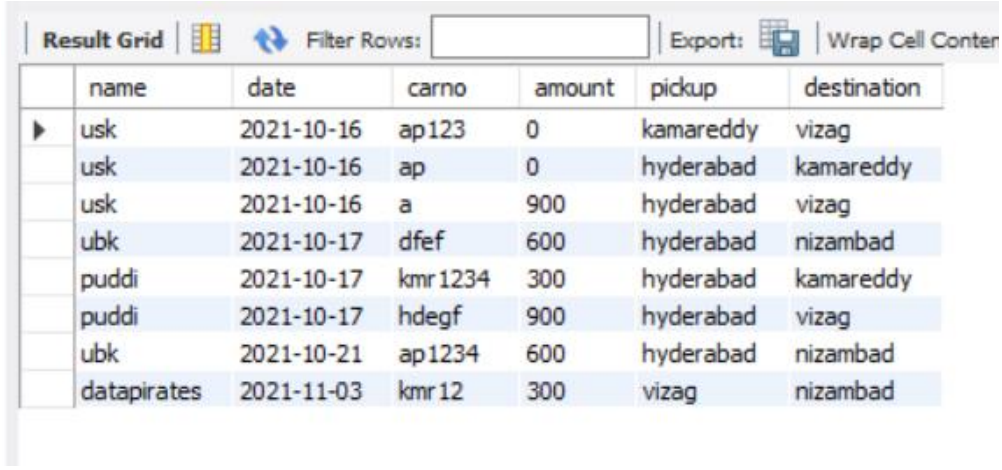
select * from pooldetails;

OUTPUT in data base:



Result Grid | Filter Rows:

username	Password	role
usk	usk	user
ubk	ubk	user
puddi	puddi	user
yagesh	yagnesh	user
datapirates	datapirates	user



Result Grid | Filter Rows: | Export: | Wrap Cell Conter

name	date	carno	amount	pickup	destination
usk	2021-10-16	ap123	0	kamareddy	vizag
usk	2021-10-16	ap	0	hyderabad	kamareddy
usk	2021-10-16	a	900	hyderabad	vizag
ubk	2021-10-17	dfef	600	hyderabad	nizambad
puddi	2021-10-17	kmr1234	300	hyderabad	kamareddy
puddi	2021-10-17	hdegf	900	hyderabad	vizag
ubk	2021-10-21	ap1234	600	hyderabad	nizambad
datapirates	2021-11-03	kmr12	300	vizag	nizambad

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	date	carno	amount	pickup	destination
usk	2021-10-16	ap123	0	kamareddy	vizag
usk	2021-10-16	ap	0	hyderabad	kamareddy
usk	2021-10-16	a	900	hyderabad	vizag
ubk	2021-10-17	dfef	600	hyderabad	nizambad
puddi	2021-10-17	kmr1234	300	hyderabad	kamareddy
puddi	2021-10-17	hdegf	900	hyderabad	vizag
ubk	2021-10-21	ap1234	600	hyderabad	nizambad
datapirates	2021-11-03	kmr12	300	vizag	nizambad

Machine -Learning Algorithm approach

Many algorithms were used in Carpooling systems to calculate the best route, shortest path and find potential passengers. Carpooling systems use algorithms and data mining techniques to allow both passengers and drivers to find a convenient trip route and to support a billing system.

However, one of the most popular machine learning algorithms is the k-Nearest Neighbor algorithm (k-NN). Generally speaking, k-NN is an efficient, simple and easy-to-implement supervised machine learning algorithm that can be used to solve different classification and regression problems.

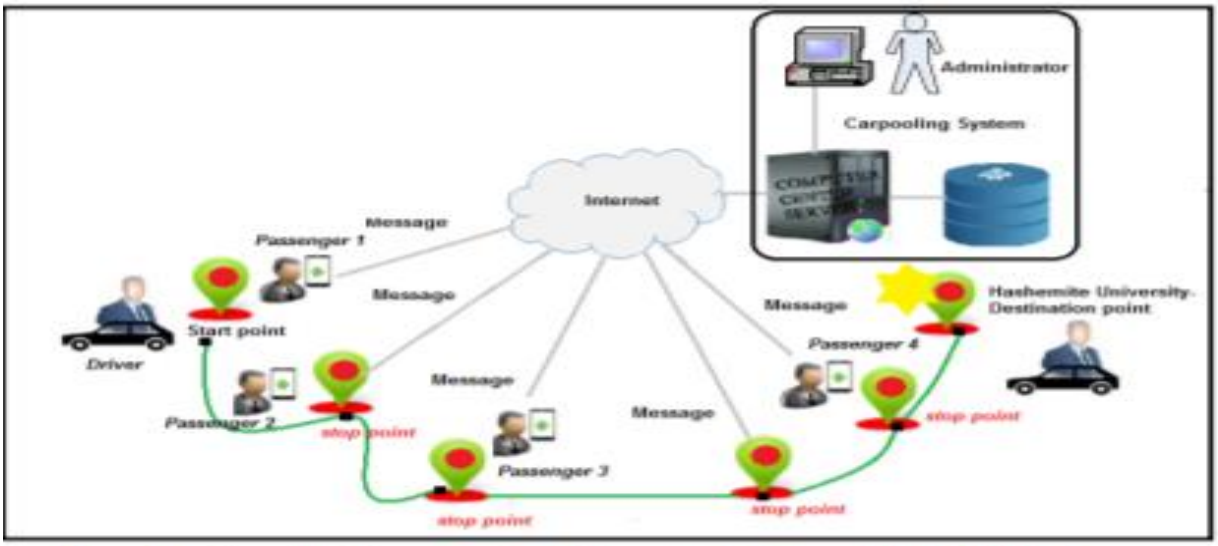


Fig. 1. The Architecture of the Proposed Carpooling Mobile Apps.

The steps of the proposed system are illustrated below based on the system architecture:

1) Initially, the legal drivers and passengers apply for the carpooling system by sending a request to join the carpooling mobile application.

2) Once the administrator assures the eligibility, especially for the drivers, the accounts are created and the details are provided to students to start using the carpooling mobile application.

3) With respect to driver, the driver log in to the mobile app and determine the start-up time (time to be in the nearest stop point), end-time (time to leave the university) and the seat availability.

4) With respect to passenger, the passenger sends a request that determines the pick-up time, stop point (meeting station) and the preferred gender of the driver (optional).

5) Once the passenger request arrives to the carpooling system, the k-NN algorithm is applied on the available drivers to provide the passenger with the "best"- matched drivers (i.e. top five drivers). The k value of the k-NN algorithm reflects the available active drivers. In our case, the value of the best available driver is set to three drivers (k=3)1 . Typically, the value is set by the system administrator. The passenger is informed with the set of the best-matched drivers obtained by the k-NN. The set of the best-matched drivers are shown as coloured cars on the map (pink car stands for female drivers and blue cars stands for male drivers). Then he/she selects the driver/s so that the request is forwarded to them.

6) Finally, the system sends a confirmation note contains the driver details, pick-up time and stop point. 7) The system provides the drivers with the suitable passenger requests fit the driver start-up and end-time. Also, the seat availability is considered. The driver can check the details of the passenger easily and accept the desired requests.

Finally, the carpooling system will generate a small notification (report) with the stop points along with the number of passengers should be collected from each.

Thus, suppose we are given two points P1 and P2 determined by their latitudes and longitudes values: P1(α1, β1) and P2 (α2, β2). The Euclidean distance dist in Cartesian coordinates is computed as:

$$\mathbf{dist}^2 = (\mathbf{x_1} - \mathbf{x_2})^2 + (\mathbf{y_1} - \mathbf{y_2})^2 + (\mathbf{z_1} - \mathbf{z_2})^2$$

Where

$$x = R \cos\alpha \cos\beta$$

$$y = R \cos\alpha \sin\beta$$

$$z = R \sin\beta$$

Algorithm 1: k-NN for carpooling system (P, DR)

{
Input:
P:passenger request details that contains the coordinate, pick-up time (t) and gender (g)
DR: the set of active registered drivers where each driver d_i contains t_i: pick-up time and g_i: gender (if determined)
Output:
S: The set of three nearest matched-drivers
1. S= θ; L= θ;
2. For each $d_i \in DR$
3. If d_i (t_i, g_i) == P(t, g) then
4. Calculate $dist_i(x_i, y_i)$: the distance between d_i and P;
5. L = ADD (d_i, $dist_i$)
6. Endif
10. End for
S=arrange L in descending order with respect to $dist_i$.
Return S
}

Architecture Overview and Design

The architecture of the proposed system comprises of the following components

1) Users: Three types of users are involved in the proposed system: driver, passenger and system administrator. Passenger – a student requesting carpooling Services while Driver– the student providing carpooling Services using his own car. Typically, drivers and passengers are students who are registered in the current academic semester. However, the system administrator is a staff member of the Computer Centre (CC) of the Hashemite University.
2) 2) Carpooling System: A web based system developed by the computer centre of the Hashemite University to manage, support and control all administration tasks. It is the most substantial component and considered as the heart of our proposed mobile application. The system encompasses of: a) System administrator – a staff member who is responsible for controlling and managing the system. The main administration tasks include (but not limited to): (i) accept new drivers' requests, (ii) accept new passengers' requests, (iii) give the students (drivers and passengers) their account details to start using the system and (iv) generate reports about the registered drivers and all related information about their cars (i.e., car brand, colour, and seat availability). b) Database – contains the actual database for all employees and students of the Hashemite University. c) Computer Centre Web Server (CC server) – this type of server contains the actual desired database of the registered and activated users necessary to operate the mobile application. Registered users are eligible drivers and/or passenger, but the activated users are those drivers who are currently ready to start a trip and commute passengers.

For the machine learning algorithm we can implement with the K-NN algorithm for the classification and regression for the car management and drivers.

We can also use the Clustering for carpooling (Here).

For presentation files (Here).

Result: Our Team Data Pirates here to declare that our project and information are genuine and successfully completed the project.

Team

DATA-PIRATES