

CAR POOLING APPLICATION

High Level Design

Document Control :

| CAR POOLING APPLICATION | | | | | | | |
|----------------------------------|---------|--|------------------------------|--|--------------------|--|--|
| Guided by- BISWA P DAS | | | | | | | |
| Date | Version | Author | Brief Description of Changes | | Approver Signature | | |
| October 20 2022 | 1.0 | Group 8: Chaganti Amrutha Varshini, Srikrupa S, Chama Divya Reddy, Abarana P, Abitha A | | | | | |
| | | | | | | | |
| | | | | | | | |

Introduction

1 INTRODUCTION:

Car pooling is an application that helps reduce the cost of repetitive or long distance trips by sharing cars, sharing rental costs, or paying the primary owner of the car. Transporters will be able to drive to their destination while sharing the vehicle and the costs incurred, and allow commuters to collaborate with transporters and plan their trip using this app after registering with them. People will be able to share the cost and not worry about being late while they can make new connections and register.

1.1 Intended Audience:

Audience could be a any responsible citizen who wish to share the ride or accept the ride.

1.2 Project Purpose:

The purpose of this project is to implement a File Handling programming-based Car Pooling application that will allow the multiple transporters to register and share their rides and expenses and also the commuter to register to collaborate with the transporter.

It provides a general architecture for car pooling application and anyone can use it as a basis for sharing rides or querying the service.

The application is written in a middle level language called C. The application is divided into two parts, transporter and commuter, the transporter in this application will provide information such as its driver's license number, personal information, a source and destination point, its vehicle type, the number of commuters it can carry, etc. and add and process it to the file. The commuters provide information such as Aadhar number and personal data.

1.3 Key Project Objectives:

- a. Allow the Transporter to add their data and register.
- b. The transporter is allowed to change his personal details.
- c. Allows the Commuter to add their details and register.
- d. Allows a registered commuter to enquire about the service after registration.
- e. Allows the Commuter to change his personal details.
- f. Allows admin to add, modify, view or delete the transporter data.
- g. Allows the admin to add, modify, view or delete the commuter data.
- h. Generates report for vehicle details.
- i. Generates report for showing busiest routes.

1.4 Project Scope :

The goal of this project is to construct the Car Pooling application. It is a very effective method of reducing pollution and car traffic in cities. It also provides an environmentally beneficial mode of transportation.

It also gives you the opportunity to meet new individuals. Because of the delays created by public transportation and the comforts given by private vehicles, most individuals today choose to travel by private vehicle. Pre-registration ensures security by allowing only verified individuals to enter the vehicle, allowing trust to be formed.

People who have registered do not have to use their private vehicle as a result, the car pooling system will be successful in reducing pollution and traffic.

1.5 Functional Overview

The following functions are included in the program:

Main Menu

- 1) The Transporter Module
- 2) The Commuter Module
- 3) The Admin Module

The Transporter Module

1) registration_process(): The person who wants to provide the transport service, is required to register on the system with his Driving License no(dlno) and personal details.

Details required for register:

- Name of Transporter
- Aadhar Number
- Driving License
- source point
- destination point
- Type of vehicle
- No_of_commutors
- ac_or_non_ac
- charge_per_km

Once the registration is done successfully the transporter is provided with a unique-id which is auto generated by the system.

2) edit_transporter_details(): After registration is done the transporter is also allowed to change the personal details.

The Commuter Module

- 1) commuter_registration_process(): The commuter need to register using his

unique aadhar no and other personal details.
details register :

- Name of Commuter
- Gender
- Aadhar Number
- Contact details

he will be given a confirmation for the same.

2) registered_commuter_service(): After successful registration :

- commuter need to enter aadhar number to avail the service
- point of boarding
- Destination
- Display the vehicle available on that route
- on selection of vehicle the charges are displayed
- To confirm the commuter need to pay 10% of total amount

3)commuter_edit_details(): The commuter is also allowed to change his personal details

- In case commuter wants to discontinue he need to request the admin and the admin will delete his record.

The Admin Module

1) maintain_transporter_data(): The admin has the option to add, edit, delete or view the transporter data.

2) maintain_commuter_data(): The admin can add, modify, delete or view the commuter data.

3) report_on_vehicle(): report containing details of all vehicles for all routes along with respective transporter name is available as an option.

4) report_on_busiest_route(): A report to show the details of the busiest route in terms of No of commuter registered for the same is also available.

2 Design Overview:

Instant Chatters comprises of the following modules:

| | |
|--------------------|--|
| Name of the Module | Main Menu |
| Handled by | Chaganti Amrutha Varshini |
| Description | It consists of the main menu and three sub menus which will provide options to the user. |

| | |
|--------------------|---|
| Name of the Module | Validations |
| Handled by | Chama Divya Reddy |
| Description | It consists of all the functions that validates all the inputs. |

| | |
|--------------------|---|
| Name of the Module | Transporter_Register |
| Handled by | Srikrupa S |
| Description | To register the new transporter using a unique Driver License No., personal and other details |

| | |
|--------------------|---|
| Name of the Module | Transporter_edit_details |
| Handled by | Srikrupa S |
| Description | To change the transporter's personal details. |

| | |
|--------------------|--|
| Name of the Module | Commuter_add_data |
| Handled by | Abarana P |
| Description | To register new Commuter and to avail the services |

| | |
|--------------------|---------------------------------------|
| Name of the Module | Commuter_edit_data |
| Handled by | Abarana P |
| Description | To edit the Commuter personal details |

| | |
|--------------------|--|
| Name of the Module | Maintain Transporter data |
| Handled by | Abitha A |
| Description | To add, modify, view & delete Transporter data |

| | |
|--------------------|---|
| Name of the Module | Maintain Commuter data |
| Handled by | Abitha A |
| Description | To add, modify, view & delete Commuter data |

| | |
|--------------------|--|
| Name of the Module | Reports of Vehicle and Busiest routes |
| Handled by | Chaganti Amrutha Varshini |
| Description | <p>A report containing details of all vehicles for all routes along with respective transporter name should be available.</p> <p>A report to show the details of the busiest route in terms of. No of commuters registered for the same is also available.</p> |

2.1 Design Objectives:

1. Add Transporters and Commuters to the records.
2. Avail the services for Commuter.
3. Updating the details of Transporter and Commuter.
4. Displays all the records of Transporter and Commuter.
5. Modify/Update the Transporter and Commuter records.
6. Deleting the Transporter and Commuter records.

2.2 Design Alternative:

We have used linked list structure to operate on the data and finally store the data in a flat file. Driving License no, personal details, route, vehicle type, no. of commuters it can carry, ac/non ac etc of Transporter and Aadhar, personal details, source and destination points.

2.2.1 User Interface Paradigms:

User is given an interface to add a new record by registering to avail the services and admin is given an interface to add a record, update a record, view the details and delete a record.

2.2.2 Error Detection / Exceptional Handling:

User should first enter the details according to the condition and if the entered detail is not according to the condition specified, sometimes it displays the message that is entered and sometimes it returns with an error.

2.2.3 Performance:

The system will work on the terminal. The performance depends on the hardware component of the user's system.

2.2.4 Validation:

- Transporter License no. should not be blank and duplication is not allowed and characters aren't allowed in the License no.
- In case of integer validation, if the entered License no. is not Integer it displays the message ID must contain Integer only and should not contain more than 6 digits.
- We check for the validity of the name; it should not contain more than 15 characters and Transporter name should not be empty and only alphabets. Similarly for all personal details validations are done.
- The routes must be from the list shown and source and destination should be different.
- The distance provided by Transporter cannot be empty and must be minimum 5 km to avail the service.
- Charge should not be more than 5 Rupees per km.
- Commuter Aadhar number must be 8 digits and not contain any character other than digit.

3.Environment Description:

3.1 Language Support: English

3.2 User Desktop Requirements:

- a. 64-bit processor, 1 GHz or faster
- b. At least 2 GB free hard drive space
- c. At least 1 GB RAM

3.3 Integration Requirements:

- 1. Language: C
- 2. Tools: Splint, Valgrind, Makefile, Vi Editor
- 3. Compiler: gcc
- 4. Environment: Linux

3.4 Configuration:

3.4.1: Operating System: Linux on cloud