



# INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume2, Issue5)

Available online at [www.ijarnd.com](http://www.ijarnd.com)

## Car Pooling Using Global Position System

Justin K. Williams, A. Benazir Begum

<sup>1,2</sup>Akshaya College of Engineering & Technology, Coimbatore.

[benazirbeguma@acetcb.edu.in](mailto:benazirbeguma@acetcb.edu.in), [justin.sahara@gmail.com](mailto:justin.sahara@gmail.com)

### INTRODUCTION

The Carpooling is not a new concept in the field of car sharing. Decades ago, when there were oil crises in Europe, people were encouraged to share their vehicles. The carpooling application will try to reduce the cost of the journey for travelers who commute to work daily which will not only save their money but this will also reduce the usage of most important non-renewable resource we have i.e. fuel, which is declining at a rapid pace.

Our application will have two kinds of users, Firstly, the travelers (drivers) who use their own vehicle (car only for now) to go to work daily and in need of some fellow passengers on their way to accompany them. Some owners might want someone to accompany them for the entire journey ie from source to destination. These drivers can use our application to create their profiles as driver under “offer a ride” section and mention their route to work they are expecting from passengers, car they have (hatchback, sedan, SUV), number of fellow passengers they need and details of their time when leave from home for work as well as when they leave from office.

The drivers or the person pooling the car for that day chooses from the available and willing sharers by the use of the algorithms implemented which could reduce the cost and the distance traveled but at the same time fills the capacity of the vehicle. Secondly, the travelers (passengers) who also travel to work daily but they either use public transport or use other modes of transportation which take more time to reach their destination due to which they feel exhausted at the end of the day. These passengers can use this application to find drivers who have the destination same as theirs and allow them to share their location and their destination. Passengers will be asked to register their profile by giving details like car model, vehicle number, capacity along with his bio data so that our application will be able to filter the owners and sharers and form a group accordingly to their specified expected fare and of course, source and destination.

Our project is an effort to make the drivers and passengers build a bond of trust for their own sake and do a bit of favor to the environment by pooling the cars on a daily basis when they go to work by being an educated and responsible citizen.

The application will ask the user to select his/her role i.e. driver or passenger for the day. After logging into the app, the role has to be selected. The passenger and the driver will enter his location and his/her destination. The owner or the person pooling the car will select from the list of sharers willing to share the car using the heuristic and greedy algorithm.

Carpooling application is deployed on mobile phones as apps work faster than websites. A well-designed mobile app can perform actions much quicker than a mobile website and Mobile apps have the advantage of utilizing features of a mobile device like a camera, contact list, GPS, phone calls, accelerometer, compass, etc. It is also easy for sending notifications

### EXISTING SYSTEM

The carpooling problem was been undertaken by various projects and have given various effective solutions to the problem.

*“Implementation of Dynamic Carpooling System on Android Platform”, February 2015*

*Yuvraj Nalawade, Vijay Waghmare, Prasmith Waghmare*

The functionality of the system is represented using following the 5-tuple format of Finite State Machine.

$(\Sigma, S, s_0, \delta, F)$

- a.  $\Sigma$  is the set of input alphabets (a finite, non-empty set of symbols).
- b.  $S$  is a finite, non-empty set of states.
- c.  $s_0$  is an initial state, an element of  $S$ .
- d.  $\delta$  is the state-transition function.
- e.  $F$  is the set of final states, is a subset of  $S$ .
- f.  $\Sigma =$  username, password, login-ID, source, destination, available seats.

The method they use to select pools of users can be described as a 3-steps process:

1. **Data collection:** Gather the following information:

- A list of users with informations such as their geographical location and whether or not they have a car.
- The location of the common destination of these users.
- The map of routes with their performances.

2. **Clustering:** Cluster users using a similarity evaluation:

They build a similarity matrix using the Pearson correlation coefficient on the distances between users. Then they use this similarity matrix with the complete linkage clustering heuristic to build clusters. These clusters are made such as they correspond to the user pools that will share a ride.

3. **Routing:** In this last step, for each user's pool, a "current provider" is selected. He is the owner of the currently shared car. Then a Travelling Salesman Problem is solved for the users of the group to find the best route to pick them all up.

This basic process is then refined by tweaking some parameters. Here are some interesting things to add or modify:

- Usage of a shortest or quickest strategy for the last step.
- Usage of a different clustering heuristic.
- Modification of the similarity index used for clustering to introduce factors taking other Attributes of the users into accounts, such as their age or hobbies. This allows forming more homogeneous groups.
- Set a threshold level for the similarity in the clustering heuristic. This way the vehicles are not always filled to their maximum capacity if other users are too far away.

## **PROPOSED SYSTEM**

The users will have our developed carpooling android application installed in their android smartphones. The carpool process will be initiated by registering the users. Then users will be able to create and share rides. These ride creation and ride browsing processes involve following activities. The application is based on Android SDK. This will be used by the Cognizant associates who are willing to share a ride. The mobile user has to be registered in this application using their Cognizant credentials to use this application. Once he/she can log into the application either as a vehicle owner or as a share rider. If the user logged in as a vehicle owner, he/she will be presented with the notification screen where he/she can view all the requests sent by various ride shares and the owner can respond back to them from there. If the user logged in as a ride share, he/she will be presented with the search screen where the user can search the available vehicle owners based on the location. The user can send a request to the from the search results.

## **DESCRIPTION OF A SYSTEM**

Carpooling system basically depends on two things, the driver who is going to make his vehicle available to get pooled as well his route information and passengers who are willing to get a ride from the available cars.

The Carpooling system the user has to be logged in to use the services provided by us. The user can either create a pool or can get a ride or both according to his needs. The driver has to create pool first and give out all the necessary details asked by the application. The passenger also has to fill information asked by the tool like his route information.

Every detail these users will mention will be stored in our database. The user will use GUI provided by the system to fill in all the details which are connected to our SQLite database. All the entries will be saved in particular columns now, after getting details from both the users, whenever a user (passenger) will click on get a ride option, the user will be diverted to the Activity page of the application where the user will be shown drivers according to his source and destination details. The system will use List View to show suggestions of drivers. Carpooling system will retrieve relevant information from a database according to the user's route details.

One of the features of our application is filtering which works here. Drivers will always be shown to the user according to the users rating for that driver on that particular route which will help passenger to choose from the most dependent drivers. A woman who is willing to get a ride can choose “Female drivers” as filtering options which will not only drive more woman to use this system but also make it a more secure and reliable system for females.

Now that the passengers have seen all the drivers using that route as per their requirement, the passenger can send a “Request” to the preferred driver for the ride. The driver, on the other hand, can accept or decline a request if he senses something wrong and can only start communication with a passenger if he accepts the request with the help of either emailed we have taken or phone number. Only these three details, Name, Phone number and email-id will be shown to the passenger before he can send a request but will only be able to communicate with the driver if he accepts the request.

After all this necessary and important steps, both driver and passenger can decide on common meeting points and time to meet to start their journey together.

Another feature which we are going to provide is location based services using Google Maps API via `google_play_services_lib` from Google where the passenger will be able to track the driver so that he can reduce waiting time at their meeting spot and will be safe too.

In the previous section, the existing solutions for carpooling were discussed. In this section we propose our design studying the previous solution, understanding the drawbacks of the previous solution and proposing some issues& feature which would help to build a more user-friendly and secure carpooling solution. The main features of our application listed below.

### **Real-time Design**

In addition to the planned or scheduled carpooling system, some features are provided wherein users can select if they are more interested in picking random passengers and not bind him with few passengers. Whenever a user leaves his home or workplace, he can leave a notification for the passengers on his route (which will be taken from GPS) and if anyone wants to join him, they can just send their location to them and can get picked.

### **Security Features**

Safety facility can be guaranteed to the passengers as only the people bound for the same organisation pool the cars between them.

In this Project, the user will be able to interact with the driver and book his seat easily. The user has to download this app and register for carpooling. They will enter the source, destination, and other details. These details will be stored in the database to maintain records. If he/she is a passenger waiting to get pooled, then he has to enter the location and wait for the owner to pick him up. The users will have our developed carpooling android application installed on their android smartphones. The carpool process will be initiated by registering the users. Based on the generic workflow of the activities for the SPCPS to compose the requirements are

1. Driver (Faculty or Student)
2. Passenger (Faculty or Student)

The main output of the system specification that meets the user needs, aiming to increase the average rate of car occupancy is:

- Information in Real Time.
- Best possible route.
- Ad-hoc trips.
- User and driver profiles.

Then users will be able to create and share rides. These ride creation and ride browsing processes involve following activities.

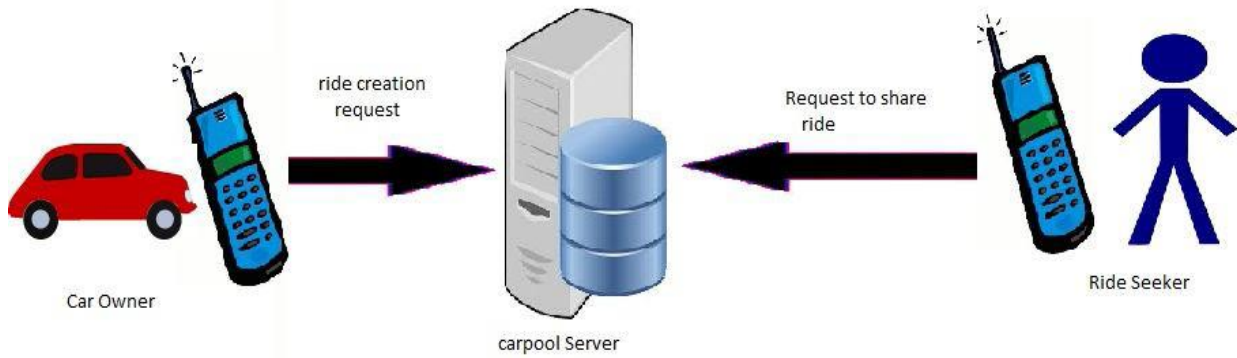


Fig. Initial carpool initiation

Figure 2.4: Initial Carpool Initiation

#### A. Creating Ride

- The car owner will enter the source, destination, starting time and available seats as input to the android application.
- This ride creation request will be transferred to the carpool server.
- Now the server will check for the existence of route between entered source and destination and will validate the other input information.
- Now ride is created and ride seekers are able to search and browse this ride.

#### B. Browsing ride

- Ride seeker will enter the source, destination, starting time as input to the android application to search and browse for rides.
- This ride browser information will be transferred to carpool server.
- Now the server will validate all inputs specified by the user.
- After validation, the server will show available rides to the ride seeker. Ride seeker can send a request to anyone of these ride creators.

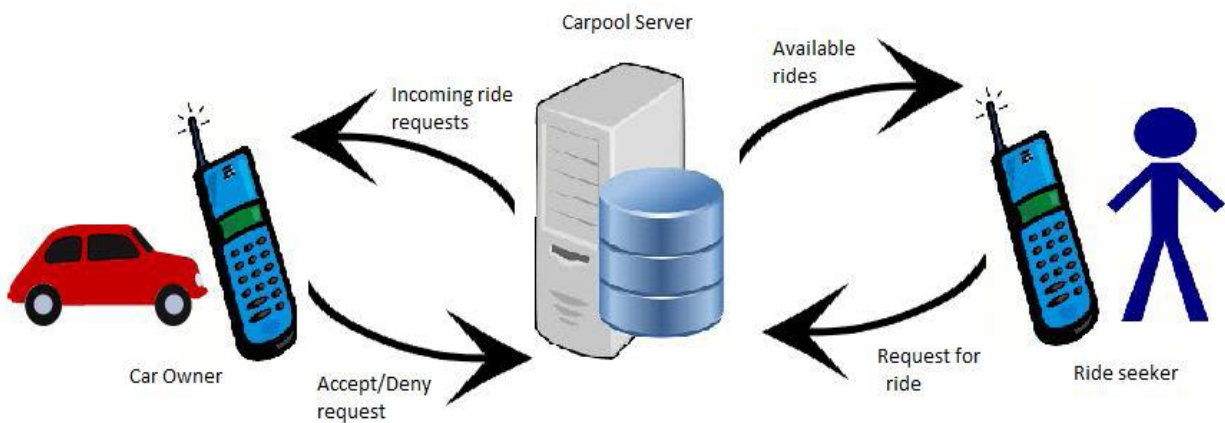
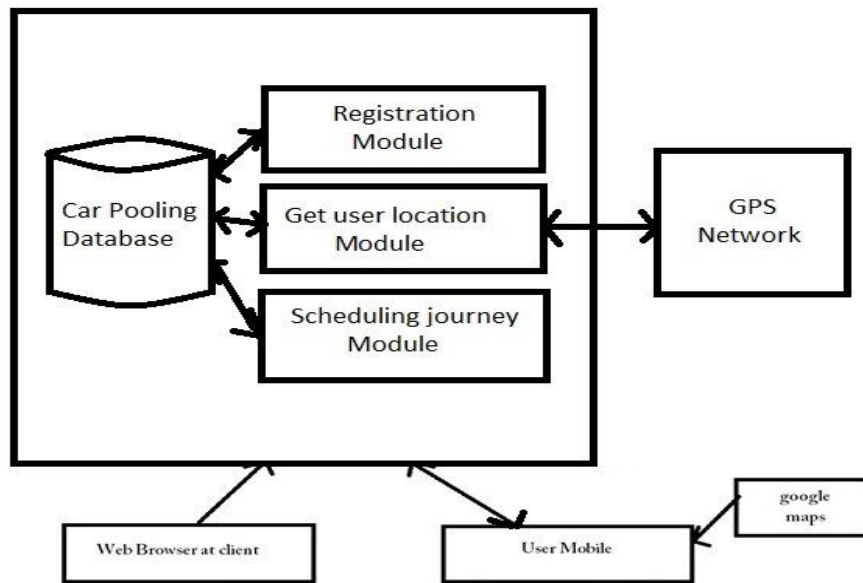


Fig: Carpool requests

Figure 2.5: Carpool Requests

## SYSTEM ARCHITECTURE



## CONCLUSION & FUTURE SCOPE

Carpooling application is implemented using greedy and heuristics approach and each time a share is picked the owner moves to the new location and finally, all the picked shares are shown on a live map.

Carpooling system is very effective means to reduce pollution and the congestion of vehicles in cities. It provides an eco-friendly way to travel as well as an opportunity to meet new people. Nowadays most people prefer personal vehicle to travel due to delay caused in public transport system and luxuries provided by private vehicles. Pre-registration ensures that only identified people get into the vehicle so that trust can be established.

## REFERENCES

1. R. Manzini and A. Pareschi, "A Decision-Support System for the Car Pooling Problem," journal on transportation technologies, Vol.2 No.2, 2012, pp. 85-101.
2. Swati. R. Tare, Neha B. Khalate, and Ajita A.Mahapadi,"International Journal of Advanced Research in Computer Science and Software Engineering".
3. Deepak B. Nagare, Kishor L. More, Nitin S. Tanwar, S.S.Kulkarni and Kalyan C. Gunda , "International Journal of Innovative Technology and Exploring Engineering(IJITEE)"
4. "SOAP Version 1.2 Part 1: Messaging Framework (Second edition). "Saddle extension to the Mule Enterprise Service Bus.",
5. Wilensky, U. and Stroup, W. 1999. "NetLogoHubNet Gridlock model. Center for Connected Learning and Computer-Based Modeling."Northwestern University, Evanston, IL, 1999.
6. Gershenson, C."Self-organizing traffic lights.", Complex Systems, Vol16 29-53, 2005