Mobile Application for Carpool System

Utkarsh Singh, Deepak Shinde, Tushar Sanap, Nilesh Patil, Prof. Rahul Jadhav

Department of Computer Engineering, Fr. Conceicao Rodrigues Institute of Technology, Vashi, Navi Mumbai

Abstract— Carpooling (also car-sharing, ride-sharing, lift-sharing), is the sharing of car journeys so that more than one person travels in a car. By having more people using one vehicle, carpooling reduces each person’s travel costs such as fuel costs, tolls, and the stress of driving. Carpooling is seen as a more environmentally friendly and sustainable way to travel as sharing journeys reduces carbon emissions, traffic congestion on the roads, and the need for parking spaces. Authorities often encourage carpooling, especially during high pollution periods and high fuel prices.

I. INTRODUCTION

In Mumbai alone, there are 230,000[2] vehicles plying on the roads at any point point of time essentially a 55%[4] increase in the past 5 years and this figure is increasing steeply everyday. This gives rise to three prime factors:

1. Pollution
2. Depletion and rising rates of fossil fuels
3. Massive traffic congestions

Most of the people, on an average, have to travel 15 km[2] to reach their workplace. Besides long working hours, the stress of driving a car or a two-wheeler for a minimum of 40 minutes adds to the exhaustion.

The idea is to bring together car owners and passengers looking for carpool service. The users have to make clear that they are willing to share personal vehicles with people travelling to the destination daily. The application seeks details about the starting and end points of travel which are processed to draw a list of professionals who have identical requirements.

Drivers and passengers offer and search for journeys through one of the several mediums available. After finding a match they contact each other to arrange any details for the journey(s). Costs, meeting points and other details like space for luggage are agreed on. They then meet and carry out their shared car journey(s) as planned.

Carpooling is commonly implemented for commuting but is increasingly popular for longer one-off journeys, with the formality and regularity of arrangements varying between schemes and journeys.

Carpooling is not always arranged for the whole length of a journey. Especially on long journeys, it is common for passengers to only join for parts of the journey, and give a contribution based on the distance that they travel. This gives carpooling extra flexibility, and enables more people to share journeys and save money.

II. LITERATURE SURVEY

The literature survey consists of two existing systems named as follows:

A. Blablacar
B. Folksvagn

A. BLABLACAR[1]

BlaBlaCar is the world's largest long-distance ridesharing community. Conceived in December 2003 by Frédéric Mazzella, and founded in 2006, BlaBlaCar connects drivers and passengers willing to travel together between cities and share the cost of the journey. BlaBlaCar has more than 20 million members across 19 countries.[3]

Members must register and create a personal online profile, which includes ratings and reviews by other members, social network verification, and rate of response. “Profiles of members show how much experience they have of the service, meaning those with more – known as "ambassadors" – attract more ride shares.

One major shortcoming of this application is that it only offers inter-city carpooling options which our application aims to rectify and add intra-city commuting options too.

B. FOLKSVAGN[1]

FolksVagn offers a community-based system that helps people share rides with others. While the passengers get rides at costs much cheaper than a regular taxi service, the car owner gets a share of the fare.

It is open only to corporate clients as it requires a corporate email for registration and has a prepaid account or online wallet system to pay for the ride.

III. PROPOSED SYSTEM

The Carpooling application will be implemented in Android operating mobile phones. The application will try to cover the following:

- User accounts for both the ride providers and the ride seekers.
- Use GPS to find nearby carpoolers.
- Find optimum paths and allow carpooler to choose one from it.
- Integrating google maps so that the ride provider can provide his detailed route and then the potential passengers can view and decide their boarding and de-boarding point.
- User profile which will have car details like registration number, color and model of the car apart from the profile photo of the user fetched from his google account.
- Option to choose the carpoolers so as to give women the option to travel with women only.
- Profile rating to ensure the quality of ride.

Chat Rooms for online users to interact and co-ordinate and quick messaging

Emergency button to send current location of the passenger to a pre-specified contact in case of emergency.

IV. SYSTEM DESIGN

This deals with the system block diagram and the data flow diagram of the proposed Carpool System.

BLOCK DIAGRAM

The block diagram for Mobile Application for Carpooling System is as shown in Figure. 1

![Block Diagram](image)

There will be 4 important modules as shown in figure 1:

1. Registration
2. Offer Ride
3. Seek Ride
4. Feedback
5. Emergency

1. REGISTRATION

This module takes in all the relevant information pertaining to both the users - ride providers and ride sharers.

2. OFFER RIDE

In this module, a user who has vacant seats in a car and is going in a particular direction, say from point A to B fills in necessary details such as Name, License plate number, car make, the route that he follows to get to his destination using Google Maps into the application which is then accessible to all other members who are registered in the application. The person offering the ride can also fill in his preferences about the smoking, music, pet travel conditions that he expects the co-passenger to abide by. The person offering the ride has the liberty to accept or decline requests from other users looking to pool with him or her.

3. SEEK RIDE

This module will be responsible of seeking or finding a suitable ride as per carpoolers’ requirements. Whenever a carpooler or a user searching for a ride queries the system for availability of rides from source A to destination B, system will check the user’s current location and reply back with all currently plying rides with vacancies on the same route within 5 km radius from user’s current location with the help of Google Maps & GPS module. The user will select most suitable ride and send a request to the person offering the ride and if the ride-provider accepts the request, then module will prompt the ride details consisting estimated arrival time, estimated delay (if any), corresponding car and ride provider details.

4. FEEDBACK

In order to provide all users a quality and hassle free service, a feedback system is implemented that will allow the co-passengers to rate the driver or ride provider and vice versa.

5. EMERGENCY

This module is responsible for assessing and responding to the emergency situations which may arise while carpooling. This service can be accessed by the emergency button provided on the screen. Emergency situations, like accidents, can be responded by sending a message to the Emergency Contacts taken from user at the time of registration and sending a request to Ambulance Services and Police Station of respective area with the help of user’s current location. For women safety concern, the emergency module can be used to inform the Emergency Contacts and the Police Station of respective area with user’s current location to get the help.

V. IMPLEMENTATION

This project uses following modules:

Volley Library:

Volley \(^5\) is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley is available through the open AOSP (Android Open Source Project) repository.
Volley excels at RPC-type operations used to populate a UI, such as fetching a page of search results as structured data. It integrates easily with any protocol and comes out of the box with support for raw strings, images, and JSON. By providing built-in support for the features you need, Volley frees you from writing boilerplate code and allows you to concentrate on the logic that is specific to the app. Volley plays a key role in sending data to the server and fetching the data from the server whenever required in a faster way.

Server handling using PHP/MySQL:
All user data is stored in the MySQL database on the server. This database is updated as the application runs its course. To handle the database manipulations PHP along with the MySQL (A relational database driver used in the PHP programming language to provide an interface with MySQL databases) is used. This PHP scripts are invoked whenever applications requests for a data change.

Google Cloud Messaging:
This application uses GCM [6] for implementing the chat rooms. Chat rooms enables various users to communicate with each other in a group chat fashion. Typically GCM implementation involves three components. Google cloud messaging server, the app server and the client app. We should take care of writing the app server and the client app. In order to make calls from the app server to GCM server XAMP protocol is followed. XAMP supports both downstream and upstream (device to GCM, then from GCM to server) messages. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. Also there are API calls to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and allow user interaction with the map. The API allows you to add these graphics to a map:

Google Maps directions API[8]:
The Directions API is used for finding the distance and directions between two locations on the map. It results the optimum route between the two locations. The Directions API calculates directions between locations using an HTTP request made from user application. Request URL editing is done in following format:

```
https://maps.googleapis.com/maps/api/directions/output?parameters
```

Application uses JSON output format and parameter include coordinates for source, destination, and maximum of 8 waypoints between them.

Returned result in contains the directions details in encoded polyline points format which gets decoded using an algorithm implemented in JAVA[9]. This returns points to be plotted on the map to generate the route.

This is used in Offer Ride where user can see decide route he/she wants to follow before creating a ride and in searched ride details where other users can get know which route ride provider is going to follow.

Google Place Autocomplete[10]
To enhance the user experience application has used autocomplete service from Google Places API for Android. It returns place predictions in response to user search queries. As the user types, the autocomplete service returns suggestions for places such as businesses, addresses and points of interest. The application uses an autocomplete widget for this functionality in where user enters search terms, the widget presents a list of predicted places to choose from. When the user makes a selection, a Place instance is returned, which is used to get location coordinates for the selected place.

Location Services[11]
GPS and Network Providers are two different ways to get Android device location (latitude and longitude). This application uses both of them to get precise location details in both outdoor and indoor locations. It is done by implementing Location Listener interface to receive continuous location updates even if device moves.

Reverse Geocoding[12]
Reverse geocoding is the process of back (reverse) coding of a point location (latitude, longitude) to a readable address or place name. This permits the identification of nearby street addresses, places, and/or areal subdivisions such as neighborhoods, county, state, or country. Combined with geocoding and routing services, reverse geocoding is a critical component of mobile location-based services and Enhanced 911 to convert a coordinate obtained by GPS to a readable street address which is easier to understand by the end user. Application uses reverse geocoding to get current location’s address to search and offer ride from current location.

VI. SCREENSHOTS

Offer Ride:

This is the screen of application where user can offer ride for carpooling. User can offer ride from his/her current location or any desired source location to any desired destination possible via car route on map.

The distinguishing feature of this application is that if user does not wish to follow the optimized path suggested by the application then user can add up to 8 waypoints to mould the route as per convenience and applications provides the optimized path through those waypoints.

Application allows user to offer two kinds of rides namely - single ride and other is return ride (to and fro) if user wishes to offer ride on return journey too. After filling in all details required for ride, user can confirm ride and this offered ride will be made available to other users to pitch in carpool request.

Seek Ride:

This is the screen of application where user can search rides for carpooling between the interested source and destination. Application allows users to search ride from either user’s current location or any desired source location to desired destination and it returns all the available rides supporting users’ needs.

User can see the route, ride provider wishes to follow by clicking the map icon in ride description from the list of available rides.

From this screen user can send carpooling request to ride provider and to show interest in his/her carpool.

Your Rides:

This page displays all the rides user has created until now. User then can view them for summary and delete them when he wants to.
Emergency Button:

When user clicks this button for 5 times continuously within the duration of 3 seconds, the application sends the GPS coordinates of the user to the predefined emergency contact in emergency contact module. This feature is essential for emergency situations and women safety. Multiple button click requirement avoids the false positives of this feature.

Chat Room:

This feature enables all the users to group chat

VII. CONCLUSION

This paper elaborates the proposed system which consists of 5 main modules which are Offer a ride, Seek a ride, Feedback, Emergency, and user authentication via Registration. This system involves support from Google maps services and GPS module to provide user specific services and through Feedback the user experiences are recorded for rating the users. The user’s safety is supported by an Emergency Module which is connected to the local emergency services to provide required help.

The main purpose of this paper is to illustrate the initial prototype of the proposed system.

REFERENCES

[1]. www.wikipedia.com
[3]. www.blablacar.in
[6]. https://developers.google.com/cloud-messaging/gcm
[7]. https://developers.google.com/maps/documentation/android-api/
[8]. https://developers.google.com/maps/documentation/directions/intro
[9]. jeffreysambells.com/2010/05/27/decoding-polylines-from-google-maps-direction-api-with-java
[10]. https://developers.google.com/places/android-api/autocomplete
[12]. https://en.wikipedia.org/wiki/Reverse_geocoding