A MOBILE CONTEXT BASED APPLICATION FOR SECURE CAB SHARING SERVICES

[1]B.Vidya Sarika, M.Tech, Dept.of ComputerScience and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India.

Email id: vbvidyasarika@gmail.com

[2] Dr.A.M.Sowjanya, M.Tech, Ph.D, Asst Professor, Department of CS&SE, Andhra University College Of Engineering, AU. Visakhapatnam, India

Email id:sowmaa@yahoo.com

ABSTRACT

A safe, efficient, and convenient transportation system is integral to the health of our economy and quality of life. Roadways is one of the most powerful means of transportation which mostly includes private passenger's car. Especially in densely populated areas, people usually face a trade-off between convenience and costs. On one hand convenience as a driving factor, people are preferring to use cars which leads to air pollution, traffic jams, and inflation of fuel price whereas on the other hand, encourage many people (e.g. commuters) to use collective transportation (CT), such as public transport systems. However, it does not support door-to door transportation and might be inconvenient due to limited services in off-peak hours or high costs when travelling long distances. A solution which can overcome the above problem is ride sharing, a form of CT making alternative transportation more affordable. In this paper, we present a modular platform supporting different forms of ride sharing. Ride sharing can be applied to road transportation modality, such as cars, taxis, trains, etc. In order to support ride sharing, a system requires a set of information, such as time, origin, or destination for finding appropriate matches (i.e., people, having the same or similar ways of travel). In modern cities like New Delhi, these ride sharing system has been more popular due to Even-Odd system [the vehicles registered with even number should travel one day and with odd number on another day], ultimately reducing traffic on roads, air pollution etc. This system proposes a secure feature which allows user to forward SMS alerts to registered users.

KEYWORDS

Collective transportation, Ride-sharing, Spatio-temporal index, optimization, cab-sharing. GPS Navigation.

I. INTRODUCTION

Transportation is a major contributor to air pollution. Traffic jams and also scarce parking space are omnipresent concerns for inhabitants of large cities since traffic jams and the hustle of parking take up a significant portion of our daily lives and cause major headaches. Solving the problem by extending the road network is a costly and non–scalable solution. A more feasible solution to the problem is to reduce the number of cars on the existing road network. To achieve this solution, collective / public transportation has to satisfy the general transportation needs of larger groups in a cost–effective manner. Public transportation is cost-effective, but does not provide door-to-door transportation which makes the cabs attractive and scarce. While being cost–effective, the services offered by public transportation often:

1) Do not meet the exact, door–to–door transportation needs of individuals.
2) Require multiple transfers and detours that significantly lengthen travel times.
3) Limitation in off–peak hours.

For these reasons, cheap service offered by cabs / taxis, which meet the exact transportation needs of individuals and also eliminate the problem of parking, are in great demand. To better satisfy this demand, this paper presents an LBS that makes use of simple technologies and tools to offer a new cost– and resource–effective, door–to–door transportation means, namely cab–sharing.

Collective transportation is not a new concept. It is encouraged and subsidized by transportation authorities all over the world. The optimization of collective transportation has also been considered. This paper proposes a location-based Cab-Sharing Service (CSS), which reduces cab fare costs and effectively utilizes available cabs.

The CSS accepts cab requests from mobile devices in the form of origin-destination pairs. Then it automatically groups close by requests to minimize the cost, utilize cab space, and service cab requests in a timely manner. Simulation based experiments show that the CSS can group cab requests in a way that effectively utilizes resources and achieves significant savings, making cab-sharing a new, promising mode of transportation.

One significant problem faced in ride sharing is that we may not be aware of who the other passengers are. Though some people may mingle easily and make new friends, in some cases it is also risky and dangerous. So, in this approach, a safety module has also been incorporated for secure ride sharing. When the passenger feels he/she in a critical situation then he/she can double tap the power button which results in forwarding the location and even selected message to the selected contacts.

In the proposed system an android application has been created as a service or platform for ridesharing which can be installed on Android based Smartphone's. This software is developed for android system only. The two main reasons for choosing Android OS instead of another one are:

Android is an open source operating system, and thus allows reusing some pieces of program to create a new application. It is also quite well documented and sources can be easily developed for developing applications.

This application is modular context-based ride sharing platform allowing to store rides consisting of both general and transportation specific-information within a database. The platform aims at enhancing convenience, reliability, and affordability of different forms of ride sharing by means of context data. A RESTful API allows external applications or services to access the platform's information and tailor its user interface for the use among different means of transport. Apart from the type of transport the users of the transport system generally can be divided into two groups:

1) users who want to share rides and are looking for one or more ride-mates (based on the capacity) and

2) users who are looking for hitching and sharing rides. Users in the first group can define a ride and enter the details to the system.

The information includes time, origin, and destination points, the ride's description, and the meeting point. On the other side, users in the second group are able to search for a ride based on the given information (time, origin, and destination points). If an appropriate ride is found, then the user can sign up and wait for the confirmation from the ride's owner (if necessary). As soon as a user signs up for a ride, a notification is sent to the ride's owner. After the owner has made the decision, a notification is sent back to the user.

II. RELATED WORK

The problem we tackle in this paper is achieving concept of ridesharing through Android application using Android smartphone. The following are the different types of ridesharing concepts with different technologies being implemented below:

- *GPS Enabled Car Pooling System*: In this, the user with current location retrieve the list of users nearby in the Google maps who wants to join ride.
- *Ridesharing System with SMS Alerts*: It provides details of the owner and his/her car to maintain transparency between users of the system. It will track the location of users those who involve in pool

through GPS Navigation system. It has SMS Alerts facilities for notification purpose.

- ♦ *Rideshare matching using GIS*: The real-time rideshare matching system can be increased to provide matches not only for origin-destination but also enroot pick-up and drop-off of riders. A Geographic information system linkup with ridesharing databases, is proposed for implementing such rideshare matching services.

III. PROPOSED WORK

Let R2 denote the 2-dimensional Euclidean space, and let T ≡ N+ denote the totally ordered time domain. Let R = {r1, ..., rn} be a set of cab requests, such that ri = htr,lo,ld,tei, where tr ∈ T is the request time of the cab request, lo ∈ R2 and ld∈ R2 are the origin and destination locations of the cab request, and te ≥ tr ∈ T is the expiration time of the cab request, i.e., the latest time by which the cab request must be serviced. A cab request ri =< tr,lo,ld,te > is valid at time t if tr ≤ t ≤ te. Let ∆t = te − tr be called the wait time of the cab request. Let a cab–share s ⊆ R be a subset of the cab requests. A cab–share is valid at time t if all cab requests in s are valid at time t. Let |s| denote the number of cab request in the cab–share. Let d(l1,l2) be a distance measure between two locations l1 and l2. Let m(s,d(.,.)) be a method that constructs a valid and optimal pick-up and drop-off sequence of requests for a cab–share s and assigns a unique distance to this sequence based on d(.,.). Let the savings p for a cab request ri ∈ s be p(ri,s) = 1− m(s,d(.,.))/|s| m({ri},d(.,.)) . Then, the cab-sharing problem is to find a disjoint partitioning S = {s1 ⊎ s2 ⊎ ...} of R, such that ∀sj∈ S, sj is valid, |sj| ≤ K, and the expression Psj∈SPri∈sj p(ri,sj) is maximized.

A. CSS Model:

Ride sharing can be applied to different types of transportation, such as cars, taxis, trains, etc. In order to support ride sharing, a system requires a set of information, such as time, origin, or destination for finding appropriate matches (i.e., people, having the same or similar ways of travel). Transportation-specific information depends on the chosen mode of transport. This might be the number of available seats (e.g., car, cab), the type of vehicle we prefer, company, gender etc.

The Cab Sharing Service model comprises of some important following points:

Ride requests: Ride requests includes the origin point and destination point of the ride. The drivers even display their route with origin point and destination point and also via places, and also how many seats are left over for them. The passengers shouldspecify how many persons are included in the trip.

Waiting time: In this cab sharing service, there will be specific waiting time near by the location. For example, if the waiting time is 5 to 10 min, then after booking a ride, the passenger should start the trip within 10 min otherwise he can cancel the trip. The passenger then should choose another ride.

Cab searching module: The cab searching module quickly selects a small set of candidate cabs with the help of the Spatio-temporal index. The Spatio-temporal index of cabs is built for speeding up the cab searching process. The cab searching process includes single side searching and dual side searching. The dual-side searching is a bi-directional searching process which selects grid cells and taxis from the origin side and the destination side of a query simultaneously.

Constraints: The crux of the cab-sharing problem is to dispatch cabs to ride requests, subject to certain constraints. We say that a cab status V satisfies a ride request Q or Q is satisfied by V if the following constraints are met.

- **Vehicle capacity constraint*:*** The number of riders that sit in the cab does not exceed the number of seats of a cab at any time.
- **Time window constraints:** All riders that are assigned to V should be able to depart from the origin point and arrive at the destination point during the corresponding pickup and delivery window, respectively.
- **Monetary constraints:** These constraints provide certain monetary incentives for both drivers and riders. That is, a rider does not pay more than without ride-sharing; a driver does not earn less than without ride-sharing when travelling the same distance; the fare of existing riders decreases when a new rider joins the trip.

B. SYSTEM ARCHITECTURE:

Fig. 1 shows the architecture diagram of Ridesharing. In the proposed system, there are two participants - Driver and Rider. Both of them can benefit from the ride sharing platform through the ride sharing application installed in their android based mobile phones. To participate in the ride sharing, both users rider as well as driver have to register for the firstly using their mobile application. This processes of registration and login activity is effected by the service registration and the user account data is stored in the Accounts profile of database. Other then login data, the accounts profile database also comprises of extra details for security such as the user address, the organization where he/she is working. The process begins with the rider registering his ride through the mobile application. The ride registration data consists of source address, destination address, journey dates and start time of the journey or ride. The Google's Geo coding service then receives the start time of the ride. This module converts the physical address into Geo location coordinates and saves those coordinates in the Google fusion tables.

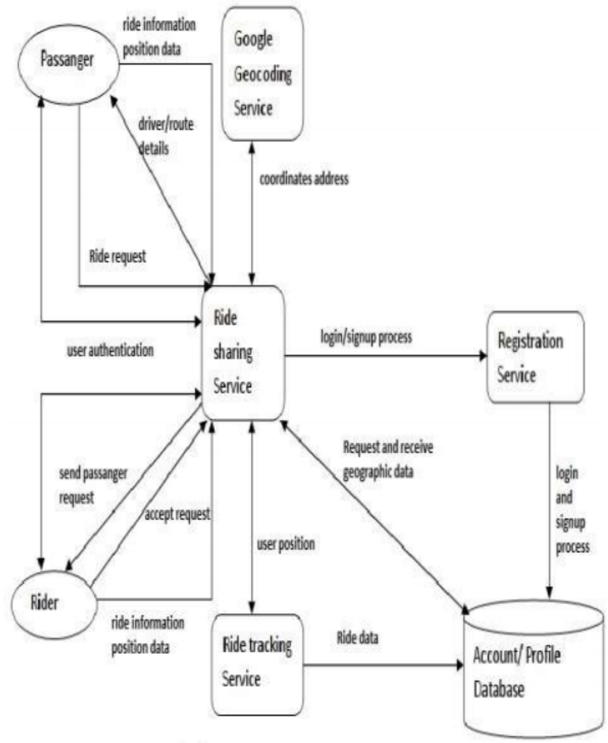

Figure 1: Architecture diagram of Ridesharing.

The rider after login gets displayed with the available rides through the ride search activity. The ride request is then processed by the web service of the ride sharing application. The rating service sorts out each rides based on the reviews provided the passenger. The filtered search result is presented to the rider or passenger along with the entire details of driver. After the rider selects a driver, rider request is then send on to the driver’s mobile application by the ride sharing service. Once the driver approves the request, driver and rider are enabled to communicate with each other through chat module. Once the ride starts, ride tracking service starts tracking the journey through the GPS coordinate data from the user's mobile device. The account profile database stores the coordinate data temporarily to provide assistance in the emergency situation. Once the journey gets finished, rider rates the journey commenced so as to further process the rated service and stores them along with profile data in database of the driver's accounts profile.

C. IMPLEMENTATION

Cab requests can be viewed as data points in spatio–temporal space. Partitioning n data points into k groups based on pairwise similarity of the data points according to a distance measure is referred to as the clustering problem, an extensively researched problem of computer science. Clustering is known to be NP-hard[2]. However, there are a number of efficient bottom-up and top-down methods that approximate the optimal solution within a constant factor in terms of a clustering criterion, which is expressed in terms of the distance measure.

Hence it is only natural to approach the cab-sharing problem as a clustering problem and adopt efficient approximations to the task at hand. For these approximation algorithms to converge to a local optima, an appropriate distance metric d(.,.) between two cab requests and/or cab– shares needs to be devised. For d(.,.) to be a metric for any three cab requests or cab–shares i,j,k is has to satisfy the following four conditions: $d(.,.) \geq 0$ (non-negativity), $d(i,i) = 0$ (identity), $d(i,j) = d(j,i)$ (symmetry), and $d(i,j) + d(j,k) \geq d(i,k)$ (triangular inequality).

While a clustering approach may find a near-optimal cab-sharing solution, it has several drawbacks. Since a cab request is only valid during a specific time interval, the set of valid cab request that can be considered for clustering is changing over time. While a hard time-constraint can be incorporated into a distance measure, the measure will not satisfy the triangular inequality requirement. An alternative approach could at every time step t retrieve the set of valid cab requests, and perform a partitioning-based clustering on the set according to some distance metric. However, since at any time instance t the number of valid cab requests nt and the number of possible K-sized valid cab–shares are comparable, an iterative partitioning-based clustering approach would entail O(n2 t) distance calculations per iteration at every time instance t, making it infeasible in practice

Step 1: Calculating Costs.

When a ride request is made, the system calculates cost of ride for the respective cab, the extra cost incurred by the cab to reach the users destination to facilitate cab-sharing is also calculated.

Step 2: Selecting the Best Cab–share.

The best cab-share is selected based on the following conditions:

I. A cab which charges less price or
II. A cab which is in the same vicinity, or
III. A cab which is already in the same route.

Step 3: Pruning the Search Space

When the ride requests meet the selected criteria, the search space is further pruned for better decision-making.

IV. CONCLUSIONS

In this paper we use a modular context-based ride sharing platform over Android system as it is most widely used operating system. This application is highly useful for in areas where it is densely populated and where there is no convenient mode of transportation. As android application is more popular, this application can be well known and useful.

As a future work we plan to further extend our platform, to e.g., support carpooling. Hence, free seats could be shared with people travelling in the same direction.

V. REFERENCES:

[1] D. Zhang and T. He, "CallCab: A unified recommendation system for carpooling and regular taxicab services," in Proc. IEEE Int. Conf. Big Data, 2013, pp. 439–447.

[2] J. Han and M. Kamber. "Data Mining: Concepts and Techniques". Morgan Kaufmann, 2005.

[3] S. Ma, Y. Zheng, and O. Wolfson, "T-Share: A large-scale dynamic ridesharing service," in Proc. 29th IEEE Int. Conf. Data Eng., 2013, pp. 410–421.

[4] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," in Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2010, pp. 99–108.

[5] W. Wu, W. S. Ng, S. Krishnaswamy, and A. Sinha, "To Taxi or Not to Taxi?—Enabling personalised and real-time transportation decisions for mobile users," in

Proc. IEEE 13th Int. Conf. Mob. Data Manage., Jul. 2012, pp. 320–323.

[6] E. Zeitler and T. Risch. Using stream queries to measure communication performance of a parallel computing environment. To appear in Proc. of DEPSA, 2006.

[7] S. Ma and O. Wolfson, "Analysis and evaluation of the slugging form of ridesharing," in Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2013, pp. 64–73.

[8] Gidófalvi, G. and Pedersen, T.B. Cab–Sharing: An Effective, Door–to–Door, On–Demand Transportation Service. Proc. of ITS, (2007), 1–8.

[9] R. W. Calvo, F. de Luigi, P. Haastrup, and V. Maniezzo, "A distributed geographic information system for the daily carpooling problem," Comput.Oper.Res., vol. 31, pp. 2263–2278, 2004.

AUTHOR'S PROFILE

B. Vidya Sarika, Pursuing her M.Tech in department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, A.P, India. She obtained her B.Tech(CSE) in S.R.K Institute of technology, Vijayawada, Krishna Dt.

Dr.A.M.Sowjanya received her M.Tech and Ph.D. Degree in Computer Science and technology from Andhra University. She is presently working as an Assistant Professor in the Department of Computer Science & Systems Engineering, College of EngineeringA), Andhra University.Her areas of interest include Data Mining, incremental clustering and Big Data Analytic.