

GigaGO: An Android-Based App for Cab Booking

Pavandeep Kaur^{*1}, Sukhmeet Singh^{*1}

^{*1}Apex Institute of Technology, Chandigarh University, Punjab, India

Abstract:

Cab service at the doorstep to be called to the Smartphone app offers passengers convenience. This research article introduces the android application for a cab booking named GigaGO. One can book a rental car with this application. This cab booking application effectively uses Google Maps to locate the cab closest to your location. For first time users, there is a one-time opportunity to book a cab before making an account. This application has two entities, Passenger and Driver. Passengers can hire a cab by requesting a nearby cab. Based on local requests, a driver can choose the nearby passenger. For the updations regarding distance left and time, one can click the BEEP-BEEP option. Passengers can also cancel the booking. Moreover, this cab booking App is a two-way service that allows all passengers in search of a ride and cab owners to register with the app, thereby reducing the gap between both.

Keywords — Cab booking, Android application, Google Maps, Parse server.

I. INTRODUCTION

With the passing of time, technology increasingly improves to make daily life simpler. Owing to people's demands, the transport network is growing. Many modes are available but commuting in metropolitan cities around the world is still a big problem [1] and people wait too long to get taxis or buses. Vehicles are used by all modes of transport; regular people tend to drive by commercial means as private vehicles are easier to purchase [2]. In the conventional car booking method, the amount of knowledge exchange was minimal. That happens due to information exchanged by the customer was not registered in the system but just listened to by the service provider, which may result in a lack of relevant information results in a delay in service. In the current method, the service response time was ordinary due to one-way contact, which deteriorates the demand and the goodwill of the company. It also takes needless time and effort, making it difficult for consumers who do not have

time to spend on such bookings to live their quick lives.

We proposed a GigaGO cab booking application. Giga is representing an enormous scale that means our application can run on a large scale. Just like any other cab booking application, our application helps people to find out a cab but along with the unique functionality of hiring a cab without registering for the first time only. This acts as a stepping stone when people don't have time to register themselves when they are in a hurry. This functionality makes our application unique from others. For storing all the details and updations, we have used the Parse server as a backend database. Back4app makes it easy to store a huge amount of data [3]. Google maps also play an important role in this framework. Java is used for coding as it is easier yet powerful than Kotlin.

This paper is arranged as follows: including the introduction in Section-1. Section 2- includes a literature review of the framework. The proposed technique is planned in Section-3. The smartphone

application is introduced in Section-4 and this document is concluded in Section-5.

II. LITERATURE REVIEW

In the literature survey, the findings of the researchers will be highlighted which will become the basics of current implementations. For achieving this, we read many publications and review articles related to our project; so that we get knowledge about the existing work and do our proposed work accordingly. To set out the basic set of accurate criteria, we have checked some of the smartphone apps in Android and iOS. EZCab with modern computing technology helps users book local cabs using their mobile phones or PDAs fitted with short-range network service interfaces. EZCab finds and hires free cabs using handheld ad hoc vehicle networks [4]. CabSense would begin by displaying the current position that wastes a certain amount of valuable time. The consumer does not want a cab from his current place [5]. "13Cabs is a good option that provides a range of services, such as wheelchair/scooter booking, locating points of interest, and is not limited to taxis" [6]. The app does not, however, endorse horizontal orientation. An app like MyTaxi doesn't specify a drop off destination. The customer is in a bind whether he/she is taken to the requested location by the taxi after he/she is picked up. However, the app does have several choices for storing your interests and your card [7].

"ToGo Cabs is an Android based smartphone cab reservation application that targets Kansas state residents" [8]. While some taxi services brag that their taxi is only a step away, while some other taxi services boast the timeliness of their action. Uber was created as "UberCab" by Travis Kalanick and Garrett Camp in 2009 and was launched the following June. Uber expanded globally from 2012 onwards [9]. In 2014, the company played with carpooling features and made more updates. We conducted a literature survey on a variety of other cab booking sites. Ola Cabs is one of the best car-hailing service applications available in India after Uber. In a few taps, it offers the easiest and shortest travel options. You need a taxi to download the app,

and you're free to go [10]. Hailo taxi (now owned by My Taxi) lets you book your taxi in a couple of taps. Real-time taxi monitoring is available after you book. Confirmed licensed drivers by local authorities have security [11]. Lyft helps you to easily order a car with a button tap and get picked up by a nearby open driver who will take you to your address away [12].

III. PROPOSED WORK METHODOLOGY

In this study, we presented a framework designed to create a user-friendly mobile application. We have done applied research approach as research methodology. The following methodology *has been followed* to achieve the objectives defined in the proposed research work: -

1. A detailed study of cab booking applications has been done.
2. Installation of the software required for the project has been done, and the relative pros and cons are identified.
3. Various parameters have been identified to evaluate the proposed system.
4. Comparison of newly implemented approach with an existing approach has been done.

We provide a forum for passengers and drivers to connect with others. The proposed system is also close to another cab booking application that provides a live map tracker and book trips remotely. Figure 1 shows the flow chart of proposed work methodology.

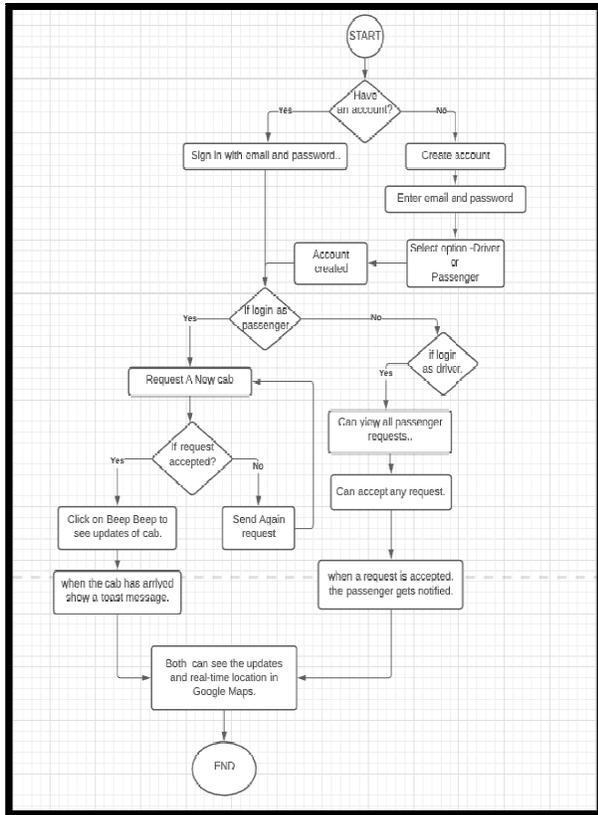


Figure 1: Flow chart of proposed work methodology

SOFTWARE SPECIFICATIONS: For this application, following software specifications have been considered as shown in Table 1.

Table 1
 Software specifications of GigaGO app

DEVELOPMENT END	APPLICATION END
Operating system : Windows 10 Language : Android SDK, Java Database : Parse Server Technologies: Android, XML, Google Maps Debugger: Android DDMS	Framework : Android SDK Version 3.1 Network : Internet (cellular or Wi-Fi)

IV. IMPLEMENTATION PART

Step 1: First link to the real-time database i.e. Parse server. Figure 2 shows the view of database. Parse Server is an open-source Backend-as-a-

Service (BaaS) application originally developed by Facebook. The platform has an involved and robust group of fundamentalist developers, who are continuously working to better the already amazing and scalable platform. At the heart of many implementations is the notion of user identities that allow users to access their information safely. Parse Server offers a specialized user class called PFUser that automatically manages a large part of the features needed for user account management.

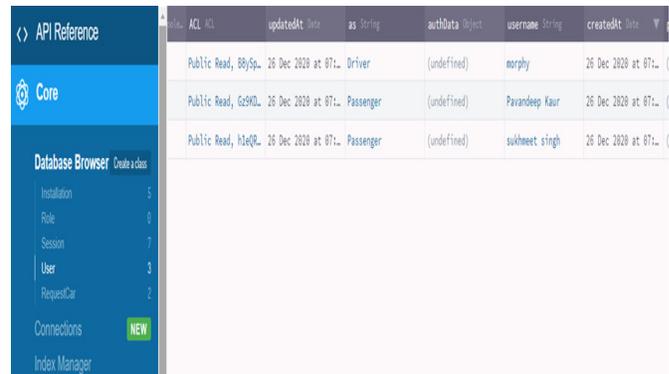


Figure 2: Parse server database

Step 2: Pick a layout style and first create a sign in tab. Figure 3 displays the Log-in/ Sign-up layout. If any user already has an account, he/she will sign in directly by filling in the credentials as shown in Figure 2.

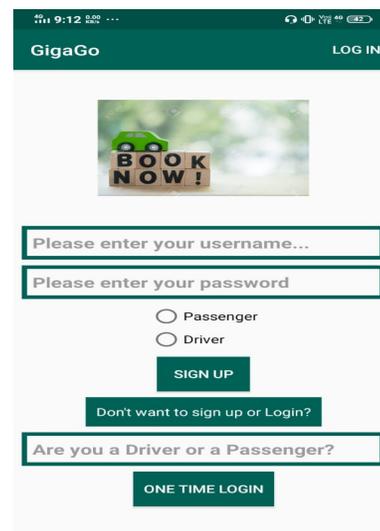


Figure 3: Sign-up/ Login page of GigaGO

Step 3: If any user does not have a registered account, they must first enter a username, a password. After that, select whether the user is a passenger or a driver. And we will get two distinct types of users. The users who are willing to assist and the others are those who require assistance. So while signing-up, it is mandatory to select one option. After that, press the Sign-up button to build an account. There is also a one-time authentication feature available. If the customer does not have a registered account, he/she can use this one-time login option to book a taxi for the first time. One should type Passenger in the text field, and then press the ONE TIME LOGIN button as shown in Figure 4.

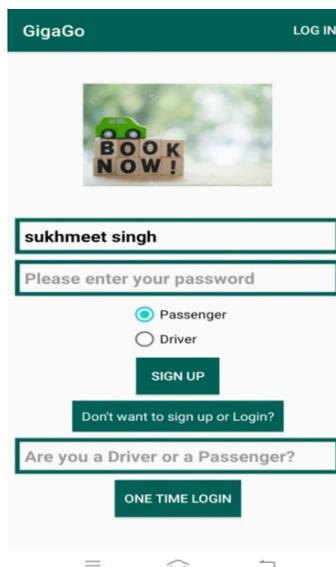


Figure 4: One time direct login without making account

Step 4: The user who requires assistance will see three buttons: request a car, Beep Beep, and Log-out (Figure 5). By clicking on the log-out button, the user can log-out from the app. By clicking on the request for a car, the user can send a request to book a cab. When the request is accepted by any driver, then the user can click on the Beep Beep button to see the updates of the driver. How far the driver is and how much distance is left, all the alerts will be made to the passenger by pressing the Beep Beep button. When the cab arrives, then a toast message will be displayed that Hurray! Your cab has arrived.

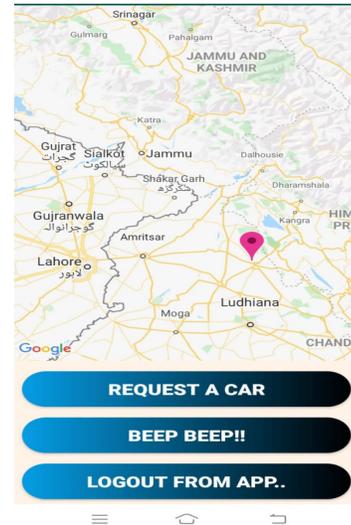


Figure 5: Layout of Passengers view

Step 5: Passengers can hire a cab by clicking on the 'REQUEST A CAR' button and can also cancel the booking by clicking the 'CANCEL YOUR GIGAGO REQUEST' button as shown in Figure 6.

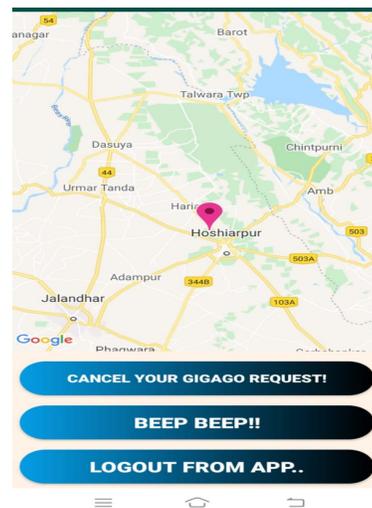


Figure 6: Cancellation of booking request

Step 6: The driver who is willing to assist can see a button named GET NEARBY REQUESTS. By clicking on this button, the driver can see all the requests. Users applying for a taxi will be seen here. As shown in the figure 7, there is a driver named Morphy and a passenger named Pavandeep Kaur. Morphy as a driver wants to give a ride to

Pavandeep Kaur who is just 24.0 miles away from him (Figure 8). It depends upon the driver whom he wants to give a ride.

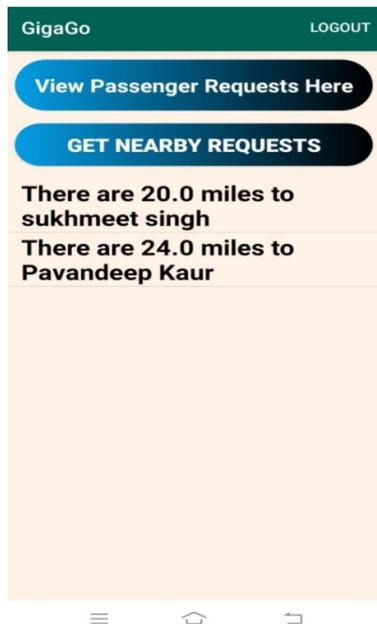


Figure 7: Get nearby requests



Figure 8: Select passenger to give ride

Step 7: Our GigaGo app asks the user to open Google Maps in chrome or any browser. Choose the option accordingly as shown Figure 9.

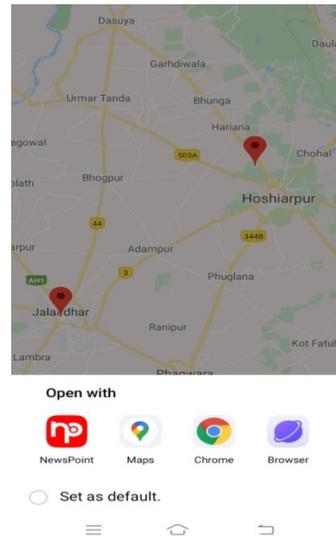


Figure 9: Open Google Maps with different browsers

Step 8: GigaGo app uses Google maps because we know that Google maps use advanced data structures such as trees and graphs. By selecting any request, the path will be opened on Google maps. And the driver can see the shortest path and the time to reach the passenger as shown in Figure 10.

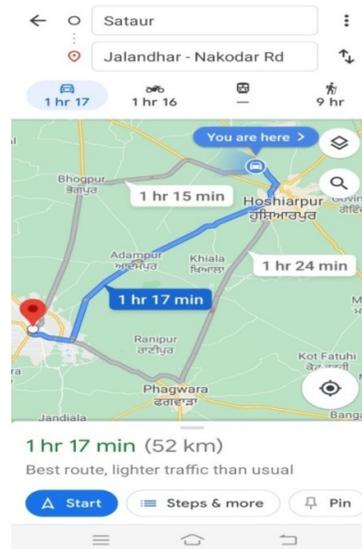


Figure 10: Choose shortest path

On the other hand, the passenger who has requested a cab will get information about the driver regarding the driver accepted the request. The user will be able to see all the real-time updates of the driver. All the data will be saved in the parse server such as the Latitude and longitude of the Driver and the passenger, name of the driver, and login credentials, etc

V. CONCLUSION AND FUTURE SCOPE

In the current situation, travel and transport are becoming very common among millions of people all over the world. That is also one of the greatest needs of several individuals when they have to travel around the world. This paper is a contribution towards society as the application is easy to use. Hence, anyone can use it. The location monitoring facility is introduced using the Global Positioning System (GPS) with the aid of Google Maps. The findings of this study suggest that, in order to please consumers, the application has to be easy to use and to reach new ideas. In the future, we will add further features to the application. Besides, the function for monitoring the real-time position of the passenger when on a trip involves working capital and simultaneous rollout action, so serious experts should be consulted and only integrated into the next iteration if considered feasible.

REFERENCES

- [1] T.Manivel, P Kowsalya, R Muthumani, S Sripavithra, T Vaitheeswari, (2020), An Efficient Android App Monitoring for Cab Booking and Servicing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCICCT (Volume 8 – Issue 08)
- [2] Adewole Adewumi and Et al, Developing a mobile application for a Taxi Service Company in Nigeria, (December 2015)
- [3] Tsyganov, Dmitry, Fundamental properties of SaaS Architecture: Literature review and analysis of development practices (2018)
- [4] Zhou, Peng & Nadeem, Tamer & Kang, Porlin & Borcea, Cristian & Iftode, EZCab: A Cab Booking Application Using Short-Range Wireless Communication.(2005) 27-38. 10.1109/PERCOM.2005.21.
- [5] Yo, taxi! CabSense finds ride and a way to crunch big data (April 16, 2020), VentureBeat, [url:https://venturebeat.com/2010/04/16/cabsense-finds-available-taxis/](https://venturebeat.com/2010/04/16/cabsense-finds-available-taxis/)

- [6] Business organizational strategy for Adelaide- 13cabs (October 9, 2019), <https://affordableessaywriting758.blogspot.com/2019/10/business-organisational-strategy-for.html?m=1>
- [7] Mytaxi: connecting drivers and passengers, Google cloud (2007)
- [8] Meesala, Nirupama , ToGo cabs: an Android phone cab reservation application (2012), url: <https://hdl.handle.net/2097/13938>
- [9] Dudley, Geoff & Banister, David & Schwanen, Tim. (2017). The Rise of Uber and Regulating the Disruptive Innovator. The Political Quarterly. 88. 10.1111/1467-923X.12373.
- [10] Panigrahi, Ashok. (2018). Success Story of a Start-up -- A Case Study of OLA Cabs. IOSR Journal of Business and Management (IOSR-JBM). 20. PP-30.
- [11] Top 10 Taxi Booking App Companies in the World (August 2017), Infinite Cab – a taxi dispatch software, url: <https://blog.infinitecab.com/top-10-taxi-booking-app-companies-in-the-world/>
- [12] Abel Brodeur, Kerry Nield, An empirical analysis of taxi, Lyft and Uber rides: Evidence from weather shocks in NYC, Journal of Economic Behavior & Organization (2018), Volume 152, Pages 1-16, ISSN 0167-2681, <https://doi.org/10.1016/j.jebo.2018.06.004>.