

# Implementation of a Voice Based App for Booking Cab Services

Baswaraju Swathi

Information Science, New Horizon College of Engineering, Bangalore, Karnataka, India

## ABSTRACT

The Transportation industry has seen an exponential technological growth with the preface of Applications and services which provide with an easy option of travel by booking Cabs which will turn up at the doorstep. Some of the foremost apps in this category are Ola, Uber and so many more. Nevertheless, these apps provide services to the common demographic of people/users. India is a country with over 13 million visually impaired individuals as per Census of 2011. With the advancement of the applications and technologies being available a means of easy and safe transport to these impaired individuals can be proposed. E-Cab Service aims at providing these voice-based transportation services aiming visually impaired as an application which is instinctive to book cabs.

**Keywords :** Visually Impaired, Cab Booking, Transportation Services.

## I. INTRODUCTION

Technology has evolved drastically over the past century, thereby making our lives significantly uncomplicated and making the world a smaller place. Various modes of transport make it possible for people to reach their destination in a matter of minutes and connect with their loved ones as and when they prefer. With the advent of transportation services, a lot of companies have started their own transportation network to help people travel from one place to another, by paying a small fare. Some of the better known companies in this sector are Ola, Uber, Lyft.

To help people with the booking process, these companies have a designated app that an Android/iOS user can download. This app uses the phone's GPS to show the user where the nearest ride is located and how long their booked cab/auto rickshaw/bike will take to reach them. It also provides an option to set pick up location as well as

drop location. It shows the amount that the ride will cost them, and displays multiple payment options for the user to pay the driver. It displays both the driver's and user's name and rating, to make the recognition proves on both ends.

Transportation network services have come, largely, as a boon to most members of society. However, as far as certain minorities are concerned, technology is yet to make the mark it has made on the vast majority. Such minorities are the people who are, in some form, impaired. These people could be visually or hearing impaired, or could have other physical disabilities. For such people, day to day commuting is a huge burden, and there is a huge gap between their needs and the available technology that provides the service.

In this system, we have focused on how to make the process of booking transportation for the visually impaired easier. Visually impaired people require an app[1] that is better suited to their needs, to make

cab/auto rickshaw bookings. The usual transportation apps have no facilities that are customised for the use of impaired people.

With the headquarters of Ola, India's biggest transportation service network, being in Bangalore, it becomes even more imperative for a cab service, within Karnataka, for the visually impaired.

E-Cab Service is aimed in providing these services to the visually impaired in the form of an application which is instinctive to book cabs. This system delves into the process of developing an app that brings cab booking services at the fingertips of visually impaired people[2], and hopes to bridge the gap between the visually impaired and other members of the society[3].

#### **A. Relevance of the Study**

The goal of E-Cab Service is to realise this potential by developing an app that is customised to the requirements of the visually impaired and makes the process of booking a cab and commuting, child's play. Over the course of 3 – 4 months, the contributors of this system have developed an Android - based cab booking app akin to Ola, which implements the technology to translate voice to text and text to voice. This is especially useful to the visually impaired because, they do not have to rely on others to make their cab bookings or payments, but can stand neck to neck with other able – bodied members of the society. Inclusivity among the masses was the main aim of developing the E-Cab Service system.

The visually impaired person may simply speak and book their ride. A voice automated system [3]tells them how far their ride is, how much the ride will cost them, what their driver's name is and much more[4]. This will help them with their daily commutes or travel lengthy distances.

#### **B. Purpose of Study**

In this vast growing population India has as many as 8.8 million people who were found to be blind in 2015 and another 47.7 million people had moderate and severe vision impairment, as per "The Lancet Global Health journal on August 2". The state of Karnataka alone has a population of Approximate 6.11 Crores out of which 264,170 individuals are visually impaired according to the census of Karnataka 2011. This demographic of the visually impaired population is rarely considered or given much importance in our country when focusing on certain factors, the one we are focusing on is Transportation. The main source of transport across cities for most people are either buses or taxi cabs. Most of the taxicabs in India are booked using either app based taxi services like Uber or a home grown Indian app based taxi services like Ola. Both these apps are available across the country in most states, but with such an enhanced outreach which is available to a number of people through the application, they do not consider the visually impaired individuals who will face difficulties in various modes of transport from buses to cabs and trains as well.

This system is an attempt for inclusion of the individuals who are visually impaired when it comes to Transportation, this can be achieved by using the existing technologies to implement an application which is intuitive and user friendly for the Visually Impaired to use with the help of Text-to-Speech for guidance and a Speech-to-text input from the user this way the user will be able to book a cab based on voice commands, and the inclusion of UPI or Mobile wallets make it easier for the payment transactions to take place.

#### **C. Problem Definition**

The problem statement is to build an Android application which helps the visually impaired

individuals to book cabs without the aid of others. This is achieved by developing an intuitive and user friendly application which allows the user to use the application in an effortless manner. There will be a Text-to-speech module which will assist the user about which element to select and the user can also use voice commands to give input to the application. Each page will have a set of fixed elements which can be interacted with, these can be text fields or buttons that can be enabled through the voice commands. An alert mechanism will be included which will alert the user if the cab is taking them in the wrong direction, any deviation will trigger these alerts. A simple payment method also will be included with the help of Ola Money, this will do away with the hassles presented by physical notes when it comes to the visually impaired.

#### D. Problem Explanation

In today's world there are many people who are visually impaired but thanks to technology they are able to make use of applications to help with their different abilities and use the smart phones just like any other who would use a smart phone. Cab service companies like Ola and Uber[6] help their customers to book cabs with just a few clicks, but this can be difficult for these visually impaired people to access such applications. To make it easier for them there is a need for integrated application. To help these people we have come up with an idea to make it easier for them to access Ola cab booking service with the help of voice access to know what's on the display and voice commands to give input to the device on what to do next. The Ola Money will help the user to pay for his/her ride directly from his digital wallet to Ola at the end of the ride making it easy and time efficient. An alert system will also help the person to know if the driver is taking them in the right direction or not.

## II. EXISTING SYSTEM

The present system of booking ride sharing services allows people to access any app like Ola or Uber, at the touch of a button. One can book a cab/auto immediately or hours in advance, to go to their required destination. The app shows the status of the ride, i.e, arriving/arrived. It also shows the time and distance covered as well as the OTP(one time password) and cab/auto number, while the cab arrives to your pick-up location[7].

Once the ride is located, you either match the given number (on the number plate) to the ride, or give your OTP to the driver. Then the driver starts the trip. On the way to your location, you can see what route the driver is taking via. Maps. You can also share your ride status with a friend. On reaching the location, the driver ends the trip. You can either pay in cash or through an integrated money sharing API like Google Pay. You can also give the driver a rating, based on multiple parameters. You can also lodge complaints in case you have been charged unfairly, or your driver has cancelled the ride.

As efficient as the present system of ride sharing services is, it fails when inclusivity to all kinds of people are concerned. In this system, we shall see what gaps are identified in the system when visually impaired people are concerned.

- 1) The entire process of locating the app on a touch screen phone, entering the pickup and destination locations and seeing how much money is to be paid to the driver, is a physical impossibility for any visually impaired person.
- 2) A visually impaired person will not know whether or not his ride has arrived, as he cannot see the cab/auto arriving.
- 3) It will be impossible for a visually impaired person to locate his ride based on seeing the

number plate of the ride. Likewise, even the driver will find it difficult to locate the client.

- 4) A visually impaired person cannot see the given OTP, that the driver would require to start the ride.
- 5) Someone who is visually impaired would not be able to see which route the driver is taking to reach a location. He/she would also find it difficult to share his/her ride status with a friend/family.
- 6) A visually impaired person would find it difficult to pay money to the driver, in cash, once the destination is reached.
- 7) Lodging a complaint or giving ratings would be a extremely difficult for someone who is visually impaired.

To carry out all of the above would be practically impossible for someone who is visually impaired, without help from an able-sighted person. The visually impaired person would not be able to book a ride on his own, exclusively. To counter some of these problems, we propose this new system that will enable visually impaired people to book rides, without requiring help from another person.

### III. PROPOSED SYSTEM

In the proposed system we would like to remove most of the limitations present in the current system. Here, the application will provide the user with a talk over feature where the user will give commands to the application like getting location which will get the current location of the device and then the user can speak out the destination address which will be converted to text format. Once the user says to book a ride the application will find the nearest cab and the ride will be confirmed. There will also be an alert system which will alert the user if the cab is going in the wrong direction. The payment can be done automatically at the end of a ride when the driver presses stop ride in his device through the Ola Money

payment gateway. This transaction will take place when there is money in the users e-Wallet.

### IV. ARCHITECTURE AND MODULES

Speech to Text (STT) [8] technique is used to afford voice commands to the system which successively will convert the voice into text design with precise action Text to Speech (TTS):- TTS is a technique that converts speech from text. TTS is vital for voice output to user GPS integration API for Android to categorize user location and tracking associated actions.

Android Pairing API to modify the pairing request dialog when attempt to pair with compatible devices over Bluetooth, BLE, and Wi-Fi probably Bar code integration.

Data streaming includes the encoding and decoding strings with a powerful GPU.

3D Visual system and sound generation, converting the object data to audio using gTTS API and play it to the user.

The various modules that were used are:

Login module: With the help of this module the user can login to the Ola cab booking platform through the api.

UI module: This module provides the user with a simple UI to give the destination address through speech and to get the current location with just a click of the button.

Payment module: With the help of this module the user can make payments directly to Ola at the end of the ride with the help of Ola Money payment gateway.

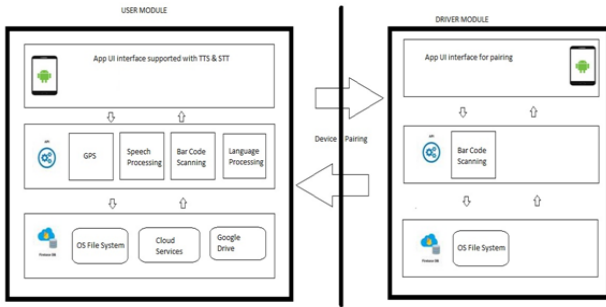


Figure1. User and Driver Module of the System

**A. Integrating Ola API**

Deep linking provides application interoperability between a native app or web view and the native Ola application. Deep links are URLs that reference the Ola app and support query parameters to affect the Ola app’s behavior once launched.

We use Web Deep Link as a means to integrate the Ola app. The following URL will open the OLA Home page and this is available as a system web view within the Affiliate App.

The following parameters can be sent through the URL by the Partner app , to provide user with some context so that he can better understand the Ola platform once he lands on it.

- **Source Identifier** (Mandatory Token For Affiliate Partner)  
 utm\_source = XAPP token received from Ola
- **Pickup Location coordinates** (Without this the location will be set to default)  
 lat = Latitude of the desired pickup point, lng = Longitude of the desired pickup point
- **DropLocationcoordinates**  
 drop\_lat = Latitude of the destination,  
 drop\_lng = Longitude of the destination
- **Category** (Type of Car)

TABLE I

Public Name	Value to be Passed
Micro	micro
Mini	compact
Sedan	economy_sedan
Prime	luxury_sedan
Share	share
Auto	auto
Lux	lux
Prime Play	prime_play
Prime SUV	economy_suv
Outstation	outstation
Rentals	local
Bike	bike

**pickup\_name** = Pickup Address string,  
**drop\_name** = Drop Address string  
**RideType**

- bk\_act** = <ride\_type> (Using either the Universal Link or App Link, considered only in App flow)
1. **rn** - Ride Now (opens confirmation page)
  2. **rl** - Ride Later (opens date-time picker)
  3. **"none" or default** - No Action (opens main page and user can choose Now or Later)
- when** = <ride\_type>
4. **later** - Ride Later
- dsw** = yes (using only Web Deep link)

**B. OLA Money Integration**

Ola money is a payment gateway[9] in today’s cashless world. It is powered by Zipcash and is a prepaid instrument as per RBI guidelines. Ola money makes payments simpler and easier with a seamless

experience. Ola money gets deducted when a ride is complete so the passenger need not carry cash. The payments will be initiated directly from merchants' app on checkout.

1) Make use of the latest version of Ola Money and that is currently v1.2 . The Ola cabs web api has the option to pay by Ola Money and already has the latest version of it.

2) **Server side requirements**

A bill generator which generates the estimate for the ride. It is already provided by the web api of Ola. An access key and secret key that can be acquired by signing up for Ola development.

3) **Access Tokens**

The merchant key or the access token can be taken from the dashboard once the developer logs on to the developer console. Here the sandbox key can be used by logging onto the Sandbox Merchant dashboard which gives a separate key other than the production key.

4) **Client Side Integration**

An android device with minimum API level 8 (Android 2.2) is required from the client side. This therefore ensures support for 100% percent of the android devices that are currently available.

5) **Incorporate button for Ola Money**

A button 'Pay using Ola Money' can be integrated into the application which will initiate the payment process once the user clicks on it or commands the device to make the payment.

6) **Initiate the payment flow**

Webview will be added once Ola is added to the users' device using the invokeOlaMoney() method.

```
public void invokeOlaMoney(String bill, String phoneNumber)
```

7) **Generating JSON bill from merchant server phone number**

```
Intent intent = new Intent("com.olacabs.olamoney.pay");
```

```
intent.setFlags(Intent.FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS);
intent.putExtra("bill", bill);
intent.setPackage("com.test.olacabs");
startActivityForResult(intent);
```

8) **Handling UI/UX for integrating Ola Money with the app**

The UI/UX aspect can be completely handled and designed by developers to reflect their company branding and logo. After this, the developer's logo can be customised by sending their logo through the intent.

9) **Handling responses from their side**

The payment process is done by Ola in a secure manner and the transaction result is transferred back to the app. An onActivityResult( ) method is used to manage transaction result.

```
switch (resultCode)
case 100: show_success();
case 104: show_failed();
```

Here the show\_success( ) is called when the response code is 100 and the show\_failed( ) is called when the response code is 104.

**V. PRELIMINARY DESIGN**

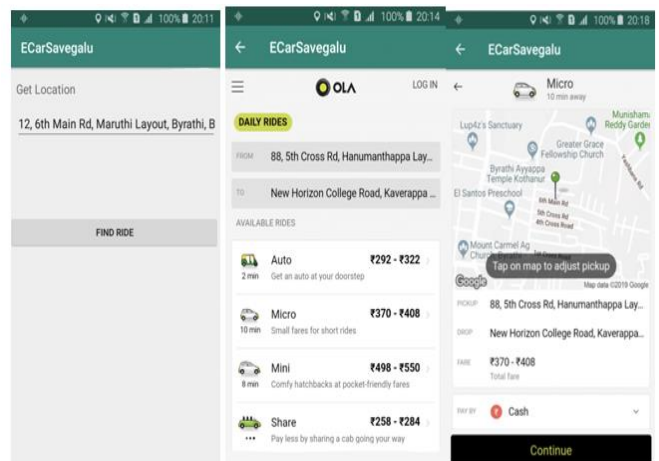


Figure 2. Preliminary Design of the system

In the preliminary builds of the application the E-Cab Service greeted the user with only a single

interactable element on the page which would be the users present location. This had to be initially entered manually by the user which was a drawback as we our aim was to make it effortless for the visually impaired to use. To overcome this drawback we implemented location services which would act as another element on the page and would auto select the users approximate location when triggered, this would do away with the task of entering it manually which would have probably resulted in the location marker being placed away from the users actual position.

In the preliminary build the user was also unable to specify their drop location in the initial stages of the application and had to wait until the data was sent to the ola server where there would be a delay before showing the available cabs nearby as the drop location value was not being sent. To curb this we included a voice enabled destination element which would allow the user to read out the destination they want to go to, and the application will convert their speech to text and read the input, this way when the data is sent to the ola server the integrated API takes only a fraction of the time to list the available rides near you.

Payment is the last transaction that takes place before ending the ride and when keeping in mind the visually impaired, cash is not the best option to be considered as it is tedious and a barrier. The initial builds only allowed the option of cash to be used as the payment methods but with the Ola Money integration in the future build we made the transaction of money effortless as the Ola wallet would be deducted with the amount of the ride.

## VI. IMPLEMENTATION

The onCreate() module where the Main UI of the application rests. It gives the basic layout for taking input from the user by voice commands. It consists of

two EditText fields and two Buttons. The first button has an onClickListener() to get the current latitude and longitude from the GPS coordinates and converts this to an address and puts the data onto the first EditText field.

```
getLocationText.setText(location.getLatitude()+"
"+location.getLongitude());
```

The second EditText box is where the destination address can be typed or spoken. The second button is used to send these data to the Ola web api.

The Api module connects to the Ola web api to the application. It also passes the data like start address, drop address, start geographic coordinates, destination geographic coordinates, utm\_source provided to ola developers and the category of the ride.

```
@GET("/")
Call<ResponseBody> FindRide(
    @Query("lat") double latitude,
    @Query("lng") double longitude,
    @Query("category") String category,
    @Query("utm_source") int utm,
    @Query("drop_lat") double drop_lat,
    @Query("drop_lng") double drop_lng,
    @Query("dsw") String dsw
);
```

The retrofit module checks if the connection to the api is successful or not. This ensures that there will be no error when a booking is taking place.

```
Retrofit retrofit= new Retrofit.Builder()
    .baseUrl(BASE_URL)
    .addConverterFactory(GsonConverterFactory.create())
    )
    .client(new OkHttpClient())
    .build();
```

```
api= retrofit.create(Api.class);
```

The checkPermission module checks if the device has GPS enabled along with the internet. If neither of the two are on it displays a popup to turn on the GPS.

```
if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
```

```

!= PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION
)
!= PackageManager.PERMISSION_GRANTED) {
    if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
        requestPermissions(new String[]{
Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.INTERNET
}, 10);
        return;
    }
}

```

The configureButton module refreshes the current location every 5 seconds so that even if the user is on the move the address will get updated to the Api.

```

private void configureButton() {

    getLocationButton.setOnClickListener(new
View.OnClickListener() {
        @SuppressWarnings("MissingPermission")
        @Override
        public void onClick(View view) {

locationManager.requestLocationUpdates("gps", 0, 5,
locationListener);
        }
    });
}

```

The onLocationChanged module changes the latitude and longitude to the current locations geographic location. It also converts the geographic coordinates to the address.

```

addressList =
geocoder.getFromLocation(location.getLatitude(),
location.getLongitude(), 1);
String addressStr =
addressList.get(0).getAddressLine(0);
String areaStr = addressList.get(0).getLocality();

```

```

String cityStr = addressList.get(0).getAdminArea();
String countryStr =
addressList.get(0).getCountryName();
String postalcodeStr =
addressList.get(0).getPostalCode();

```

### A. Dependencies

The Gradle build system in Android Studio to include external binaries/ erstwhile library modules to the build as dependencies. The dependencies can be positioned on the machine or in a remote repository, transitive dependencies are automatically included.

```

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation
com.android.support.constraint:constraint-
layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation
'com.android.support.test:runner:1.0.2'
    androidTestImplementation
'com.android.support.test.espresso:espresso-
core:3.0.2'
    implementation
'com.squareup.retrofit2:retrofit:2.5.0'
    implementation 'com.squareup.retrofit2:converter-
gson:2.4.0'
    implementation "com.google.android.gms:play-
services-wallet:16.0.1"
}

```

The appcompat-v7 ensures that the lower level Api devices can access the features of the higher level Api devices. The retrofit turns your HTTP API into a Java interface. By using annotations we can describe the HTTP request:

- URL parameter replacement and query parameter support
- Object conversion - request body (example : JSON)
- Multipart request body and file upload



## VIII. REFERENCES

The annotations on the interface methods and its parameters indicate how a request will be handled. Every method must have an HTTP annotation to provide the request method ,relative URL. A request URL updated dynamically by means of substitute blocks and parameters on the method. A replacement block is an alphanumeric string bounded by opening and closing braces. A corresponding parameter to be annotated with @Path by means of the identical string. Retrofit is the class during which your API interfaces are curved into callable objects. By default, Retrofit to provide sane defaults and allows customisation. Retrofit will be able to barely deserialize HTTP bodies into OkHttp's Response Body type and it can only accept its Request. Body type for @Body. Converters can be added to support other types. Retrofit works at minimum Java 7 or Android 2.3.

## VII. CONCLUSION

E-Cab Service is a system that attempts to bring visually impaired people to the forefront of today's corporate world, as well as cater to their personal needs, by simplifying the process of booking cab services without having them rely on other people to book rides for their commutes.

This is an Android based app, that integrates voice based technology (TTS and STT) to make the process of booking rides easy and simple.

The visually impaired user merely speaks a few words and his/her ride is booked. The app integrates all the features of the Ola app, namely, creating an account, booking a ride, determining how far the user's ride is, showing the pickup and drop off locations, notifying the user how far the ride is and whether the cab has arrived or not, paying the driver and much more. This system can be enhanced in the future to add more features and tide over the gap between the visually impaired people with rest of the society.

- [1]. "The Impact of Mobility and Public Transport on the Independence of Visually Impaired People", Alicia Montarzano et al, Journal Visual Impairment Research The official publication of the International Society for Low-vision Research and Rehabilitation ISL Volume 9, 2007 - Issue 2- 3.
- [2]. "Public Transport Information System for Visually Impaired and Blind People" Michał Markiewicz, Michał Markiewicz, Marek Skomorowski, <https://www.researchgate.net/publication/226131391>
- [3]. "Audio Transportation System for Blind People", Jaime Sánchez, Márcia de Borba Campos, International Conference on Universal Access in Human-Computer Interaction
- [4]. UAHCI 2013: Universal Access in Human Computer Interaction. Applications and Services for Quality of Life
- [5]. "Mobile applications aiding the visually impaired in travelling with public transport", Piotr Korbel ; Piotr Skulimowski ; Piotr Wasilewski ; Piotr Wawrzyniak, 2013 Federated Conference on Computer Science and Information Systems
- [6]. "A comparative study of Ola and Uber customers in Mumbai"- Dr. Rupali Rajesh, Snehal Chincholkar
- [7]. "A study of passengers' motives for using mobile app based cabs" ,Dr. Mukund H. Khupse, Kaav International Journal of Economics, Commerce & Business Management
- [8]. A refereed blind peer review quarterly journal, KIJECBM/JUL-SEP (2017)/VOL-4/ISS3/A68
- [8]. "A review on speech to text conversion methods" Miss. Prachi Khilari, Department of E&TC Engineering. G.H.R.C.O.E.M, Ahmednagar, Savitribai Phule University of

Pune. Prof. Bhope V. P, Department of E&TC Engineering. G.H.R.C.O.E.M, Ahmednagar. Savitribai Phule University of Pune, International Journal of Advanced ,Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 7, July 2015.

- [9]. “Comparative study of different payment Gateway”, Monika Verma, Prof.Pankaj Jagtap, International Journal of Scientific & Engineering Research Volume 8, Issue 6, June-2017

**Cite this article as :**

Baswaraju Swathi, "Implementation of a Voice Based App for Booking Cab Services", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 3, pp. 639-648, May-June 2019. Available at doi : <https://doi.org/10.32628/CSEIT1953150>  
Journal URL : <http://ijsrcseit.com/CSEIT1953150>