**Rickshaw Application**

**Overview of App:**

# Rickshaw Driver App:- App developed by Rickshaw Driver for Rickshaw Drivers.

This App is for Rickshaw drivers. In which Driver can book ride by using three way

**1)Offline** – In offline mode , There are Two way of Ride

* **By Fixed Route** – In this mode , route is fixed from source to destination. Fare will be calculated by Google API.
* **By Dynamic Route** – In this mode , route is not fixed. Kilometer is calculated by application and send to server for calculating fare.

**2)Online -**

In this mode , driver will receive one or multiple booking ride by on screen notification. This notification driver need to accept within 30 second and after accepting ,app will show location of customer. After picking up customer , it will work like offline fixed route.

Customer can pay by cash or online payment. For online payment customer will get payment link by SMS.

**Users Of App-** Only Rickshaw drivers can use this Application**.**

**Class Used In Application-**

1)**Splash Screen**- Splash.class

on Splash screen, app ask permission to user (Storage, Camera, Location etc.) to use camera , GPS location etc.

**2)Login Activity- (**LoginActivity.class**)**

After Splash Screen it will naviagate to Login Activity. On Login Activity if user is not register then press sign up button it will naviagate to sign up Activity.

If user aleardy register he can login using mobile number and password.

We send device id and firebase id with mobile no. and pass. If firebase id is null then regenerate using FirebaseInstanceId.getInstance().getToken(). And stored in share pref.

We maintain separate shared preference (Constant.FIREBASE\_SHARED\_PREF\_NAME) for firebase.

We also check gps is on or off ,if off then checkLocationSetting() show popup to turn on GPS. If gps is off I will not login. Also if permission not allowed user can not login.

loginDriver() -This function invoke when user try to login

<http://tretakalp.com/driver_test/index.php/Api/Login/user_verify>

If response status is 3 then it will navigate to DocumentActivity to upload document.

Status 3 mean user did not upload the driving license.

If status = 1 then user uploaded all documents

If status =2 then user only uploaded driving license.

If user is already register then in onResume method we maintained session if its true then navigate to MapActivity.

If user click on signup it will check value from shared preference Constant.SIGNUP\_PROCESS if it is 0 then navigate to signUpActivity. If 1 means driver completed registration but he did not upload document so he will navigate to documentActivity.

3)**SignUpActivity-** SignUpActivity.class

On Sign up screen first getcity func call API to fetch cityList from server. Citylist will be attach to city spinner. User need to fill mobile number and other fields. When press next button it will first check mobile number ,device id and vehicle number is already register or not and also if user fill referral code then it will check it is valid or not.

If all condition is satisfy then it will show otp and user will get otp by msg.and otp will be verified and user registerd . and navigate to document page.

getCityList()- This Function will invoke first when user enters on this page. It gets list of city and rto code. And added into spinner.

getOtpVerification()- when user fill all field then first it will check mobile number, vehicle ,device id,number is valid or not.

3) **DocumentAcitivity Page**-DocumentActivity.class

This doucument page will be showed after signup and login acitivty on checking condition.

If user navigate from sign up then need to fill documents. If does not fill driver licence then user can not login in. He need to take photo of dl and upload to server.

Other document should be upload within 8 trial days. It will navigate to MapActivity mean home page.

If user did not upload any document(driving license) and go back to login and try to login then server give status 3 that will navigate to documentAcivity page.

When user want to upload document it will navigate to cameraActivity page. For bank related data upload it will naviagate to bankdetailsActivity.

4)**MapActivity** –MapActivity.class

This is App home page or Main Page after login or after signup->document->MapActivity.

MapActivity contains Navigation Drawer. When it opens. It will first open Home Fragment where google map is integrated. It will show users currnet location.

Home fragment will check isgps available or internet is available. If not then it will show dialoge box to the user.

**GetVersionCode()-**If user app is older then it will show popup to update app to the user.

**sendFirebaseKey()** - sending firebase to server if firebase does not exist on server.

Following two function will invoke once in a day.

1. **checkDriverCurrentStatus()** check driver activated or not if not then logout.
2. **addNotification()-** if user is on trial days then it will show notification.

In onResume app will start **LocationService** – it will work in background it will send location to server after 1 or 2 min. if driver kill app then it will stop service.

5)**HomeFragement-** HomeFragment.class

In this fragment google map is integrated and fused api is used. When map is ready it will call onMapReady. And added marker on map which represent driver or user position.

Driver’s marker color is in green color.

**Offline (STATIC Ride)-**

**i)By Fixed Route-**

If Customer will come to rickshaw driver and ask to start trip to specific destination then

Driver can search destination by entering destination. For searching india’s places SupportPlaceAutocompleteFragment is used.

By submtting it will call **getDistanceFare()** API which will get fare and distance and it will show popup to the driver.

There is start ride button after pressing it will ask customer mobile number. After entering cust mobile number server will send OTP to the customer and driver can start the ride. After start ride driver now onyl view endtrip and resend otp button.

Resend otp button is used when if cust does not reciceve otp.

navigation button will show naviagation route source to destination.

if cust want to end the ride then driver press end trip and before end trip it will ask otp from customer and after valid otp it will endtrip with entrip function name. On endtrip driver will get total amount it will navigate to RideSummaryActivity to complete payment transaction.

**Functionality:**

If driver started ride then app receives ride session code and other fields , this fields will store in shared pref for further use. It is a unique id that need to send when driver end ride.

**SupportPlaceAutocompleteFragment** will be hide. And endtrip button and resend otp buttons will be shown to driver.

After started the ride.. ISRIDE\_START value is set to 1. This value is always checked to ensure tht ride is not ended and if user came back from home screen of device to rickshaw app then in onResume or onMapready function it will check and change the view on the map, it means showing endtrip btn etc. Fetch the data from shared pref. and set to searchdestination fragement .

**ii)By Dynamic Route**-

This is same as fixed route ride. Driver has to press dynamic button to use this route.

In this type of ride ,customer will come to the driver and ask to go some specific destination

by dynamically route means no any fixed route required. The total distance calculated by driver app.. not by server. So for that, App will store distance after 20 sec of interval and calculate total distance when trip will be end , this total distance will be send to server . Then server will calculate Amount by total dist and total time and regenerate amount and send response to App.

Server Link used same as used in fixed route.

**Online Ride** –

In this ride type , driver will receive ride by firebase notification. When driver accept any one notification then related booking id, fare ,amount ,customer location, destination address, lat long will be store in shared pref.

After accepting notification it will navigate to mapActivty home fragment. Then **RIDETYPE**  status will be online .

Notification data will send to server for accepting notification **sendOnlineAcceptDataToServer()** Api for and send booking and receive session id.

by checking ‘online’ status and ACCEPT\_RIDE value it will hide SupportPlaceAutocompleteFragment and show three button

**1)starRide 2)Cancel Ride 3)Call 4)navigation button**

Cancel Ride will cancel ride by sending all data with session id and booking id before cancel ride **getCancelTripReason()** Func call API that api will fetch cancel reason list which will show in popup box. Driver need to select tht reason and press on submit **cancelTripRequest()** will call and cancelled the trip with tht session id and booking id and other data. After **cancelTripRequest()** executed successfully it will reset all view to start a new ride.

Navigation button will navigate route to customer position through google map

Call button is pressed will call customer so he can know what is exact position of customer.

**StartRide btn**- will start ride when driver will reach to customer. And customer will receive two otp –start otp and end otp by sms on cust mob no.

Driver will enter otp to start ride when he want to end trip driver will press end trip btn and Total amount will be received.

(Online Notification will be recived and show in incoming trip screen. It will describe on next page.)

**Database related:**

When driver start ride then data wil be inserted into trip\_ride(TABLE\_TRIPS) table.

driverId ,duration , amount , sessionId,bookingId,triptype,mobileno etc will be stored.

When trip end , need to update that same trip ride table using session id with total amount ,total distance and other data.

**RideSummaryActivity**- RideSummary.class

In this screen, when ride or trip end, total amount will receive from server and it will show on this screen . In this there is two option for make payment 1)online and 2) cash.

If user want to payment by online then he received one link for online payment after ride end. By this link he can end trip and driver will click online button ,if he get successful msg then payment is done it will navigate to paymentActivity and home fragment to start next ride.

If customer making payment by cash then driver will take cash and press cash button and trip will end with successful response.and it will navigate to paymentactivity and home fragment to start next ride.

On Cash or Payment button API makePayment() will call and get response and set RideType to null and Accept Ride status to 0,So driver can start ride.

If driver already accepted another online trip in ongoing trip then ACCEPT\_RIDE\_SECOND(Shared pref value) set to one and that will check then assign that second trip value to first data value. (note; First observe the code then you will understand). So when first trip end , second trip will be started immediately.

**Incoming Trip Screen-** TripNotification.class

In this screen , online driver will receive multiple notification. Notifications will be sent from web panel. Customer will contact to admin and customer location ,mobile number , destination address with this data notification generated and it will send to driver (who is online and under 2 km radius of customer location).

Driver will receive notification and screen open and driver need to accept any notification before 30 sec. other wise notification will gone. If other driver accept notification then that notification will gone from driver screen.

For receiving notification we implemented Firebase Push notification(Check instruction on internet how to integrate).

**MyFirebaseInstanceIDService.Class** will create firebase Id using onToken refresh that will send to server at time of login or signup or from homefragment.

**MyFirebaseMessagingServiceNew-** MyFirebaseMessagingServiceNew.class

It is like broadcast reciver will recive firebase notification and that will get data in json response. This data will be store in sqlite **insertIncentiveNotiData()** of database helper

If driver is login in then it will show notification other wise not.

Create notification and if driver is in the app in foreground and if incoming trip is not showing then startActivity to Incoming trip and show data in recyclerview fetching data from database

If incomingTrip is showing it will send it from local broadcast receiver and refersh the list.

To check incomingTrip screen is visible or not we manage Boolean value in shared pref.

**IS\_TRIP\_NOTIFICATION\_ACTIVE**.

If one driver accept ride then it will receiver json with booking id and status 0 it mean we have to delete that booking id trip from database and refresh recyclerview so second driver can not see that ride.

If one driver will accept one ride and cancel that same ride then that ride notification will receive again to second driver.

To show to list of trip **NotificationListAdapter** is used. And for showing countTimer used CountDownTimer.

After accepting notification or ride TABLE\_INCENTIVE\_NOTIFICATION will be deleted.

After accepting notification further process already describe above in ONLINE ride.

**All API Link -**

**TestLink**

<http://tretakalp.com/driver_test/index.php/Api/>

**LiveLink**

**Note: ALL test links below mentioned are same for Live links too except the 2 changes: 1. driver\_test will change to ickshaw and api version will change to new created version**

1)<http://tretakalp.com/rickshaw/index.php/Api1.3/>

2)http://tretakalp.com/rickshaw/index.php/Api1.4/

3) next link updated by accourding version

4) last version [http://tretakalp.com/rickshaw/index.php/Api1.9/](http://tretakalp.com/rickshaw/index.php/Api1.3/)

1)Constant Class

HomeFragment Contain following API

getDistanceFare()-

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/getStaticFare>

insertStartRide() –

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/startRide>

EndTrip()-

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/endTrip>

MakePyament()-

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/makePayment>

getCancelTripReson()-

<http://tretakalp.com/driver_test/index.php/Api/document/getCancelledReason>

resendOtp()-

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/resendOTP>

sendOnlineAcceptDataToServer()-

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/tripAccept>

2)MapActivity-

sendFirebaseKey()-

<http://tretakalp.com/driver_test/index.php/Api/login/changeFirebase>

**Profile Menu-** ProfileActivity.class

When click on drawer menu profile, Profile Activity will open. In this Screen, First it will fetch

All trip history and insert into sqlite table. In this screen , It shows profic pic of driver , todays total earning and name of driver.

**EditProfile**: EditProfileActivity.class -

If you want to edit profile press edit btn of profile activity it navigate to Edit Activity.

This screen will fetch data of city getCityList and getprofile data and show driver profile data.

There link of change password. If user click on that user will navigate to that page.

getCityList()-

<http://tretakalp.com/driver_test/index.php/Api/Login/findCity>

getProfile()-

<http://tretakalp.com/driver_test/index.php/Api/Login/profile>

**ChangePassword Activity:** ChangePasswordActivity.class

In this page, driver can change password.

Change Password Link:

<http://tretakalp.com/driver_test/index.php/Api/Login/changePass>

**Trip Histroy**: TripHistory.class

It open triphistory Fragment. In this screen, it will show the driver trip history according to date.

First it will call getAllTripEarnHistory() and store all history data of one month. If any data already exist it will delete and insert new data. It call only one time.

Driver can see data by selecting specific period from spinner. It will call getTripEarnHistory()

Function and get data which will show in expandable view.

It works in offline also. It will fetch data from sqlite if net is not available.

getAllTripEarnHistory():

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/trip_history>

if frmdate and toDate are last month date then it will fetch All data

getTripEarnHistory():

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/trip_history>

if frmdate and toDate are old then it will fetch All data

**Earning:**

EarningFragment class

In this menu, earning fragment , it contains earning and refer amount contains.

In earning , it will show the driver earnings according to date.

getEarning () will fetch earning data with from date and toDate parameter.

getEarning ():

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/earningDriver>

**ReferAmount:**

ReferEarningActivity.class

This screen show total refer count and refer amount that earned.

getReferData() will get total refer count and refer amount from server.

<http://tretakalp.com/driver_test/index.php/Api/login/getReferEarn>

**Current Balance Statement:**

PaymentDetailFragment class.

In this screen, drivers payment details will be shown according to date.

getPaymentDetails() will show data according to selected from and to date.

getPaymentDetails()

<http://tretakalp.com/driver_test/index.php/Api/document/paymentDetails>

**Refer and Earn:**

ReferEarnFragment class

It share invite code .

**Account Setting-**

AccountFragment class.

In this screen there are two menu 1) Documents 2)Language Setting

**i)Documents:-**

DocumentActivityNew.class

It shows screen same as document class like which we showed at time after sign up. But here is little difference.

In this screen , first it will call document data which uploaded during signup.

getDocument() will fetch that document data.

<http://tretakalp.com/driver_test/index.php/Api/document/getDocument>

ii)Camera Activty iii)BankDetails Activity same as which we use at sign up time

upload document data.

Both Class call

<http://tretakalp.com/driver_test/index.php/Api/document/add_document>

**Language Setting**-

LanguageActivity.class

This screen, change the language and update to the server.

changeLanguage(String langauge) will change language and update value on server.

<http://tretakalp.com/driver_test/index.php/Api/login/changeLanguage>

**Incentive and notification** -

*IncentiveAndNotification.class*

This fetch notification data

getIncentiveData()

<http://tretakalp.com/driver_test/index.php/Api/document/getIncentive>

Response will be save in sqlite.

If Incentive Push notification received it will store in sqlite and show to driver in this screen.

But if notification with same noti Id is present in sqlite then it will delete and store new notification in android.

**Report Issue:**

ReportIssueFragment.class

Driver can report an issue/ or query via this option he will generate a query, and a unique code will be generated. Admin can view all these reported queries and reply them required solution. Once the query is resolved , admin will close the query and driver wont be able to post any new comments on the same query

getHeaderQueryData() fetch main chat menu with query Id .

<http://tretakalp.com/driver_test/index.php/Api/query/getQueriesHeader>

in response get status . If status is 0 then disable reply box of chatActivity.

sendHeaderQueryData()-

send user new query to server.

<http://tretakalp.com/driver_test/index.php/Api/query/addQuery>

with empty query Id.

If driver click on any query of list then it will navigate to chatActivity

**ChatActivity Class-**

If status is 0 then disable reply box.

getQueryData() data and show in chat window.

<http://tretakalp.com/driver_test/index.php/Api/query/getQueries>

To send query use

sendReplyData() with queryId

<http://tretakalp.com/driver_test/index.php/Api/query/addQuery>

when get success response call getQueryData again to refersh list.

**About-**

AboutFragement.class show information of App.

**Help:**

HelpFragment.class

1)Topic one and Topic Two open same TopicTwo.Class.

Topic one shows Info in web view. And topic two shows video in web view.

Topic one and topic two this name is temporary . It will change in future.

Topic two URL - <http://tretakalp.com/driver_test/index.php/Api/video_app>

Topic one URL - <http://tretakalp.com/driver_test/index.php/Api/Document_link>

2)SignIn-

TopicTwo.Class.

Sign In URL- <http://tretakalp.com/driver_test/index.php/Api/sign_up>

3)Issue-

IssueActivity.class

getIssueData()

<http://tretakalp.com/driver_test/index.php/query/getIssues>

show issue list in English or Marathi

**Legal:**

LegalActivity.class

URL - http://tretakalp.com/termsandcondition

**DatabaseHelper.class-**

This class used to store data of app in sqlite. Rickshaw.db is the database name.

There are Four table used.

TripTable , NotificationTable, IncentiveNotificationTable and TripHistoryTable.

TripTable used to store trip related data.StartTrip data and EndTrip data.

insertTripData()- insert trip data when startTrip . relate to startTrip

updateTripTable\_withEndTripData- Update in Trip Table when driver press endTrip.

updateTripTable\_withPaymentMode()-Update Trip Table when payment done.

**Notification table** is used to store online trip notification details to show trips in listview. It will delete record from database when accept any trip or navigate from Tripnotification to mapactivity.class .

**TripHistory table** store one month of record of history. It is used to show data in offline mode. When tripHistory.fragment open that time data existed it will be deleted and new data stored in this table .

insertTripHistoryDetails()- Insert TripHistory Details

getTripDateBySpecificDateNew()- Get TripHistory Details By from and to date.

getTripDetailsByDateNew()- Get TripHistory Details By from and to date.

getTripCountDetailsNew()-get count of trip by date

getTodayTotalSumNew()-get todays earning

getTripIdData-to get trip Id of Trip

deleteAllTripHistory()- delete trip history data before calling insertTripHistoryDetails()

**IncentiveNotificationTable** store data to show in IncentiveNotification Fragment.

It will delete if data is exist.

insertIncentiveNotiData() insert data.

LocationService.class –

This is background service class that can get location details of user and calculate the distance for dynamic status. It used fused location API.

**Floating Widget** – Floating widget is attach in service when service start it creates floating widget. If Application is in background and Ride is started then it will show on device screen with rickshaw icon. If it is showing mean ride is started and user should open Application. If user click on this widget then it will navigate to Rickshaw application.

If user open app then , onResume() method of MapActivity.class it disable widget.

**System Configuration:**

Tool: Android Studio

Language: Android, Java, sqlite

**All API Links :**

**Note: ALL test links below mentioned are same for Live links too except the 2 changes: 1. driver\_test will change to ickshaw and api version will change to new created version**

Fetching Data from URL Link –

For Get Login

<http://tretakalp.com/driver_test/index.php/Api/Login/user_verify>

For Get Sign Up

<http://tretakalp.com/driver_test/index.php/Api/Login/signUp>

For Get OTP

<http://tretakalp.com/driver_test/index.php/Api/Login/signupOTP>

For Get Driver Profile

http://tretakalp.com/driver\_test/index.php/Api/Login/profile

For getting estimated fare calculation

http://tretakalp.com/driver\_test/index.php/Api/static\_calculation/getStaticFare

For getting Total amount and Total distance on End Trip

http://tretakalp.com/driver\_test/index.php/Api/static\_calculation/endTrip

For getting document status

<http://tretakalp.com/driver_test/index.php/Api/document/getDocument>

For getting Current status of driver-

<http://tretakalp.com/driver_test/index.php/Api/login/driverCurrentStatus>

For getting earning of driver-

<http://tretakalp.com/driver_test/index.php/Api/static_calculation/earningDriver>

For getting trip history details-

http://tretakalp.com/driver\_test/index.php/Api/static\_calculation/trip\_history

Getting Incentive Details-

http://tretakalp.com/driver\_test/index.php/document/getIncentive

Getting Payment Details-

<http://tretakalp.com/driver_test/index.php/document/paymentDetails>

Getting Sign webview-

<http://tretakalp.com/driver_test/index.php/sign_up>

Getting Topic one-

<http://tretakalp.com/driver_test/index.php/Document_link>

Getting Topic Two-

<http://tretakalp.com/driver_test/index.php/video_app>

Getting Title of Header query in Report Issue Screen-

<http://tretakalp.com/driver_test/index.php/query/getQueriesHeader>

Getting Issue-

<http://tretakalp.com/driver_test/index.php/query/getIssues>

Getting Refer and Earn Amount-

<http://tretakalp.com/driver_test/index.php/login/getReferEarn>

Sending Data URL Link -

Send OTP For verification-

<http://tretakalp.com/driver_test/index.php/Login/otpVarification>

Add document URL-

<http://tretakalp.com/driver_test/index.php/document/add_document>

Update Profiel-

<http://tretakalp.com/driver_test/index.php/login/profileUpdate>

To start ride-

<http://tretakalp.com/driver_test/index.php/static_calculation/startRide>

To end Ride-

<http://tretakalp.com/driver_test/index.php/static_calculation/endTrip>

Change Password-

<http://tretakalp.com/driver_test/index.php/Login/changePass>

Change Password-

<http://tretakalp.com/driver_test/index.php/static_calculation/resendOTP>

Insert Location Tracking –

<http://tretakalp.com/driver_test/index.php/tracking/insertTracking>

To cancel the accepted trip-

[http://tretakalp.com/driver\_test/index.php/ static\_calculation/cancelTrip](http://tretakalp.com/driver_test/index.php/static_calculation/resendOTP)

Change Language or driver-

[http://tretakalp.com/driver\_test/index.php/ login/changeLanguage](http://tretakalp.com/driver_test/index.php/static_calculation/resendOTP)

Forgot Password-

[http://tretakalp.com/driver\_test/index.php/ login/forgotPass](http://tretakalp.com/driver_test/index.php/static_calculation/resendOTP)

Change Firebase or update firebase -

[http://tretakalp.com/driver\_test/index.php/login/changeFirebase](http://tretakalp.com/driver_test/index.php/login/changeFirebase )

[APK And Project-](http://tretakalp.com/driver_test/index.php/ login/forgotPass  APK And Project-  Ricksha Path - /Users/Mac/AndroidStudioProjects   APK Path-   /Users/Mac/Desktop/Swapnil_Project_Documents/RickshawApp/APK )

[Ricksha Path - /Users/Mac/AndroidStudioProjects](http://tretakalp.com/driver_test/index.php/ login/forgotPass  APK And Project-  Ricksha Path - /Users/Mac/AndroidStudioProjects   APK Path-   /Users/Mac/Desktop/Swapnil_Project_Documents/RickshawApp/APK )

[APK Path-](http://tretakalp.com/driver_test/index.php/ login/forgotPass  APK And Project-  Ricksha Path - /Users/Mac/AndroidStudioProjects   APK Path-   /Users/Mac/Desktop/Swapnil_Project_Documents/RickshawApp/APK )

[/Users/Mac/Desktop/Swapnil\_Project\_Documents/RickshawApp/APK](http://tretakalp.com/driver_test/index.php/ login/forgotPass  APK And Project-  Ricksha Path - /Users/Mac/AndroidStudioProjects   APK Path-   /Users/Mac/Desktop/Swapnil_Project_Documents/RickshawApp/APK )

Dependency Classes –

1)LanguageActivity.class->>LogingActivity.class

2)SplashActivity->>LoginActivity.class

3) LoginActivity.class->>(For Sign Up)SignUpActivity.class

LoginActivity.Class->>(After Login) MapActivity.Class

LoginActivity.Class->>(After Login) DocumentActivity.Class

4)SignUpActivity.class ->> DocumentActivity.class->>MapActivity.Class

5)MapActivity.class->>HomeFragment

6)HomeFragment->>ProfileActivity->>EditProfileActivity.class->>ChangePassword.class

7)HomeFragment->>TripHistory.class

8)HomeFragment->>EarningmenuFragement ->>Earning.class-

9)HomeFragment->>EarningmenuFragement ->>ReferEarningActivity.class

10)HomeFragment->>PaymentDetailsFragment.class

11)HomeFragment->>ReferEarnFragment.class

12)HomeFragment->>AccountFragment.class->> DocumentActivityNew.class

13)DocumentActivityNew.class->>(DrivingLicense,AdharCard,Vehicle Permit,Vehicle Insurance)->>CameraActivity.class

14)DocumentActivityNew.class->>(ProfilePic)->>ProfilePicActivity.class

15)DocumentActivityNew.class->>(Bank Details)->>BankDetailActivity.class

16)HomeFragment->>AccountFragment.class->> LanguageActivity.class->>MapActivity.class

17)HomeFragment->>IncentiveAndNotificationFragment

18)HomeFragment->>ReportIssueFragment.class->>ChatActivity.class

19)HomeFragment->>AboutFragment.class

20)HomeFragment->>HelpFragment->>(For Topic one , TopicTwo,signUp)->>TopicTwo.class

21)HomeFragment->>HelpFragment->>(For issue)->>IssueFragment.class

22)HomeFragment->>LegalFragment.class

23)TripNotification.class->>MapActivity.class