

Travel-buddy

By: Arpit Shah and Shishir Bijalwan



April 28, 2017

android Project

Table of Contents

[Executive Summary 3](#_Toc481185867)

[Flow and Partitioning 4](#_Toc481185868)

[Interfaces 5](#_Toc481185869)

[Login Page 5](#_Toc481185870)

[Create Account 7](#_Toc481185871)

[Diary Recycler Page 9](#_Toc481185872)

[Detail View of Memory 13](#_Toc481185873)

[Create new Memory 15](#_Toc481185874)

[Attractions Recycle View 17](#_Toc481185875)

[Attraction Detail View 18](#_Toc481185876)

[Voice Communicator 20](#_Toc481185877)

[Landing Page 21](#_Toc481185878)

[Map Page 22](#_Toc481185879)

[Drawer 23](#_Toc481185880)

[Emergency Page 25](#_Toc481185881)

[Uses 26](#_Toc481185882)

[Pen down memories: 26](#_Toc481185883)

[Getting real time weather update 26](#_Toc481185884)

[Emergency Service 27](#_Toc481185885)

[Conversation 27](#_Toc481185886)

[Finding place to visit 27](#_Toc481185887)

[Future enhancements 27](#_Toc481185888)

[Text Recognizer 27](#_Toc481185889)

[Adding offline voice conversion 27](#_Toc481185890)

[Booking 27](#_Toc481185891)

# Executive Summary

“The gladdest moment in human life, me thinks, is a departure into unknown lands.” – Sir Richard Burton

Day-to-day life can become very repetitive and travelling is a chance to discover new places, languages, accents, foods, and ways of life. It opens your eyes to what is beyond your neighborhood. It's a chance to see with your own eyes landmarks that you've only seen in photos or heard of.  
  
Travelling can also a means of unifying people. For example, if you travel with friends the experience is likely to strengthen your friendship as you and your friends have lived through a unique experience together. We know how important these moments are for you, so we have come up with an idea which will help you live your dream too fullest and revile it latter at your choice.

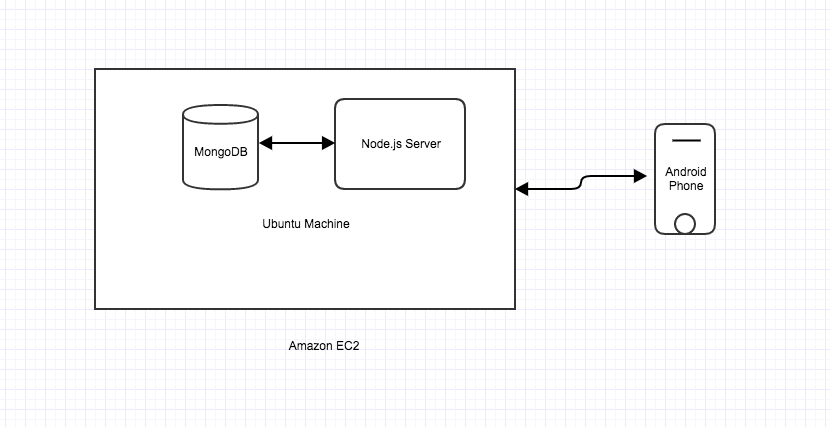
Most of the travel application in the market are just to make money by giving facilities like travel booking and reviews. None of them talks about some of the major needs while travelling:

* Communication Problem
* Emergency Situations
* Reliving memories

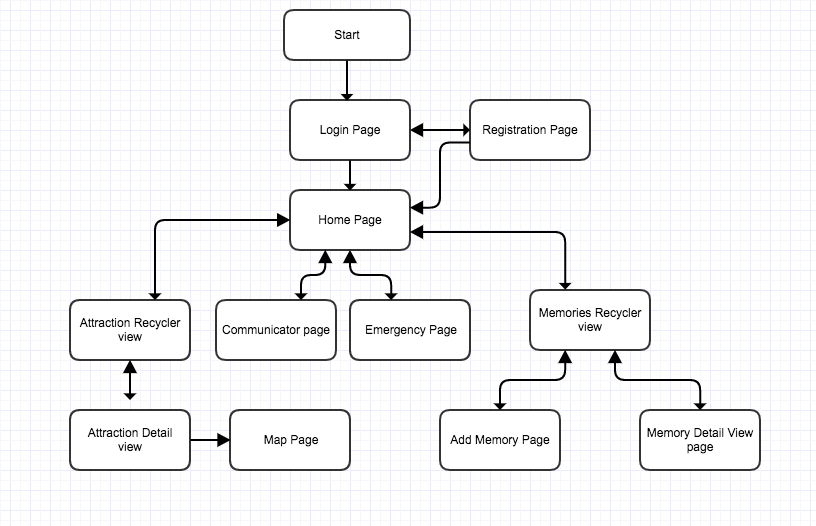
TravelBuddy is an application which takes care of the induvial not for profit but to help them to enjoy each and every moment of their travel.

# Flow and Partitioning

Our application system can be divided into three parts. The first two MongoDB and the node js server are present on Ubuntu machine on amazon EC2. The third part of our system is present in the android phone.

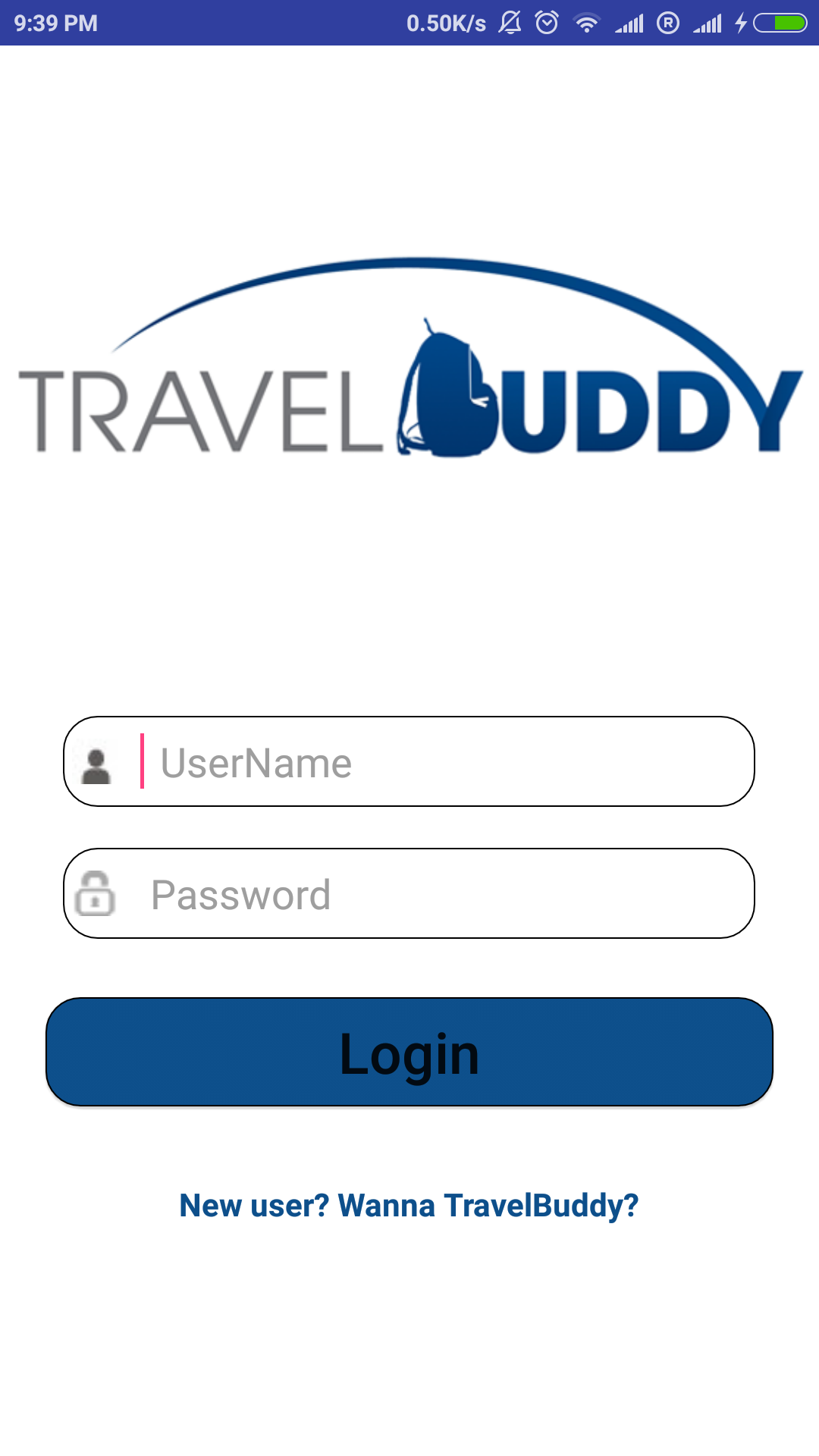
****

**System Distribution**

In the next image we have given a brief over view of the flow of our application ****

# Interfaces

## Login Page

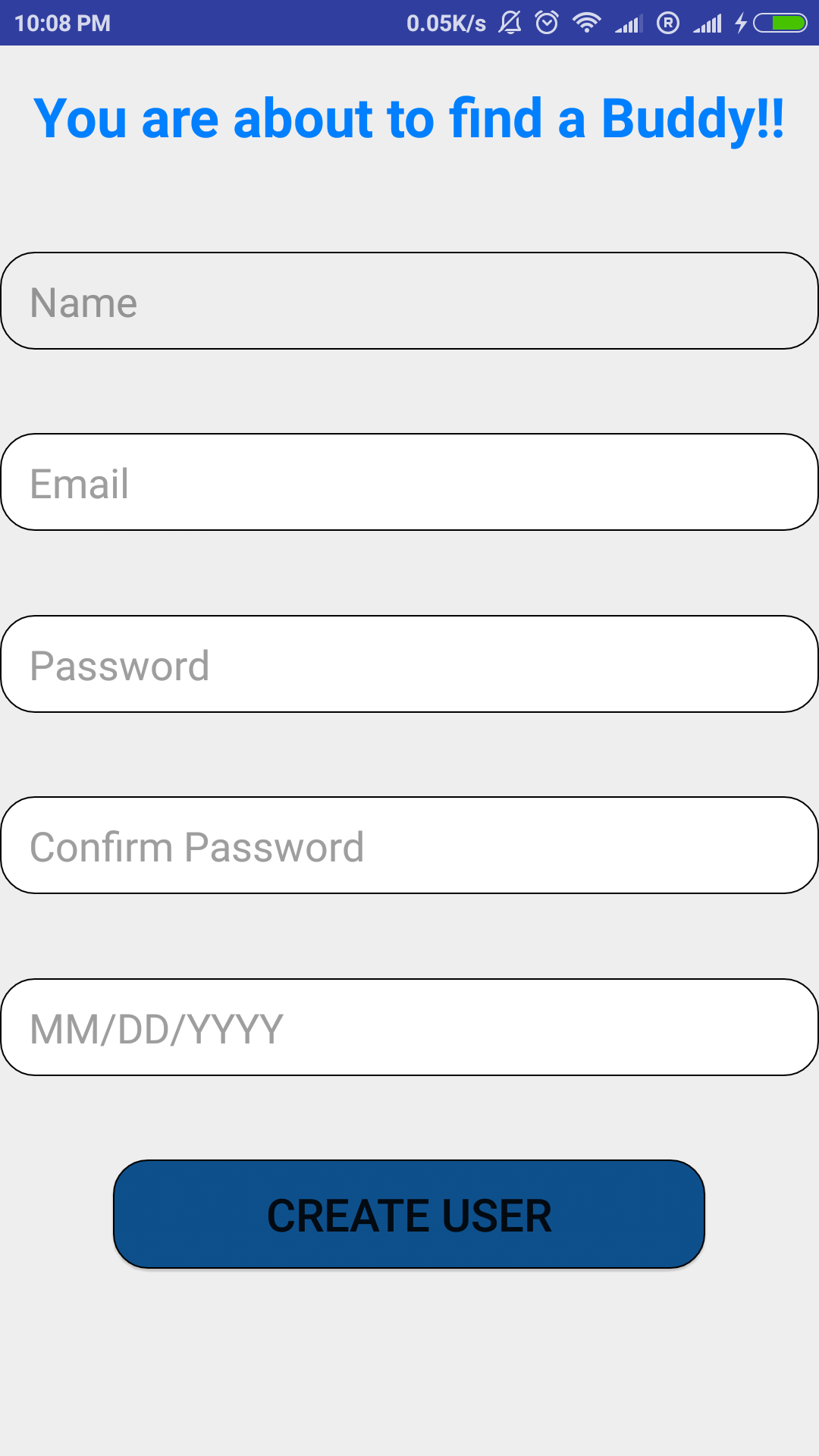


This is the first page that will pop up when user click on our application. Here the user has two options either to create a new account if using for first time or login if existing user. In case user doesn’t enter either the username or password, the application will alert the user to make the entry by vibrating. In case the user has entered a wrong password a message will be displayed at the bottom of the screen stating “invalid credentials”. If the combination of username and credential is correct we will take the user to the main page of the application.

Interactions:

* Login button: On click of login button if the credential is correct user will go to homepage of app
* Sign Up button: On click of sign up button the user will be taken to create account page.
* **class** getData **extends** AsyncTask<String, Void, Boolean> {  
    
    
   @Override  
   **protected** Boolean doInBackground(String... params) {  
    
    
   Boolean status = **user**.validateUseronLogin();  
    
    
   **return** status;  
   }  
    
   *// Get a string from an input stream* @Override  
   **protected void** onPostExecute(Boolean result) {  
   **super**.onPostExecute(result);  
    
   **if** (!result)  
   **resultTextview**.setText(**"Invalid credential"**);  
   YoYo.*with*(Techniques.***Flash***)  
   .duration(700)  
   .repeat(2)  
   .playOn(findViewById(R.id.***wrongCredentialTextBox***));  
   **if** (result) {  
   **resultTextview**.setText(**"You got it write"**);  
   startActivity(**new** Intent(MainActivity.**this**, LandingPage.**class**));  
    
   }  
    
   }  
  }

## Create Account



This page is for the users who are using the application for the first time. We take Name, emailed, password and date of birth as user input. If the both the password entered by the user don’t match, the user will be asked to enter the password again. Once the application has verified that the password is matching if will check in the database that the user with the email id provided is present or not. If the user is not present it will create a new user and move him to the view pager where he will get an overview of the application. In case the user email id is already present the user will be asked to create account with different user information.

Interactions:

* Create User: On click of this button if the email id is not registered already the user will be taken to application section to get overview of the application. If not will be asked to use another email id.

**boolean** createUser() {  
 **password**=**""**;  
 **boolean** check = validateUser();  
 **if** (check)  
 **return false**;  
 **else** {  
  
  
 HashMap sendObject = **new** HashMap();  
 sendObject.put(**"name"**, **tempName**);  
 sendObject.put(**"email"**, **tempUser**);  
 sendObject.put(**"password"**, **tempPassword**);  
 sendObject.put(**"DOB"**, **tempDate**);  
 sendObject.put(**"image"**,**null**);  
  
 JSONObject js = **new** JSONObject(sendObject);  
 **utility**.*sendHttPostRequestTwo*(**url** + **"newUser"**, js);  
  
 }  
  
  
 **return true**;  
}

## Diary Recycler Page



This page has been developed using Coordinate layout and provided advance features of coordinate layout such as collapse toolbar which has an image view when it is pulled down and when the user swipes down to see diary entry it collapses with just a text display and search option. It also provides a floating button option which is again a feature provided in coordinate layout.

This page id the center point where user get access to his old memories (Diary entries) and option to create new

Interactions:

* Card View: The user could go to the detailed view of each memory
* Add floating button: Will take user to create new diary entry page.

We have used coordinate layout with collapsing toolbar.

**public class** ViewHolder **extends** RecyclerView.ViewHolder {  
 **public** TextView **MemoryTitle**;  
 **public** TextView **MemoryDetail**;  
  
 **public** ImageView **menuPop**;  
 **public** ViewHolder(View itemView) {  
 **super**(itemView);  
 **MemoryTitle** = (TextView)itemView.findViewById(R.id.***movieName***);  
 **MemoryDetail** = (TextView)itemView.findViewById(R.id.***movieDesc***);  
  
 **menuPop**=(ImageView)itemView.findViewById(R.id.***imageButton***);  
 itemView.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (*mitemClickListener* != **null**) {  
 *mitemClickListener*.onItemClick(v, getPosition());  
 }  
 }  
 });  
 *//on long click* itemView.setOnLongClickListener(**new** View.OnLongClickListener(){  
  
  
 @Override  
 **public boolean** onLongClick(View v) {  
  
  
 **if**(*mitemClickListener*!=**null**)  
 *mitemClickListener*.onItemLongClick(v,getPosition());  
  
 **return true**;  
 }  
 });  
  
  
 }  
 **public void** bind( DiaryEntry diaryEntry) {  
  
 **MemoryTitle**.setText(diaryEntry.**Title**);  
 **MemoryDetail**.setText(diaryEntry.**data**);  
  
 }  
}

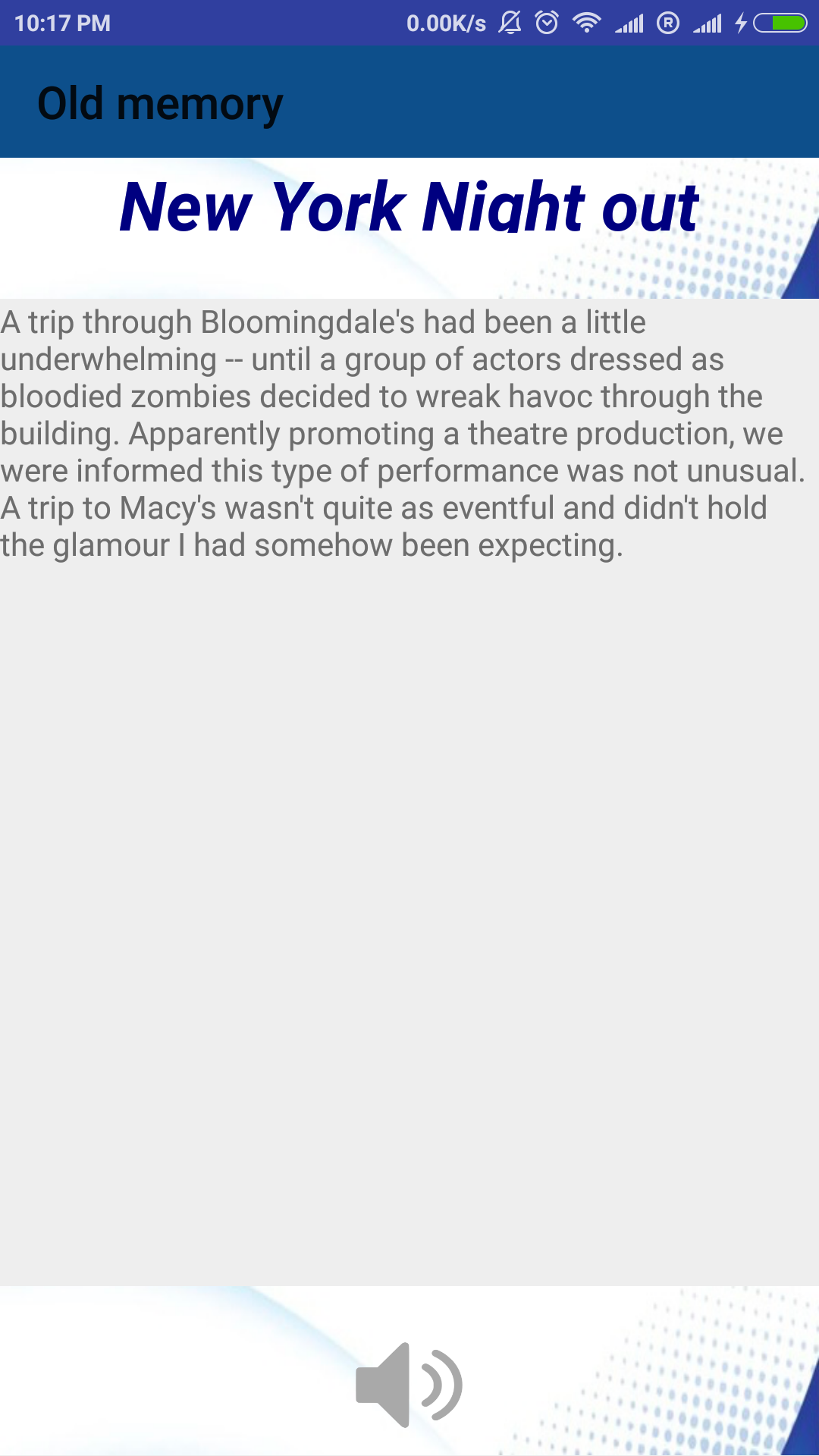
//functions

**public void** onCreateOptionsMenu (Menu menu , MenuInflater  
 inflater ) {  
 **if**( menu.findItem (R.id.***searchoption*** ) == **null** )  
 inflater.inflate (R.menu.***diaryentrymenu*** , menu );  
 SearchView search = ( SearchView )  
 menu.findItem(R.id.***searchoption***).getActionView();  
  
 **if**(search!=**null**){  
  
  
 search.setOnQueryTextListener( **new** SearchView.OnQueryTextListener(){  
  
  
 @Override  
 **public boolean** onQueryTextSubmit(String query) {  
**return true**;  
 }  
  
 @Override  
 **public boolean** onQueryTextChange(String newText) {  
 **return true**;  
 }  
 });  
 }  
  
 **super**.onCreateOptionsMenu(menu ,inflater);  
}

//Call back

*///* **public class** ActionBarCallBack **implements** android.view.ActionMode.Callback {  
 **int position**;  
 **public** ActionBarCallBack(**int** num){  
 **position**=num;  
 }  
  
 @Override  
 **public boolean** onCreateActionMode(android.view.ActionMode mode, Menu menu) {  
 mode.getMenuInflater().inflate(R.menu.***longclickmenu*** , menu );  
  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onPrepareActionMode(android.view.ActionMode mode, Menu menu) {  
**return false**;  
 }  
  
 @Override  
 **public boolean** onActionItemClicked(android.view.ActionMode mode, MenuItem item) {  
 **int** id= item.getItemId();  
 User user;  
 **switch** (id){  
 **case** R.id.***actiondelete***:  
 **new** deleteDiaryEntry(**position**).execute();  
mode.finish();  
 **break**;  
 **case** R.id.***actionsort***:  
 user=User.*getInstance*();  
 user.sortDiaryEntry();  
 *recycleAdapter*.notifyDataSetChanged();  
 mode.finish();  
 **break**;  
 **default**:  
 **break**;  
  
 }  
 **return false**;  
 }  
  
 @Override  
 **public void** onDestroyActionMode(android.view.ActionMode mode) {  
  
 }  
 }  
  
  
  
}

## Detail View of Memory

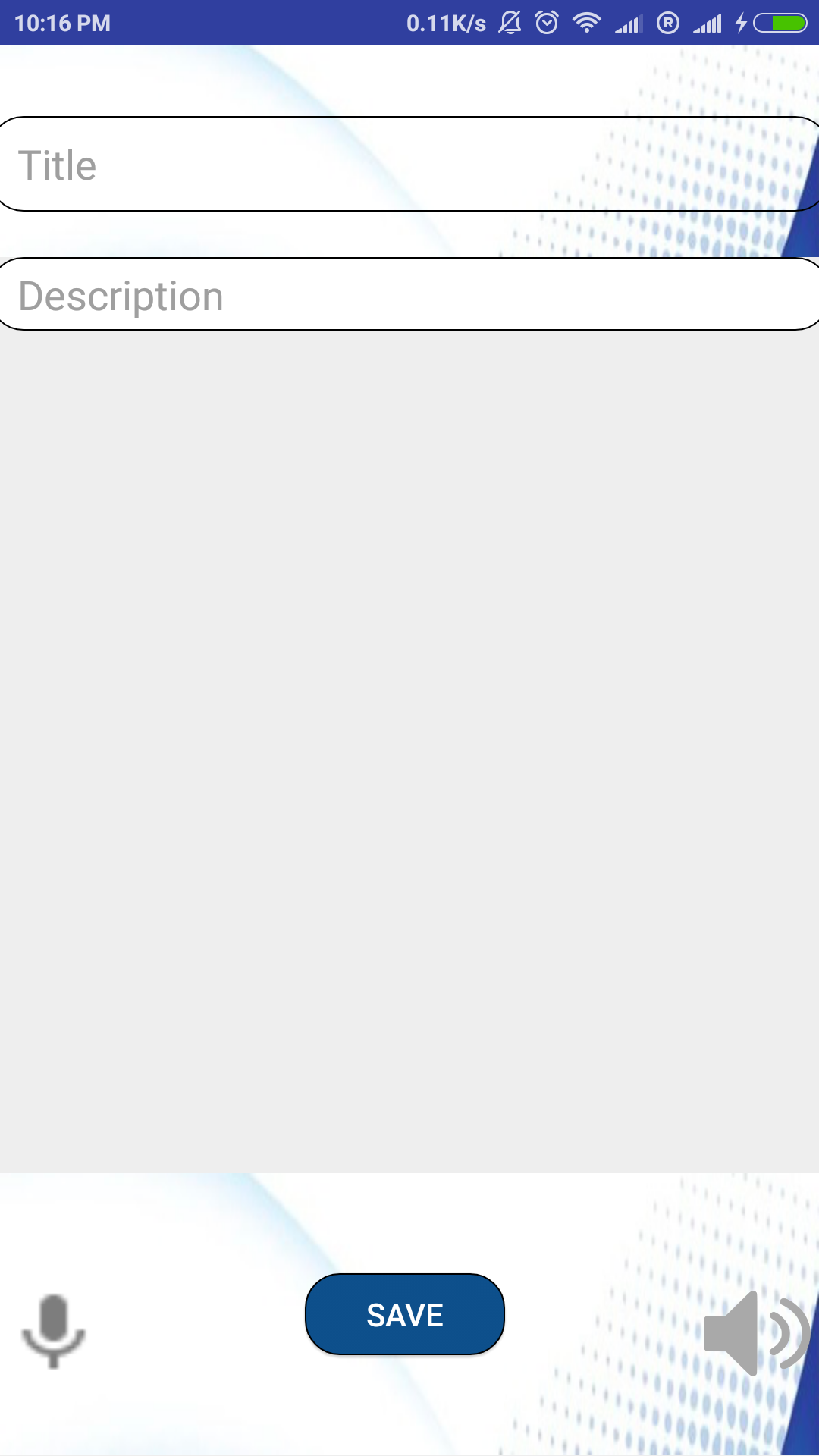


Once the user clicks on a memory in the recycler view, the user come to the detail view of that memory. Here user has two options one to read the diary entry made by him, or sib back eyes closed and relive those moments by listening while our apps speak it loud for the user.

Interactions:

* Speaker: On click of speaker button the user can listen the whole text
* @Override  
  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, **final** Bundle savedInstanceState) {  
   View rootView = **null**;  
   **diaryEntry** = (DiaryEntry) getArguments().getSerializable(**"DiaryEntry"**);  
   View view = inflater.inflate(R.layout.***diary\_details\_view***, container, **false**);  
   *//speak=(Button)view.findViewById(R.id.speakButton);* TextView title = (TextView) view.findViewById(R.id.***diaryTitle***);  
   **final** TextView detail = (TextView) view.findViewById(R.id.***edittextDiaryDetail***);  
   detail.setMovementMethod(**new** ScrollingMovementMethod());  
   title.setText(**diaryEntry**.**Title**);  
   detail.setText(**diaryEntry**.**data**);  
    
    
   Toolbar toolbar = (Toolbar) view.findViewById(R.id.***DiaryDetailToolBar***);  
   toolbar.setTitle(**"Old memory"**);  
   ((AppCompatActivity) getActivity()).setSupportActionBar(toolbar);  
   setHasOptionsMenu(**true**);  
    
   ImageButton speakButton = (ImageButton) view.findViewById(R.id.***speakerButtonDiaryDetail***);  
   speakButton.setOnClickListener(**new** View.OnClickListener() {  
   @Override  
   **public void** onClick(View v) {  
   **tt**.speak(detail.getText(), TextToSpeech.***QUEUE\_FLUSH***,**null**,**null**);  
   **tt**.speak(detail.getText(), TextToSpeech.***QUEUE\_FLUSH***,**null**,**null**);  
   }  
   });  
    
    
    
   YoYo.*with*(Techniques.***RotateIn***)  
   .duration(1000)  
   .repeat(1)  
   .playOn(view.findViewById(R.id.***edittextDiaryDetail***));  
    
    
    
   **return** view;  
  }

## Create new Memory



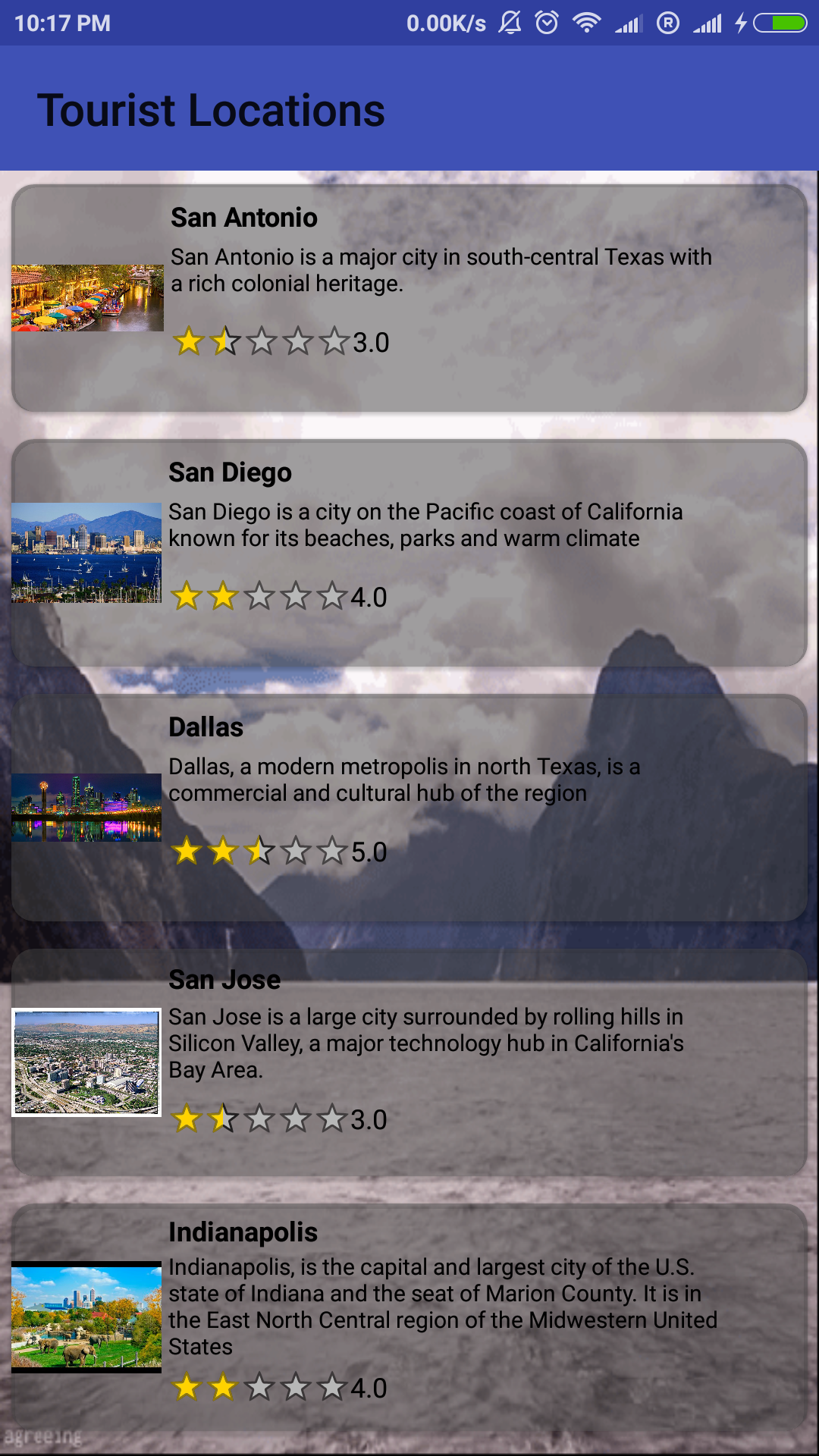
During early time when people didn’t use to have camera to capture moments, they used to pen down everything each night during their trip. Camera took the place of diary entry, but they could not give the exact feel after 5 years down the lane as nothing can explain the emotions better than putting them in words. So in our app we went to the vintage way of travelling and provided the user the option to pen down these memories with modern technology.

In this page user first need to set the title of the memory and then he can either type the details of the day or he can speak and our app will write about it.

Interactions:

* Mike: On click of mile button the user can start speaking and the it will get noted and displayed in the text box.
* Speaker: To speak loud the text.
* Save: To add the entry to database.

## Attractions Recycle View

****

* This view is used to give a list of attraction to the user and give search facility, so that they can find location to visit. To create this view, we are fetching information of attraction list from our cloud database using self created APIs.
* This view will have card view for each location which consists brief description of location, star rating and image of one of the good place.

Interactions:

* Card View: The user could go to the detailed view attractions. User can have lot more information from that detail view.
* Search: User can type the name and search the location instead of swiping through recycle view.

**public void** bind( Location location) {  
MyDownLoadAsyncTask task1= **new** MyDownLoadAsyncTask(**locationImage**);  
 task1.execute(location.**ImageUrl**);  
 **locationName**.setText(location.**locationName**);  
 **movieDesc**.setText(location.**Description**);  
 Double rating=location.**rating**;  
 **ratingBar**.setRating(rating.floatValue()/2);  
 **locationRating**.setText(rating.toString());  
 }  
}

## Attraction Detail View

****

When user click on one of the location of attraction recycle view, it will render the detail view which is much more informative about the location. It had one YouTube video, which will help you to have overall view of that place.

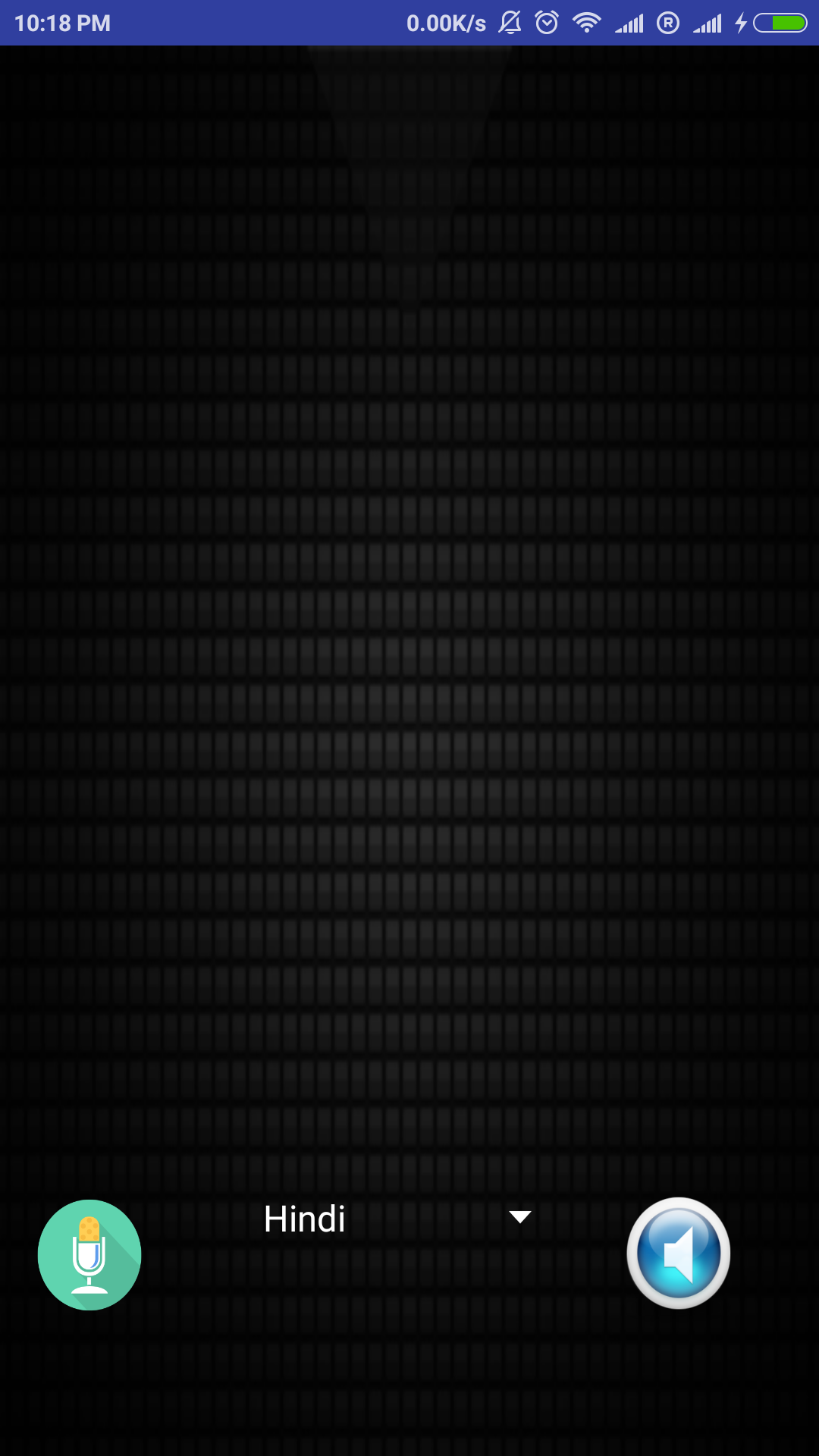
It also have description about the place, and start rating of that place. It have map button which will give user direction to that place with approximated time and distance.

Interactions:

* Map button: It will render map using Google Maps, and show the direction to the place from your current place. We are using latitude and longitude of current place and the place you want to go using GPS, and showing the direction, approximated time and distance using google maps.
* Youtube Video: We are showing a video on this page which will give some information about that place. We are storing the state of the video till that fragment get destroyed. Example like, if user go on map, the state of the video will get stored, but if user went back, video will turn from first frame.

@Override  
**public** View onCreateView(LayoutInflater inflater, ViewGroup container, **final** Bundle savedInstanceState) {  
 View rootView = **null**;  
 **location** = (Location) getArguments().getSerializable(**"Location"**);  
 View view = inflater.inflate(R.layout.***attractions\_details***, container, **false**);  
 String movieName = **location**.**locationName**;  
 String myvideokey=**location**.**VideoUrl**;  
  
  
 Toolbar toolbar=(Toolbar)view.findViewById(R.id.***attaractionDetailToolBar***);  
 toolbar.setTitle(**location**.**locationName**);  
 ((AppCompatActivity)getActivity()).setSupportActionBar(toolbar);  
 setHasOptionsMenu(**true**);  
 RatingBar ratingBar=(RatingBar)view.findViewById(R.id.***ratingBarDetails***) ;  
 Double rating=**location**.**rating**;  
 ratingBar.setRating(rating.floatValue());  
 TextView tMovieDesc = (TextView) view.findViewById(R.id.***movieDesc***);  
 tMovieDesc.setText(**location**.**Description**);  
 ImageButton direction=(ImageButton)view.findViewById(R.id.***deirectionbtnn***);  
 direction.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **mdirect**.directionButtonClicked(**location**.**Logitute**,**location**.**Latitue**);  
 }  
 });  
 **webView**=(WebView)view.findViewById(R.id.***webView***);  
 **webView**.getSettings().setJavaScriptEnabled(**true**);  
 **webView**.loadData(myvideokey,**"text/html"**,**"utf-8"**);  
 **web**=**new** WebChromeClient();  
 **webView**.setWebChromeClient(**web**);  
  
 LinearLayout lr= (LinearLayout) view.findViewById(R.id.***attractiondetaillayout***);  
 Bitmap bmImg=*decodeSampledBitmapFromResource*(getResources(), R.drawable.***backgroundone***, 200, 200);  
 BitmapDrawable background = **new** BitmapDrawable(bmImg);  
 lr.setBackgroundDrawable(background);  
  
 **return** view;  
}

## Voice Communicator



This is the super cool feature which will motivate the people to go to that type place which have their own language. People at that type of place usually don’t know other language except their national language.

User can set the language and can speak by holding one of the button, application do the rest for you and convert your spoken language to the selected language and speak it. So guys Good news for travelers! No communication barriers from now!

Interactions:

* Mic button: User can hold the button and speak whatever he/she want to speak to other people. Application will convert that spoken language to selected language and speak for you.
* Speaker button: If user want to pass the same message again, then user don’t have to speak again. User can use the speaker button and it will speak the same last spoken statements.
* Drop Down: User have privilege to select the output language, which will be used to convert the spoken language into selected language, and speak that converted language.

**private class** MyDownLoadJsonAsyncTask **extends** AsyncTask<String,Void,String> {  
  
 @Override  
 **protected** String doInBackground(String[] params) {  
 *// do above Server call here* **try** {  
 **InputString**= URLEncoder.*encode*(**InputString**,**"UTF-8"**);  
 } **catch** (UnsupportedEncodingException e) {  
 e.printStackTrace();  
 }  
 **OutputString**=Utility.*downloadJSONusingHTTPGetRequest*(**"https://www.googleapis.com/language/translate/v2?q="**+**InputString**+**"&target="**+**currentLanguage**+**"&source=en&key=AIzaSyAM9EVP4FeV1GOyqAprBsy-RkoT-Fgy-x4"**);  
 **try** {  
 JSONObject obj=**new** JSONObject(**OutputString**);  
 **OutputString**=((JSONObject)((JSONArray)((JSONObject)obj.get(**"data"**)).get(**"translations"**)).get(0)).get(**"translatedText"**).toString();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 System.***out***.println(**OutputString**);  
 **return OutputString**;  
 }  
  
  
 @Override  
 **protected void** onPostExecute(String message) {  
 **text**.setText(message);  
 **tt**.speak(message, TextToSpeech.***QUEUE\_FLUSH***,**null**,**null**);  
  
  
 }  
  
}

## Landing Page

****

It is a main Home Page of our application. This has navigation drawer attached to it which is pointing to the different functionality of the application. Landing page has quick access of used-to feature. User can have access to attraction facility, voice converter facility, diary handling facility from Home Landing page.

## Map Page



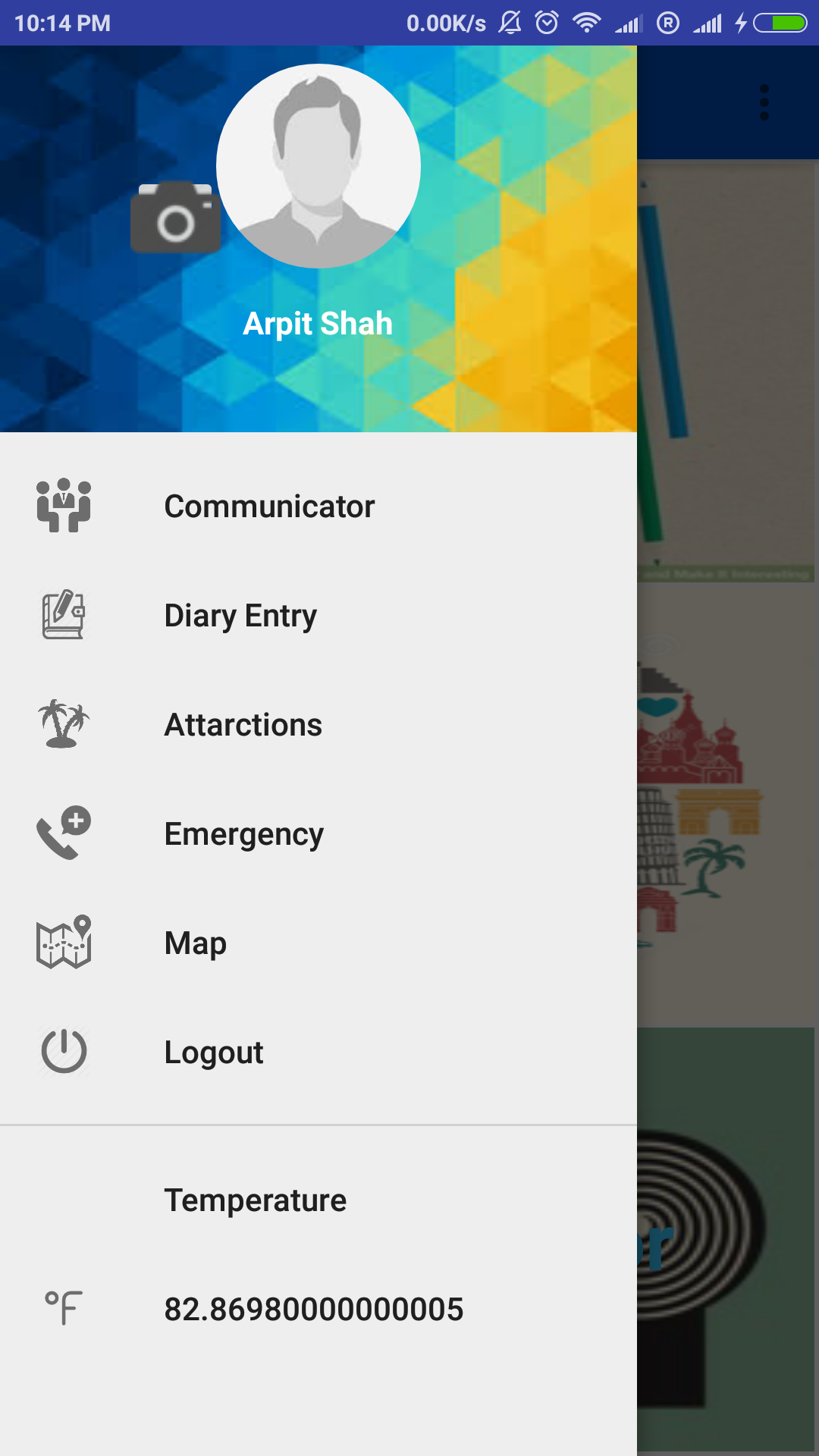
It is used to access google map service on this application. User can have quick access to google maps on this application without opening any other application. It will give user direction to the selected place with approximated time and distance.

Interactions:

* Find button: User can enter to and from address and have the direction, time and route details by clicking this button.

**private** String createUrl() **throws** UnsupportedEncodingException {  
 String urlOrigin = URLEncoder.*encode*(**origin**, **"utf-8"**);  
 String urlDestination = URLEncoder.*encode*(**destination**, **"utf-8"**);  
  
 **return *DIRECTION\_URL\_API*** + **"origin="** + urlOrigin + **"&destination="** + urlDestination + **"&key="** + ***GOOGLE\_API\_KEY***;  
}

## Drawer



It is quick access to all the feature of the application from navigation drawer including Log out button. User can quickly jump and use any feature using navigation drawer

It contains following items:

1. User image
2. User information
3. All the feature quick access
4. Temperature information of the current place

@Override  
**public boolean** onNavigationItemSelected(MenuItem item) {  
 Intent intent;  
 **switch** (item.getItemId()){  
 **case** R.id.***communicator***:  
 intent= **new** Intent(**this**,VoiceConverter.**class**);  
 startActivity(intent);  
 **break**;  
 **case** R.id.***diaryEntry***:  
 intent = **new** Intent(**this**,DiaryRecycleActivity.**class**);  
 startActivity(intent);  
 **break**;  
 **case** R.id.***attarctions***:  
 intent = **new** Intent(**this**,RecycleActivity.**class**);  
 startActivity(intent);  
 **break**;  
 **case** R.id.***emergency***:  
 intent = **new** Intent(**this**,EmergencyActivity.**class**);  
 startActivity(intent);  
 **break**;  
 **case** R.id.***Map***:  
 intent = **new** Intent(**this**,MapsActivity.**class**);  
 startActivity(intent);  
 **break**;  
 **case** R.id.***logout***:  
 intent= **new** Intent(**this**,MainActivity.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP***);  
 startActivity(intent);  
 **break**;  
 **default**:  
 **break**;  
 }  
 **drawerLayout**.closeDrawer(GravityCompat.***START***);  
 **return true**;  
}

## Emergency Page



This feature has been kept in the mobile for keeping an eye on your bad day. Every country has its own emergency numbers and a traveler might not be aware about it, our app keeps the emergency numbers updated for different countries. It uses the calling service of the phone to make the call.

**public void** policecall(View view) {  
 Intent callIntent=**new** Intent(Intent.***ACTION\_CALL***);  
 *//get data from user* callIntent.setData(Uri.*parse*(**"tel:"**+User.*getInstance*().**Policenumber**));  
 startActivity(callIntent);  
}  
  
**public void** embassycall(View view) {  
 Intent callIntent=**new** Intent(Intent.***ACTION\_CALL***);  
 *//get data from user* callIntent.setData(Uri.*parse*(**"tel:"**+User.*getInstance*().**Ambessynumber**));  
 startActivity(callIntent);  
}  
  
**public void** medicalcall(View view) {  
 Intent callIntent=**new** Intent(Intent.***ACTION\_CALL***);  
 *//get data from user* callIntent.setData(Uri.*parse*(**"tel:"**+User.*getInstance*().**Ambulancenumber**));  
 startActivity(callIntent);  
}

## Service

The service class will continuously run in background and get the location to get the temperature based on that location using a 3rd party Real data API.

**public int** onStartCommand(Intent intent, **int** flags, **int** startId) {  
 Timer timer = **new** Timer();  
 timer.schedule(**new** getLocationTask (), 0, 660000);  
  
 **return *START\_STICKY***;  
}  
  
@Override  
**public void** onDestroy() {  
 **super**.onDestroy();  
 *//stopping the player when service is destroyed*}  
  
*//helper class***class** getLocationTask **extends** TimerTask {  
 **public void** run() {  
 getcurrentLocation();  
 **new** MyDownLoadJsonAsyncTask().execute(**"hey"**);  
  
 }  
  
  
 **private void** getcurrentLocation(){  
 Geocoder geocoder;  
 String bestProvider;  
 List<Address> user = **null**;  
  
 GPSTracker tracker = **new** GPSTracker(BackGroundService.**this**);  
 **if** (!tracker.canGetLocation()) {  
 tracker.showSettingsAlert();  
 } **else** {  
 **lat** = tracker.getLatitude();  
 **lng** = tracker.getLongitude();  
 }  
 **Latitude**= String.*valueOf*(**lat**);  
 **Longitute**= String.*valueOf*(**lng**);  
  
 *// Toast.makeText(ServiceClass.this,"Latitue"+str+" "+str2,Toast.LENGTH\_SHORT).show();* }  
  
}  
  
  
  
  
  
**private class** MyDownLoadJsonAsyncTask **extends** AsyncTask<String,Void,String> {  
  
 @Override  
 **protected** String doInBackground(String[] params) {  
 *// do above Server call here* **OutputString**=Utility.*downloadJSONusingHTTPGetRequest*(**"http://api.openweathermap.org/data/2.5/weather?lat="**+**Latitude**+**"&lon="**+**Longitute**+**"&appid=44a65448b32f0ee01cbfa3b5ada9f1e1"**);  
  
 **try** {  
 JSONObject obj=**new** JSONObject(**OutputString**);  
 **OutputString**=((JSONObject)obj.get(**"main"**)).getString(**"temp"**);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 System.***out***.println(**OutputString**);  
 **return OutputString**;  
 }  
  
  
 @Override  
 **protected void** onPostExecute(String message) {  
  
  
 User ur= User.*getInstance*();  
 Double celciusVal= Double.*parseDouble*(message)-273;  
 System.***out***.println(celciusVal);  
 **double** d=(**double**)9/(**double**)5;  
 **double** a=(d)\* celciusVal.doubleValue();  
 **double** b=32;  
 **double** c=a+b;  
 ur.**temperature**=String.*valueOf*(c);  
 }  
  
}

# Uses

## Pen down memories:

The user can save his experience in the application by typing as well as speaking them. This feature helps the user to recollect those moments that will be washed with time and are hard to collect by seeing a photo. You can note how you felt at a place, note name of people whom you spoke, things that you liked, things that you didn’t.

## Getting real time weather update

People who like to travel, pack their bags without any plan where they are headed. Sometimes they even don’t know the name of the place from where they are pass. In big mobiles like iPhone user has to select the place to know the temperature over there. Our application does it for you. It has a background service running which keeps the track of the place you are at and provides weather update to you in the drawer.

## Emergency Service

This service has been kept in mind specifically for the people who like to travel abroad. This feature has been kept in the mobile for keeping an eye on your bad day. Every country has its own emergency numbers and a traveler might not be aware about it, our app keeps the emergency numbers updated for different countries. So user doesn’t need to worry about it and help it available at a call to him.

## Conversation

User can use this feature to communicate with the people who don’t know each other’s language. This feature enable all the places where residents are rigidly following their own language. Using this feature, different language people can communicate easily.

## Finding place to visit

User can find good place to visit using our attraction database. It will be very easy to find and pick good location using our highly informative views and nourished database.

# Future enhancements

## Text Recognizer

One feature which we wanted to give to our application but could not give because of the time constraint is text recognizer. We want to give the facility to the user where he can click photo of any product in market and he will get information about that product in his local language or he can use it to read boards in country of which he doesn’t know the language.

## Adding offline voice conversion

Currently our app is using google translator API to convert language. But what happens if he is at a place which does not have the availability of internet. To help user in these type of scenarios, we want to provide this enhancement.

## Booking

We are planning to join area like hotel booking, travel booking with this application which enable user to book their full journey with travel aide.