Tour guide mobile application

by

Gor Chobanyan

Theory of Economics, Yerevan State University, 2012 - 2016

A thesis submitted in partial satisfaction of

the requirements for the degree of

Master of Science

in

Computer & Information Science

in the

COLLEGE OF SCIENCE AND ENGINEERING

of the

AMERICAN UNIVERSITY OF ARMENIA

Supervisor: ___________________________

Signature: ___________________ Date:_____________

Committee Member: ____________________

Signature: ___________________ Date:_____________

Committee Member: ____________________

Signature: ___________________ Date:_____________

Committee Member: ____________________

Signature: ___________________ Date:_____________

Licenses for Software and Content

Software Copyright License (to be distributed with software developed for masters project)

Copyright (c) <2018> <Gor Chobanyan>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(This license is known as "The MIT License" and can be found at http://opensource.org/licenses/mit-license.php)

Content Copyright License (to be included with Technical Report)

LICENSE

Terms and Conditions for Copying, Distributing, and Modifying

Items other than copying, distributing, and modifying the Content with which this license was distributed (such as using, etc.) are outside the scope of this license.

1. You may copy and distribute exact replicas of the OpenContent (OC) as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the OC a copy of this License along with the OC. You may at your option charge a fee for the media and/or handling involved in creating a unique copy of the OC for use offline, you may at your option offer instructional support for the OC in exchange for a fee, or you may at your option offer warranty in exchange for a fee. You may not charge a fee

for the OC itself. You may not charge a fee for the sole service of providing access to and/or use of the OC via a network (e.g. the Internet), whether it be via the world wide web, FTP, or any other method.

2. You may modify your copy or copies of the OpenContent or any portion of it, thus forming works based on the Content, and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified content to carry prominent notices stating that you changed it, the exact nature and content of the changes, and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the OC or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License, unless otherwise permitted under applicable Fair Use law.

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the OC, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the OC, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Exceptions are made to this requirement to release modified works free of charge under this license only in compliance with Fair Use law where applicable.

3. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to copy, distribute or modify the OC. These actions are prohibited by law if you do not accept this License. Therefore, by distributing or translating the OC, or by deriving works herefrom, you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or translating the OC.

NO WARRANTY

4. BECAUSE THE OPENCONTENT (OC) IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE OC, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE OC "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK OF USE OF THE OC IS WITH YOU. SHOULD THE OC PROVE FAULTY, INACCURATE, OR OTHERWISE UNACCEPTABLE YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

Contents

Abstract

The goal of this project to create an Android application which will help tourists during their travel to Armenia. It provides them with necessary information about places they want to go. As the most of the touristic attractions are in the regions where locals don’t speak English, it contains Armenian phrases which will help tourists to interact with people. The application is able to serve as a guide and give tourists historical information about monuments and other famous attractions. It also shows the location of the monuments in the map. The application is able to get users location and show attractions near to user. Was implemented login/register and logout functionalities with connection with the database.

Introduction

As tourism is one of the main factors of development of Armenian economy, it is important to have an application which will help tourists during their travel to Armenia. It should provide them with necessary information about places they want to go, places of hotels and restaurants, interesting events, weather information, etc. As the most of the touristic attractions are in the regions where locals don’t speak English, it should also contain Armenian phrases which will help tourists to interact with people. The application should also be able to serve as a guide and give tourists historical information about monuments and other famous attractions.

The users of the application include people who are the active android smartphone or tablet users, who enjoy traveling and experiencing new things, but most commonly people who do not sure where to go in Armenia.

Let’s consider some similar mobile applications.

Tourist Guide Information System using Google Map and GPS [1]

This is a tourist guide mobile application which is using Google Map and GPS to find information about the places near the user location. The app is working for Yangon City. For finding users location the application uses Trilateration method which is an equation for navigation. The program consists of three components: mobile device, server, and database.

All the information is stored in the database. When the app starts it checks whether network and GPS are enabled. When the current location has defined the data about relevant location items is downloaded. Then the user may input place to view details and app show information about that place. All the calculation of the distance between the location items and the current location are done on the server by the Haversine Theory.

Pros of this app are that it has the server for database and SQLite database for offline mode. Also, it works with Google Maps and includes pictures of the places included in the app.

Interactive Mobile Based Tour Guide [2]

The researchers of this project tried to create an interactive tour guide application. The main features of this application are:

- it provides the user the possibility of doing a virtual tour
- it has a voice-based information provider which will serve as a guide to tourists while they are looking at the surroundings
- it can locate the user
- users can select a path on the map to their preferred destination

The system has two components the web server and the mobile application. The user interacts with the mobile application and the server provides the required information.

The application has “Map view” and “Camera view”. The map view provides 3 options: path selection, Information provider, Virtual tour. The path selection function takes users input for start and end locations and returns all the existing paths for reaching the destination. The information provider function gives the user all the information about the selected place including the audio description. The virtual tour function gives the user the possibility of taking a virtual tour by providing a 360-degree view of the location.

The camera view uses Augmented Reality (AR) which is a technique for visualizing virtual information where the physical reality is part of the scene. For creating a location-based augmented reality application the system should use the mobile’s camera, compass, GPS and the Accelerometer.

This application provides virtual information of the places and it works with web servers. The disadvantage of the project is for providing a 360-degree view of the location app should use a huge amount of data stored in the database and it may influence application performance.

From the above-mentioned examples, we decided to use several features. Our app will use Google Maps and will get user’s location with GPS. We also decided to use Google Firebase API as it provides easy authentication process for users. It also provides analytics and error reports for developers.

Requirements

User requirements

The application has only one type of user.

Users should be able to:

- register/login into the system
- search for Armenian places by inputting the name
- view the tour objects categorized by popularity.
- rate the tour objects
- view places near to the user's location
- view Armenian popular phrases, their translation into the English and listen to the pronunciation of the phrases
- view the specified location on the map

System requirements

Functional requirements

If a user wants to register into the system a system will provide a registration form to him/her. The data that user will input will be added to the database. If the user wants to login he/she will input the username and password, the system will check if there is a matching entity in the database if there is the user will login into the system.

User Interface should provide a search field for searching a place where the user should input a name of the place. The application will send the information to the database system which will find the entity with matching name and will return it to the application which will represent the information in the user interface.

The user interface should contain a menu from which user can choose an option to view tour objects by specific categories and then clicking the menu item should bring the user a new screen with specified information. The menu items should provide following functionalities

- If the user wants to see popular Armenian phrases and their translations into English. The application should open a new window containing phrases.
- If the user wants to see the popular places in Armenia the system will search in the database for the tour objects with highest grades. The list of results will be shown in the new screen.
- If the user wants to view the places near him he/she will click on the corresponding menu item. The application should open a new window which will contain places near the user.

  There should be a menu item for log out

The application must have a function of grading a place.

Non-functional requirements

- The database of tour objects must be up-to-date. New tour objects must be added occasionally.
- The information of the database should quickly load into the application. This process shouldn't take much than 0.5 seconds.
- Onclick functions of the buttons must do the required process in less than 0.5 seconds.
- Source code should be readable.
- The application should use a minimum possible amount of memory
- The database attributes should be understandable for other programmers.

Program structure and techniques

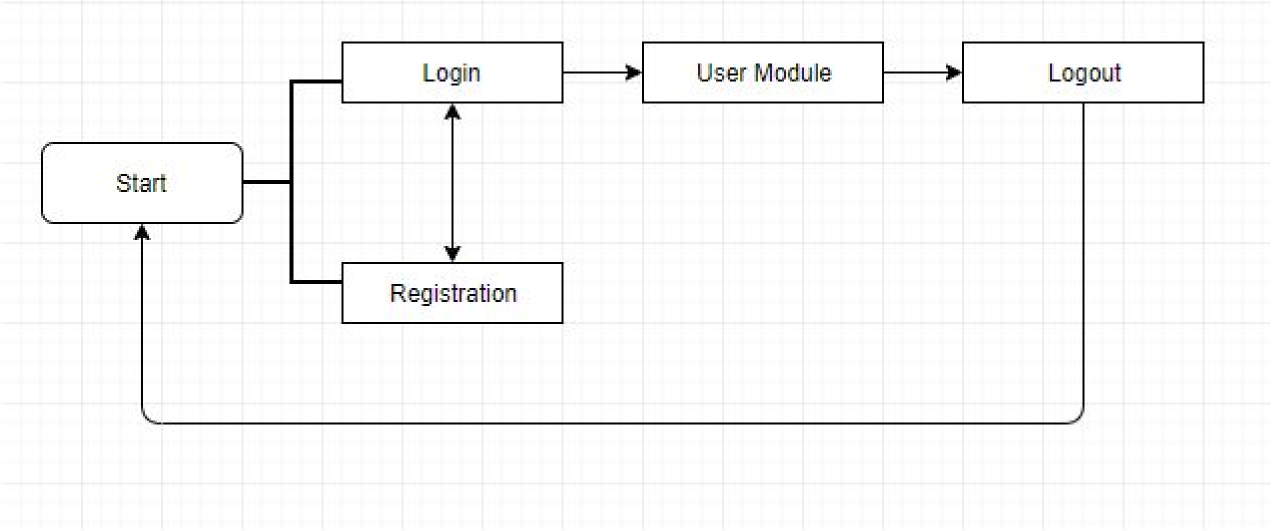
Program structure

For representing the main structure of the program we will use flowcharts. A flowchart is a graphic and schematic representation of the logic or steps in a program or system. It represents how a program or activity moves through various processes or program routines. It uses symbols to represent the activities and uses arrows to represent the direction of activity through the processes. Flowcharts can define the behavior of a single program or a system (a combination of programs).[4]

Main Module is a top-level flowchart(picture 1). The user starts from the ‘Start". If the user doesn't have an account he/she starts the registration process. Otherwise, the user skips registration process and logs into the system. The login screen is provided to him/her. It should contain two text input fields one for username, another for the password. After successfully login User Module is provided to him/her. Before user leaves, he/she can Logoff logoff is not required. The user can repeat this process again.

If the user closes the program without logging off after reopening application will remember the last used account.

Picture 1

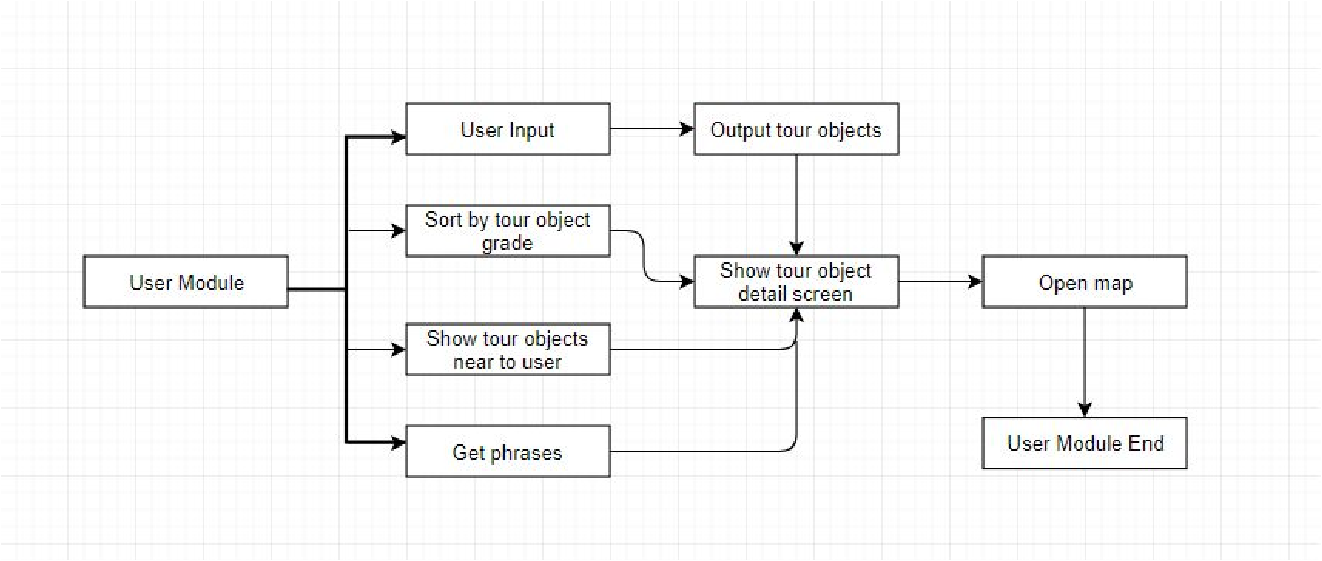


In the Picture 2 User Module is represented. It is a sub-module of the Main Module. From the User Module, user can choose search functionality where he/she will input a name of a tour object. The program will connect to the database and provide user input. The database will search a tour object which name matches with the input and return it. The output will be shown to the user.

The user can choose Sort by tour object grade state where a list of places provided to the user which are sorted by their grades.

If the user chooses near me option the application will show only tour objects which are located near to users current location. The list of objects will be downloaded from the database.

Picture 2


User Module
User Input
Sort by tour object grade
Show tour objects near to user
Get phrases
Output tour objects
Show tour object detail screen
Open map
User Module End

In the "Get Phrases" state the list of phrases in Armenian and English will be provided to the user. Also, the pronunciation of the phrases will be provided.

From the "Output tour objects", "Sort by tour object grade" and "Sort tourist objects near to user" states where list of places is provided to the user by clicking on the item of the list user will go to the "Show tour objects detail screen", In this state detailed information of the selected place in Armenia will be represented. Detailed information will include the name, the grade and the picture of the place and information about the place. There should be a button for the map. If the user clicks that button application will go into open map state where a map will be shown to the user and the location of the place.

Techniques used for the program implementation

Real-Time Databases

For this project, we have decided to use a real-time database which is popular for mobile development. Let's consider advantages and disadvantages of two of such databases which are Google firebase API and Mongo DB. These two databases return JSON objects.

Google Firebase API.

Advantages of this database system are

- easy authentication process where firebase gives access to the application only to authorized users. The login can be done by Gmail, Twitter, Facebook, etc.
- Analytics: The application developer can get information about who are active users or what feature of your application is used most.
- Firebase is a cloud-based database. Even if the connection is lost data is maintained
- Crash reporting: Firebase creates error reports and groups them by their severity.

Disadvantages of this database are

- Firebase is free with the limitation of 100Mb memory. For using memory higher than 100Mb should be paid.
- It supports offline operation but if a client goes offline and then comes online it may lead to inconsistencies.
- It uses queues operations between servers and prevents concurrency

Advantages and disadvantages of Mongo DB are

Advantages

- Mongo DB can consolidate any sort of information.
- It can analyze data and provide you with results in real time.
- Mongo DB is free and can run on Linux.

Disadvantages

- It doesn’t support joins.
- The data size created with Mongo DB is usually large [3].

Google Firebase API and Mongo DB have their advantages and disadvantages but for this project, we have decided to use Firebase due to the advantages mentioned above. It stores data as a JSON object and gives it to the program. In the Firebase, two kinds of objects are created. One for storing user information, another for storing data about places in Armenia. User object will contain username and password fields. Objects about places will contain several fields: name of the place, grade of the place, description, and location of the place which consists of two components latitude and longitude.

Android techniques

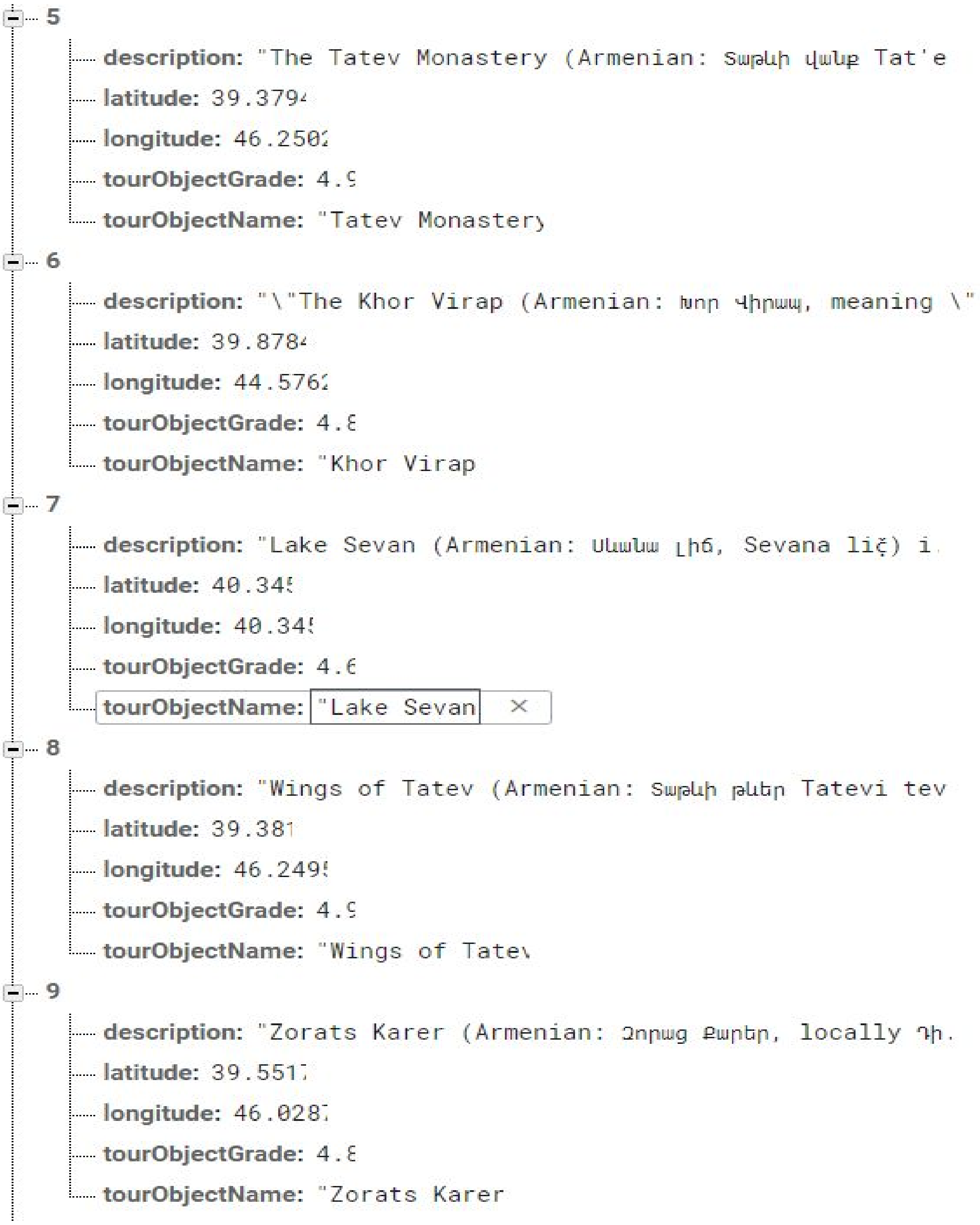
In Android development, view components are used for showing information on the screen. Activities and fragments are used for creating this application. Activity [6] is the main screen for any android project and it has its own lifecycle. The structure of the screen we created for the project is entered into the onCreate() method where activity is created. There are three activities in the project for the login screen, for the main screen and the last one for showing a map. Fragments are used for creating flexible UI [7] for the Main screen and the Map screen. Fragments represent part of the user interface [8]. More than one fragment can be visible in a single screen. In the main activity, fragments are used to show a list of places to the user. If part of UI (user interface) is changed and the different one should be shown on the screen only the fragment is replaced by another fragment but the activity stays unchanged. A menu is attached to the main activity as it provides main functionality and should be on the screen when other components of the screen are changed. All the UI components which should be changed occasionally are attached to the fragments.

Implemented application and testing

Testing

For testing the application Huawei Honor 4c with Android 5.1.1 platform and Android phone emulator was used for running the program. 20 attractions in Armenia are entered to the Firebase database as a dummy data to test program functionality. Each tour object in the database has five values with corresponding keys (Picture 3) which are description, name, grade, and location (latitude and longitude) of the tour object.

Picture 3



```
5
    description: "The Tatev Monastery (Armenian: Տաթևի վանք Tat'e
    latitude: 39.3794
    longitude: 46.2502
    tourObjectGrade: 4.9
    tourObjectName: "Tatev Monastery
6
    description: "\"The Khor Virap (Armenian: Խոր Վիրապ, meaning \"
    latitude: 39.8784
    longitude: 44.5762
    tourObjectGrade: 4.8
    tourObjectName: "Khor Virap
7
    description: "Lake Sevan (Armenian: Սևանա լիճ, Sevana lič) i
    latitude: 40.345
    longitude: 40.345
    tourObjectGrade: 4.6
    tourObjectName: "Lake Sevan
8
    description: "Wings of Tatev (Armenian: Տաթևի թևեր Tatevi tev
    latitude: 39.381
    longitude: 46.2495
    tourObjectGrade: 4.9
    tourObjectName: "Wings of Tatev
9
    description: "Zorats Karer (Armenian: Զորաց Քարեր, locally Դի.
    latitude: 39.5517
    longitude: 46.0287
    tourObjectGrade: 4.8
    tourObjectName: "Zorats Karer
```

Implemented Application

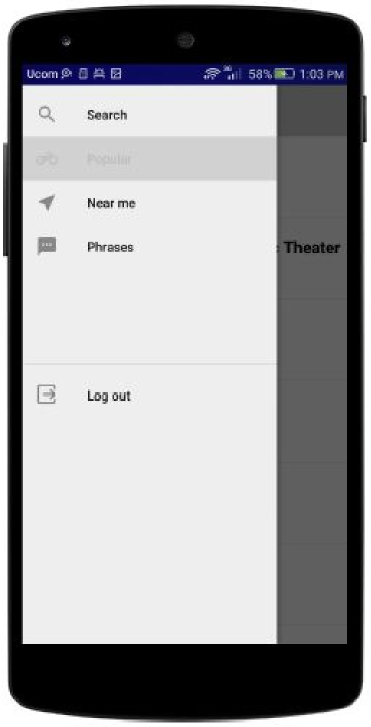
When the program starts, a login screen is provided with two buttons one for logging in with created account another for using a google account. In the registration, screen user enters an email, password and a name.

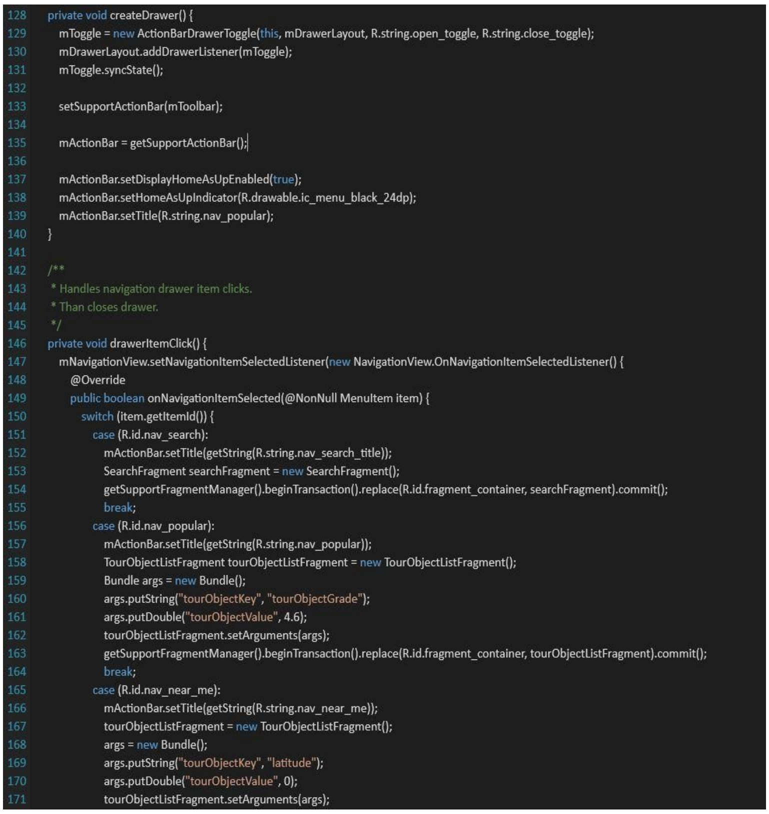
After registration, a new screen appears on the phone. It is the main activity where the main menu is attached. Other parts (for example list of tour objects, phrases, etc.) are in the fragments which are attached and replaced by the main activity.

In order to open main menu user can click on the home button or swipe a finger from right to left.

The main menu contains 5 item (picture 4) Search, Popular, Near me, Phrases and Log out.

Picture 4

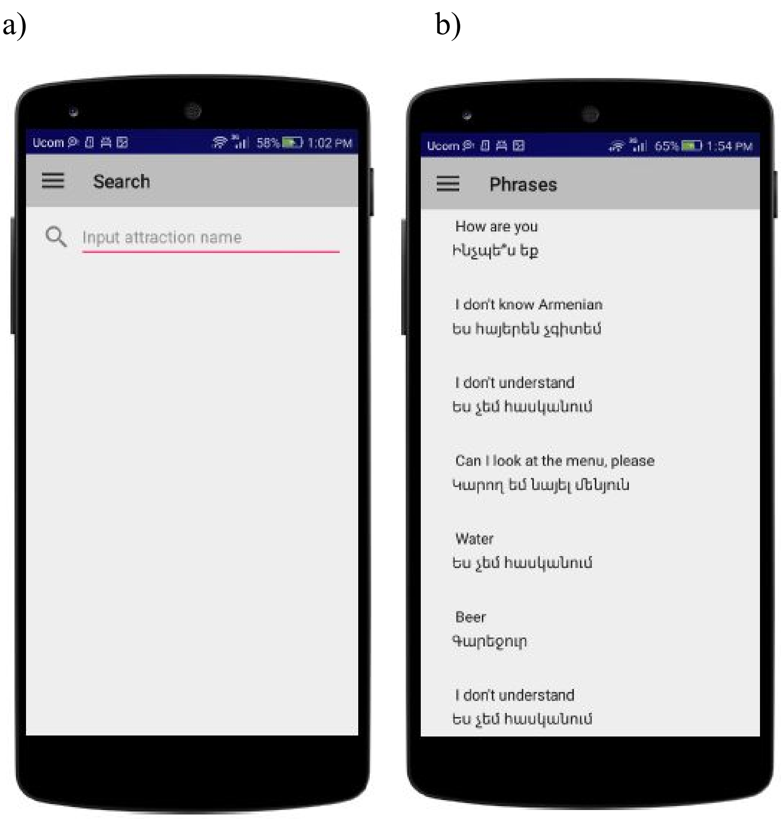




```java
private void createDrawer() {
   mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, R.string.open_toggle, R.string.close_toggle);
   mDrawerLayout.addDrawerListener(mToggle);
   mToggle.syncState();

   setSupportActionBar(mToolbar);

   mActionBar = getSupportActionBar();

   mActionBar.setDisplayHomeAsUpEnabled(true);
   mActionBar.setHomeAsUpIndicator(R.drawable.ic_menu_black_24dp);
   mActionBar.setTitle(R.string.nav_popular);
}

/**
 * Handles navigation drawer item clicks.
 * Than closes drawer.
 */
private void drawerItemClick() {
   mNavigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
      @Override
      public boolean onNavigationItemSelected(@NonNull MenuItem item) {
         switch (item.getItemId()) {
            case (R.id.nav_search):
               mActionBar.setTitle(getString(R.string.nav_search_title));
               SearchFragment searchFragment = new SearchFragment();
               getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, searchFragment).commit();
               break;
            case (R.id.nav_popular):
               mActionBar.setTitle(getString(R.string.nav_popular));
               TourObjectListFragment tourObjectListFragment = new TourObjectListFragment();
               Bundle args = new Bundle();
               args.putString("tourObjectKey", "tourObjectGrade");
               args.putDouble("tourObjectValue", 4.6);
               tourObjectListFragment.setArguments(args);
               getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, tourObjectListFragment).commit();
               break;
            case (R.id.nav_near_me):
               mActionBar.setTitle(getString(R.string.nav_near_me));
               tourObjectListFragment = new TourObjectListFragment();
               args = new Bundle();
               args.putString("tourObjectKey", "latitude");
               args.putDouble("tourObjectValue", 0);
               tourObjectListFragment.setArguments(args);
```

When the user clicks on one of the Search, Popular, Near me, Phrases items corresponding fragment is attached to the main activity. Search fragment (Picture 5 a)) contains a field where the user can input attraction name and a tour object with that name will be returned to the screen. In the Phrases fragment, a list of popular phrases is presented to the user (Picture 5 b)). Every Armenian phrase contains a translation into the English.

Picture 5



How are you
Ինչպե՞ս եք

I don't know Armenian
Ես հայերեն չգիտեմ

I don't understand
Ես չեմ հասկանում

Can I look at the menu, please
Կարող եմ նայել մենյուն

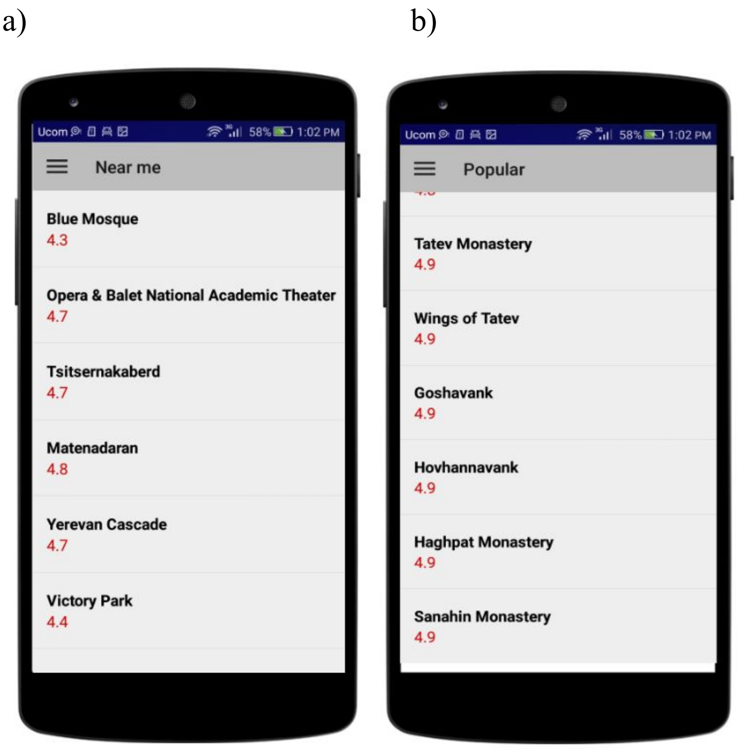
Water
Ես չեմ հասկանում

Beer
Գարեջուր

I don't understand
Ես չեմ հասկանում

In Picture 6 Near me and Popular screens are represented. Near me and Popular fragments contain a list of tour objects (name and the grade of the attractions in Armenia). In the Popular screen list of the places are sorted by their grades. In the Near me screen is presented places that are not far than approximately 1.3 km from the user's current location.

Picture 6



Blue Mosque
4.3

Opera & Balet National Academic Theater
4.7

Tsitsernakaberd
4.7

Matenadaran
4.8

Yerevan Cascade
4.7

Victory Park
4.4

Tatev Monastery
4.9

Wings of Tatev
4.9

Goshavank
4.9

Hovhannavank
4.9

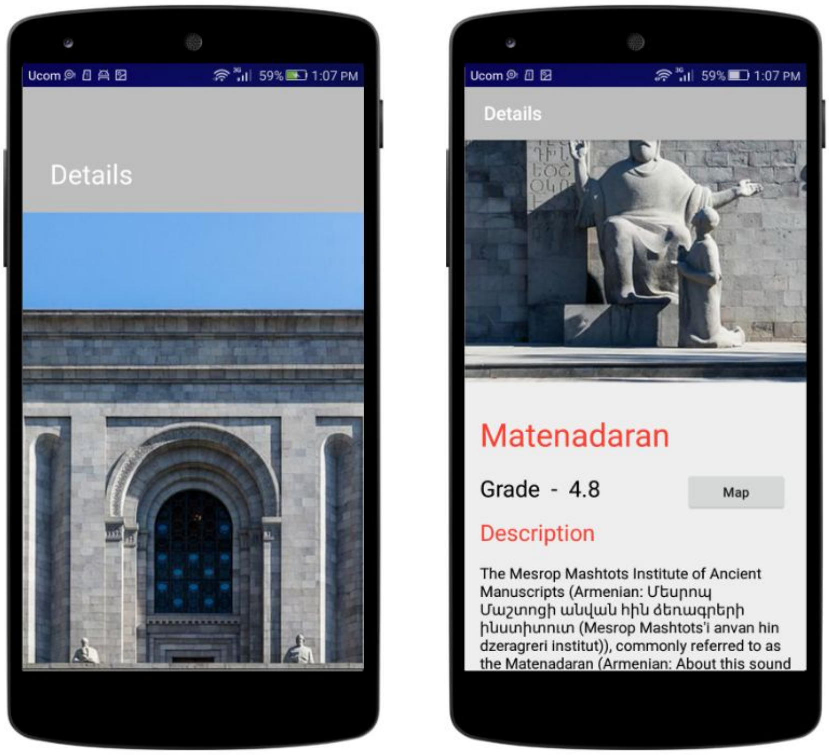
Haghpat Monastery
4.9

Sanahin Monastery
4.9

When the user clicks on the one attraction from the provided list a new scrollable activity starts (Picture 7). It contains the name of the attraction, its grade, picture, detailed information and the Map button. When the user clicks on the map button a map appears on the screen where the location of the attraction is shown (Picture 7 b)).

Picture 7

a)



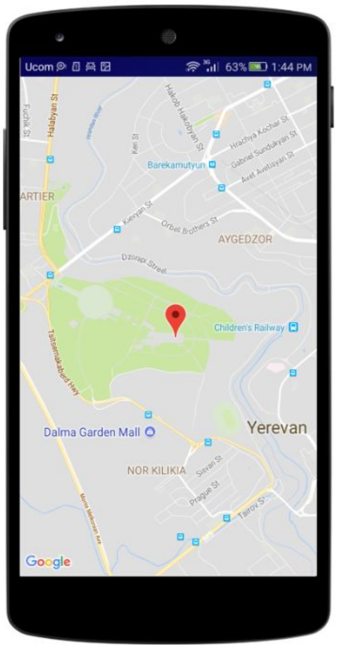
Matenadaran

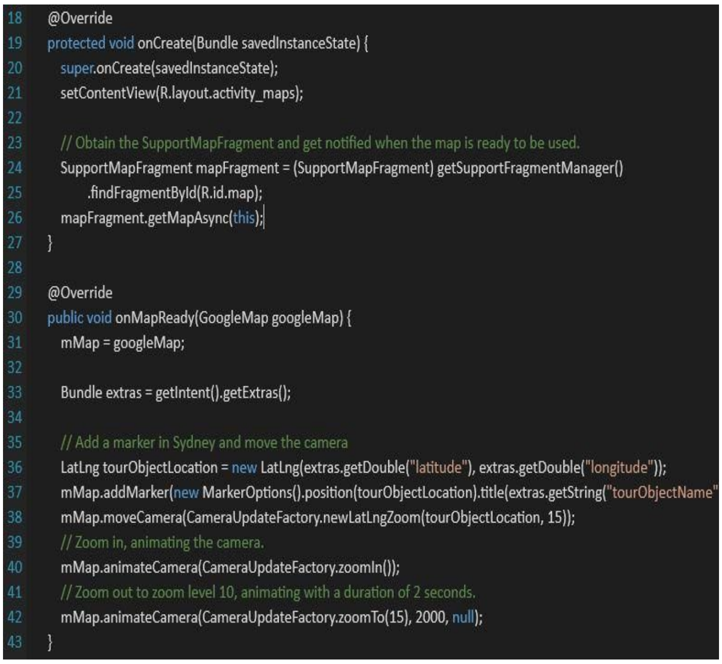
Grade - 4.8

Description

The Mesrop Mashtots Institute of Ancient Manuscripts (Armenian: Մեսրոպ Մաշտոցի անվան հին ձեռագրերի ինստիտուտ (Mesrop Mashtots'i anvan hin dzeragreri institut)), commonly referred to as the Matenadaran (Armenian: About this sound

b)





```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    Bundle extras = getIntent().getExtras();

    // Add a marker in Sydney and move the camera
    LatLng tourObjectLocation = new LatLng(extras.getDouble("latitude"), extras.getDouble("longitude"));
    mMap.addMarker(new MarkerOptions().position(tourObjectLocation).title(extras.getString("tourObjectName"
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(tourObjectLocation, 15));
    // Zoom in, animating the camera.
    mMap.animateCamera(CameraUpdateFactory.zoomIn());
    // Zoom out to zoom level 10, animating with a duration of 2 seconds.
    mMap.animateCamera(CameraUpdateFactory.zoomTo(15), 2000, null);
}
```

Conclusion

As tourism is one of the main factors of development of Armenian economy, it is important to have an application which will help tourists during their travel to Armenia which is the goal of this project. It provides them with necessary information about places they want to go. As the most of the touristic attractions are in the regions where locals don't speak English, it contains Armenian phrases which will help tourists to interact with people. The android application is able to serve as a guide and give tourists historical information about monuments and other famous attractions. It also shows the location of the monuments in the map. The application is able to get users location and show attractions near to user. Was implemented login/register and logout functionalities with connection with the database.

Future works

In the future, it is going to be added functionality for users to upload objects about places in Armenia by filling required fields (attraction name, description, etc.). Uploaded objects will be added to the database but will be marked as unread in order to differentiate from the existing ones. The administrator will check if the input is valid. If it is valid the unread mark will be removed otherwise that object itself will be deleted from the database.

Grading functionality for the user will be implemented in the future.

Another task is to show users location on the map and draw a path from user to the specified location.

References

[1] Honey Soe, Myint Myint Sein / Tourist Guide Information System using Google Map and GPS

[2] M.U.E.Wijesuriya, S.U.Mendis, B.E.S.Bandara, K.P.Mahawattage, N.Walgampaya, D. De Silva / Interactive Mobile Based Tour Guide

[3] Sonam Khedkar, Swapnil Thube / Real Time Databases for Applications

[4] "Flowchart" https://en.wikipedia.org/wiki/Flowchart

[5] Official Firebase webpage https://firebase.google.com/docs/database/android/start/

[6] Activities in Android https://developer.android.com/guide/components/activities/intro-activities

[7] Building a flexible UI https://developer.android.com/training/basics/fragments/fragment-ui

[8] Fragments https://developer.android.com/guide/components/fragments