SmartWays - path planning app

Marius Olariu,
Andrei Muresanu,
Politehnica University of Timisoara,
E-mail: mariuslucian.olariu@gmail.com

Contents

1 Introduction

Long time ago people used physical maps to get directions but nowadays as we all know, software runs the world! The next step was marked by sites which provided directions like MapQuest, but now in the dawn of the smartphone era people use map apps to get turn-by-turn directions. The best-known apps are **Google Maps**(all modes of navigation), **Waze**(focused on driving experience) or **Apple Maps**.

SmartWays is a path planning app focused on suggesting driving directions to users. It represents a laboratory project for the *Mobile Systems and Applications* course at *Politehnica University of Timisoara*. To us it seemed an interesting project to develop because there are a lot of concepts that we acquired and could be applied, like path finding algorithms or design patterns for the architecture of the app. Also we tried to dive into the world of mobile applications development by practicing the development of a real mobile app.

2 Design and implementation

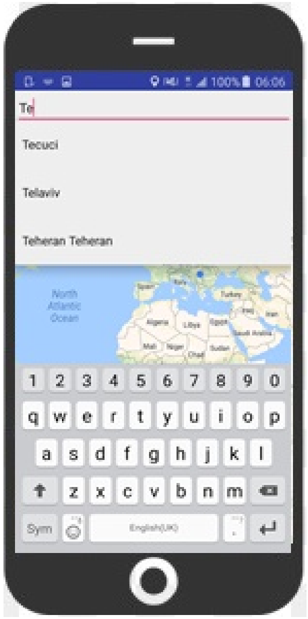
We did not try to reinvent the wheel when implementing this app, thus we used technologies provided by Google. Namely we used *Google Maps Api* for rendering the map and for interacting with it. The interaction is done by using an instance of /emphGoogleMap. However, while you can obtain directions from point A to B using the Directions Api from Web Services in form of a json/xml file this feature is missing in Google Maps Android Api. It is so because Google wants us to *"Innovate, but don't duplicate"* and it is strictly forbidden to create a substitute for **Google Maps**. However, since at this stage the project is used for educational purposes and we do not intent to publish it we implemented the functionality to obtain the directions and the drawing of paths on map. In order to obtain the directions we issues a request using http with the following format:

```
String request = DIRECTIONS_URL + params + GOOGLE_MAPS_KEY
```

- DIRECTIONS_URL - is a constant url
- params - a map representing the options specified by the user, e.g ¡origin, Timisioara¿, ¡destination, London¿ etc.
- GOOGLE_MAPS_KEY - a key that is used in all calls to Google Maps Api

After parsing the json response we obtain a series of informations about the available paths, among these being a list of Latitude (Lat) and Longitude (Lng) points for each path. The Lat&Lng points are afterwards used to draw the paths on map.

Whenever a path is returned we save the origin & destination locations in a list representing suggestions that will be displayed to user when searching for another route. The origin and destination locations were also supposed to be saved in firebase for the data to persist on other devices that share the same Google Play account (feature not working atm).



Te

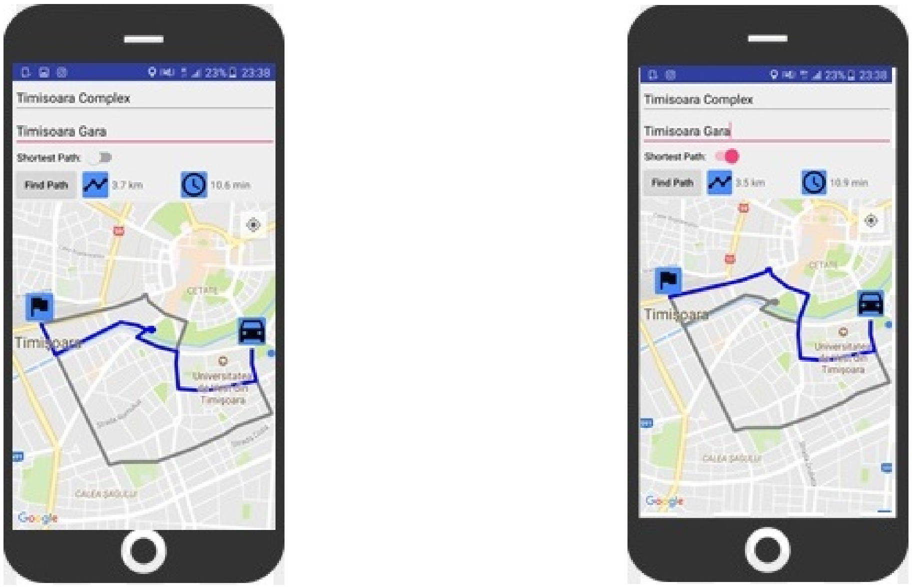
Tecuci

Telaviv

Teheran Teheran

Figure 1: Suggestions for origin location, input string ”Te”

The user has the option to ask for the shortest/fastest path. The default option is the fastest path and is returned by the Api based on the information reported to Google by users of Google Maps or info received from other sources. If the shortest path is desired then we parse all the possible paths and draw the one with the smallest distance with color blue to highlight it (alternative paths are drawn using grey).



Timisoara Complex

Timisoara Gara

Shortest Path:

Find Path 3.7 km 10.6 min

Timisoara Complex

Timisoara Gara

Shortest Path:

Find Path 3.5 km 10.9 min

(a) The fastest route highlighted with blue

(b) The shortest route highlighted with blue

Figure 2: Fastest vs shortest route comparison

3 State of the art

3.1 Google Maps

Google Maps app is a turn-by-turn navigation app that was the leader in this segment of apps for a long time. Nowadays is still the best overall navigation app because it has access to a large database of informations, always up to date and very well maintained. However, we believe that for the driving mode there is at least one better app. It moves towards becoming a social app now since you can share your location, get suggestions for restaurants/leave reviews for restaurants etc. . It can give you directions on public transportation, driving, walking or biking. Moreover, it can be used in offline mode too by downloading the offline information regarding a certain zone (e.g. your city).

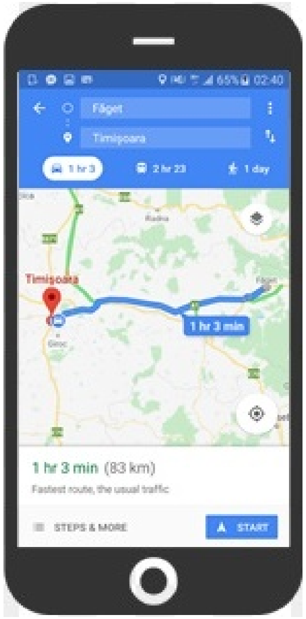


Figure 3: Google Maps app navigation screen

3.2 Waze

Waze is a turn-by-turn navigation app focused exclusively on driving, developed by a group of Israeli developers. It is very efficient when it comes to routing due to the fact that it uses the real-time feedback received from users. Because the users are encouraged to provide feedback, Waze can alert the user to avoid accidents, items on the road or police (!). Another cool feature is the speed alert, namely whenever the user goes over the speed limit it is notified by the app to slow down, and the GPS speedometer is way more accurate than the vehicle's speedometer . Moreover, safety of the driver is taken into account by implementing a series of measures (e.g. the user can't start the app while driving, the user can't search for directions while driving etc.). Having tons of features, a strong Waze community and being solely focused on driving, Waze is in our opinion the best driving companion app. As a result of its success, the app was bought by Google in 2013 for $1.15 billion.

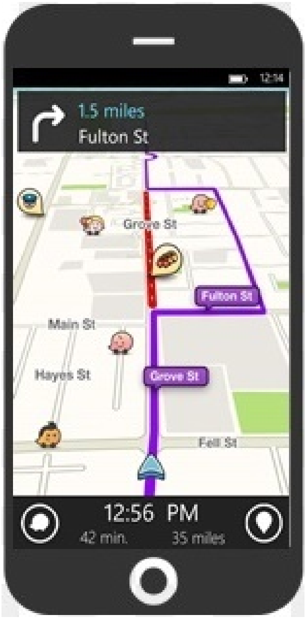


Figure 4: Waze app navigation screen

4 Usage

After the app is launched to the user there is displayed a splash screen for 2.5 seconds.

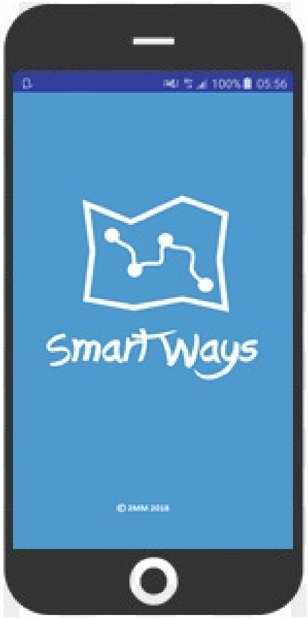
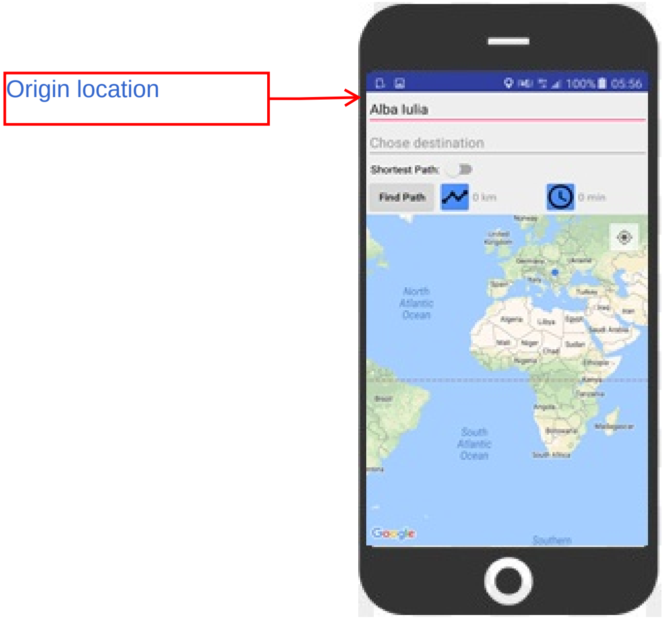


Figure 5: Splash screen

4.1 Insert origin

The user inserts the origin location as a string into the text field marked with red in the figure below. The inserted string is converted into a Lat&Lng pair by the Google Directions Api.



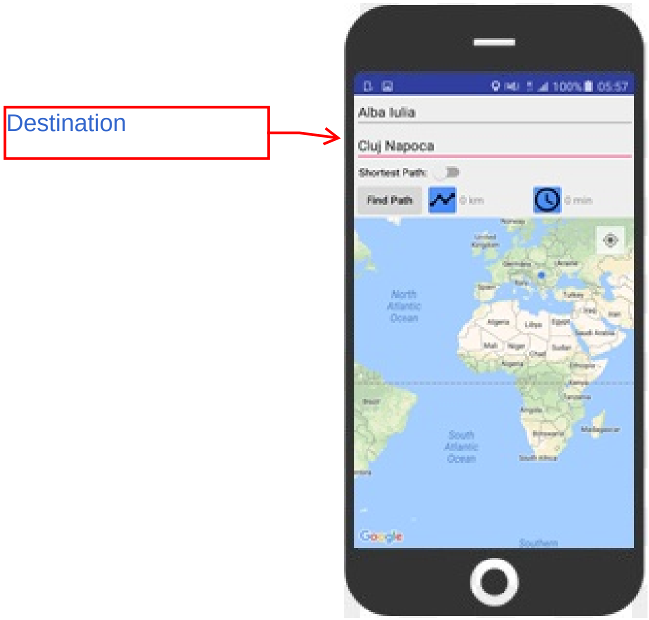
Alba Iulia

Chose destination

Figure 6: Origin location provided

4.2 Insert destination

The user inserts the destination location as a string into the text field marked with red in the figure below



Alba Iulia

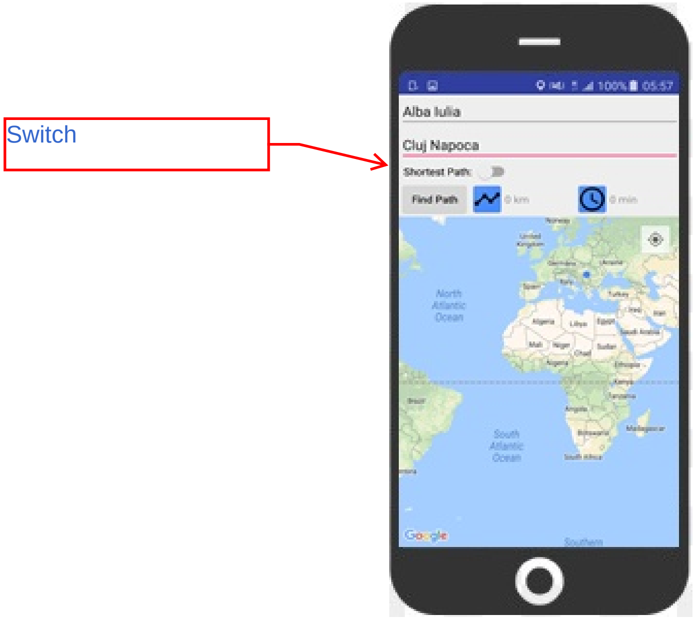
Cluj Napoca

Shortest Path:

Figure 7: Destination provided

4.3 Choose between shortest/fastest path

At this point the user selects the fastest path by leaving the ”Shortest path” switch off or the fastest by turning it on.



Alba Iulia

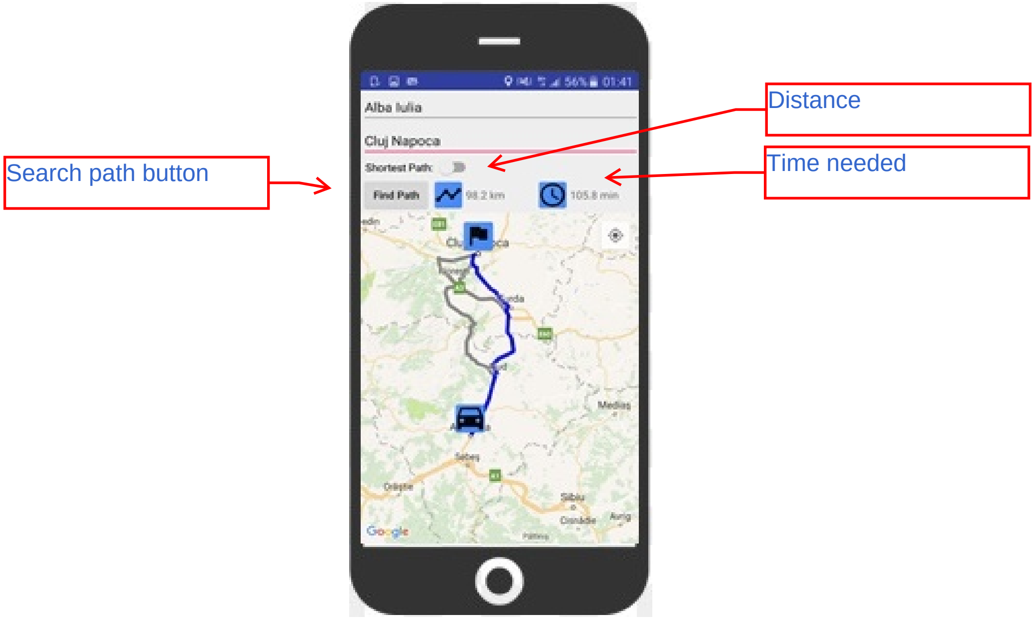
Cluj Napoca

Shortest Path:

Figure 8: Option : ”fastest” route

4.4 Hit ”Find Path” button

The user presses ”Find Path” button and the available paths are rendered on the map. The origin location is represented *car* icon and the destination with a *flag* icon. Also the distance and an estimation of time needed for the selected path are displayed.



Alba Iulia

Cluj Napoca

Shortest Path:

Find Path 98.2 km 105.8 min

Search path button

Distance

Time needed

Figure 9: The route result

5 Conclusions

While implementing this project we broadened our knowledge of mobile apps development and learned how to use some existing technologies. The tools used for developing the project were Android Studio as IDE, Github for Version Control and Skype in order to discuss topics related to app development. The main challenge that we faced was integrating the Firebase database module for the searched routes by a certain user, currently this part is not working and is on a separate branch in our Github repository.

The application can be extended to support combined ways of transportation, for example, if the user has to go from A to B no matter the ways of transportation, the app will generate a route to the nearest bus station or train station (on foot) with the fastest way of reaching the destination.

Github repository:

https://github.com/mariusolariu/Spp