



ThessOpenTrip Manual

Contents

Summary	2
Log in	3
Initial Screen	4
Navigation Bar	5
Map Pushpins	5
Map Layer Types	6
Points of Interest	7
Adding a new POI.....	11
Creating a Plan	12
Sending an incident report	15
Evaluating an incident report	16
Main Menu	16
Settings menu	18
General Notes.....	18
Report evaluation procedure	20
Application backend	21
REST API	22

Summary

ThessOpenTrip is a mobile application for tourists who visit Thessaloniki, aiming at facilitating their stay and sightseeing in various ways. The application deploys **crowd-sourcing** techniques for generating data by the end users and contributes to the **open data** community by allowing anonymous access to the user-generated data, provided through an open Application Programming Interface (API). ThessOpenTrip allows the crowd to create, evaluate and interact with points of interest (POIs) as well as to submit reports about various incidents, which take place in real time in the city of Thessaloniki and displayed on the city map (ThessOpenTrip incorporates OpenStreetMaps map system). In particular, the incidents that are supported are related to street events (e.g. festivals) and or current urban mobility conditions (e.g. live traffic). In this way, i.e. through the exploitation of crowd sourcing techniques, the application builds an open data repository of POIs and incidents that are of high interest to the city visitors, but also to data analysts, journalists, etc.

One of the key innovations of ThessOpenTrip relies on a mechanism for evaluating the reliability of the submitted reports, based on the reputation of the user, their location with respect to the event they report and their current speed. The real users also participate in this evaluation process in a crowd sourcing fashion, as follows. All users who are moving close to the reported incident are asked to validate the submitted report in real time. User feedback is used by a probabilistic model in order to estimate the degree of reliability of the submitted report. This mechanism contributes to the validation of the data that the crowd submits, providing a data filtering facility, as opposed to the most common applications of this kind, which allow any data, even spam information to be shared among the users of the same network. The ThessOpenTrip reliability estimation mechanism assures that no malicious information is injected into the system. E.g. only users with a high trust level (above 60%) can add new POIs.

On top of that, ThessOpenTrip enables users to create and perform cultural tours through the city's numerous monuments, landmarks, museums, restaurants or religious sites, based on the user's preferences with respect to available time, starting location, number of nearby places, and types of POIs, thus providing an integrated touristic information service that supports real time information.

This report provides a detailed guide to the ThessOpenTrip functionalities.

Log in

After successful installation, the application asks user to log in. This can be done either by using the credentials (email/password) created on the registration page or by user's Facebook account. Figure 1 shows screenshots of the login screens and the registration page. In order to use Facebook credentials, the user has to authorize the application to use the registered email. No personal information is used. After a successful log in, each time the application is used there is no need for authentication, unless the user chooses to log out.

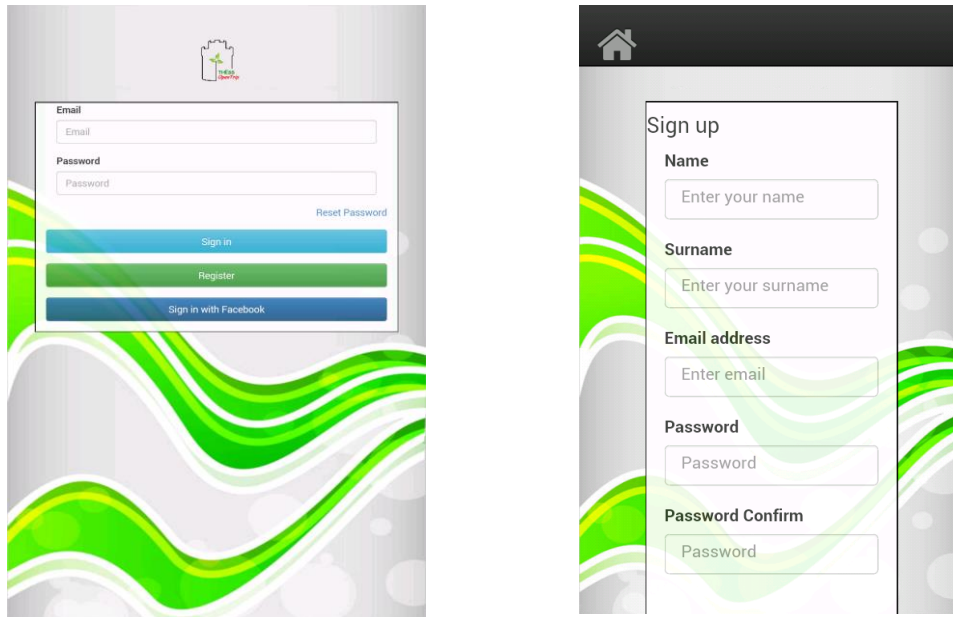


Figure 1. Login screen and registration page

Initial Screen

Once the application is successfully loaded and the user has signed in, the application shows the main screen. On the top of the screen there is the navigation bar (as a black ribbon) and at the bottom of the screen there are the action buttons, including the report button (light-red), the plan button (light-blue) and the POI button (light-green). Between the navigator bar and the action buttons, the OpenStreetMaps integrated map area is illustrated showing the user's current location, POIs, as well as any potential incident reports from other users. Clicking on the icon of each POI or report, an information pop-up icon appears providing some useful information, as well as relevant possible actions about the selected place. Figure 2 illustrates the main screen.

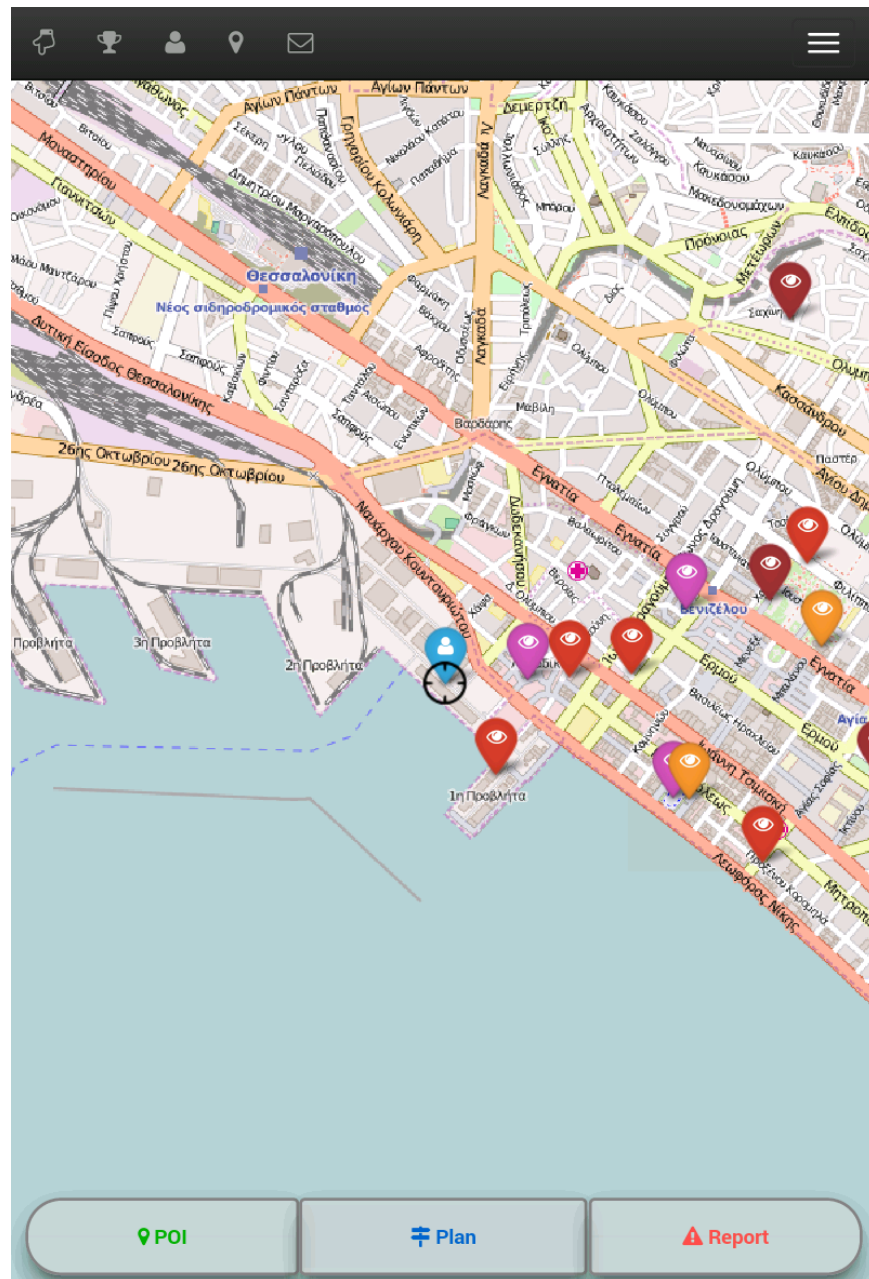


Figure 2. Initial Screen

Navigation Bar

The navigation bar (Figure 3) is always located on the top of the main screen. It has six buttons. The functionality of each button is described below.














-  centers the map on current user's location.
-  shows user's credits and statistical data on user activity.
-  displays a menu for filtering the POIs based on type and distance range.
-  /  shows a menu for changing the operational mode (walk/drive/cycle/public transport).
-  shows user's inbox in which reports from other users are pending for evaluation.
-  shows the main menu.






Figure 3. Navigation Bar

Map Pushpins

Different types of reports have their own representative pushpins on the map based on three parameters (Figure 4). Each pushpin has its own **icon type** - depending on the **type** of the incident, its own **background color** - depending on the **severity** of the incident and its own **transparency** - depending on the **reliability** of the report (as calculated and returned from the Report Evaluation System).

Type	Icon
Festival	
Street Event	
Scenery	
Traffic	
Weather	
Other	

Severity	Color
High	
Medium	
Low	




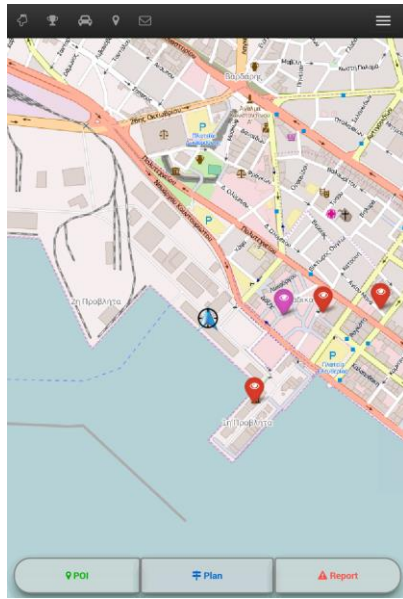
Reliability	Transparency
High	
Medium	
Low	

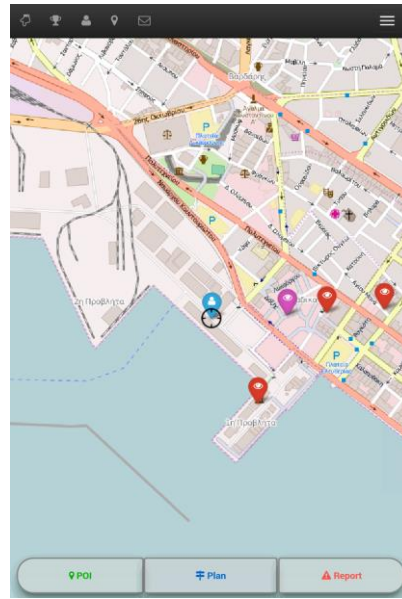
Figure 4. Map Pushpin Types

Map Layer Types

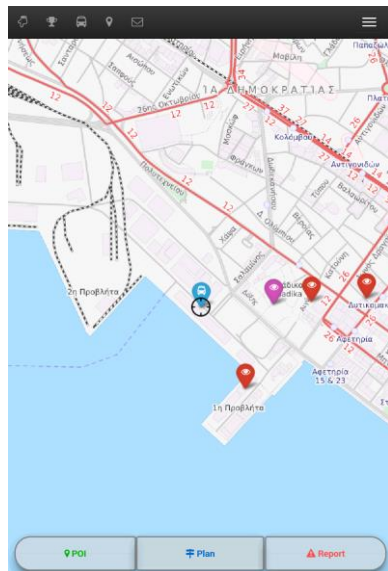
Depending on the selected user mode, there is a corresponding map layer where additional information is displayed on map. ThessOpenTrip integrates OpenStreetMaps, by implementing the corresponding open API. **Drive** and **Walk** modes share the same map layer whereas **Transportation** and **Cycle** ones have their own custom layers (Figure 5). Transition from one map layer to another occurs automatically, as soon as user changes mode.



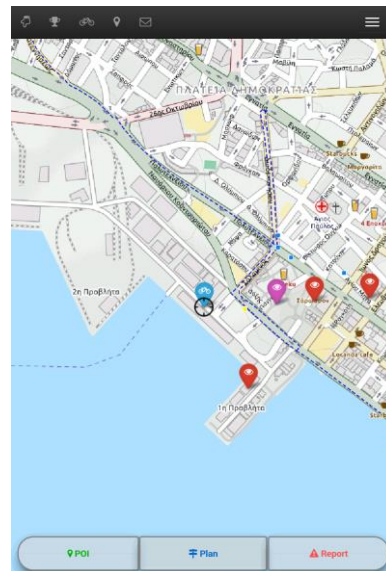
Drive Layer



Walk Layer



Transportation Layer



Cycle Layer

Figure 5. Map layer types

Points of Interest

Points of Interest (POIs) comprise places that a tourist can visit in the city of Thessaloniki. Each time the user opens the main map, POIs are downloaded from an **Open Data service** that ThessOpenTrip server provides (see section REST API) and then illustrated on the map as pushpin icons with different colors based on their type. There are five basic types of POIs:

- Museums (red color)
- Monuments (orange color)
- Landmarks (purple color)
- Religious Sites (dark red color)
- Restaurants (dark purple color)

All POIs can be filtered based on the desired type and the range from the user's current location. Figure 6 illustrates these options:

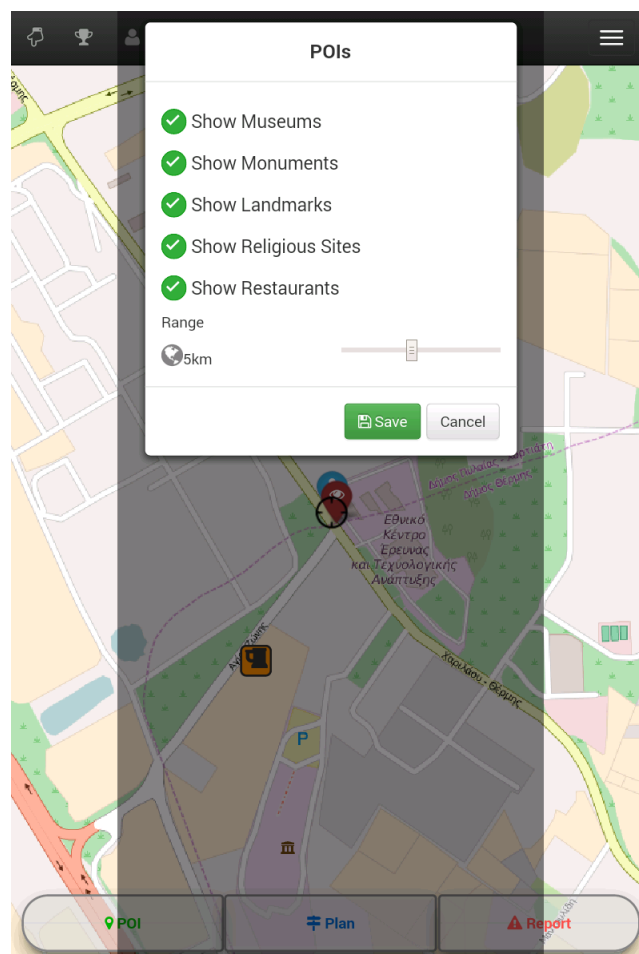


Figure 6. Filtering POIs preferences

Clicking on the icon of each POI, an information pop-up icon appears providing some useful information and some possible actions about the place (Figure 7).

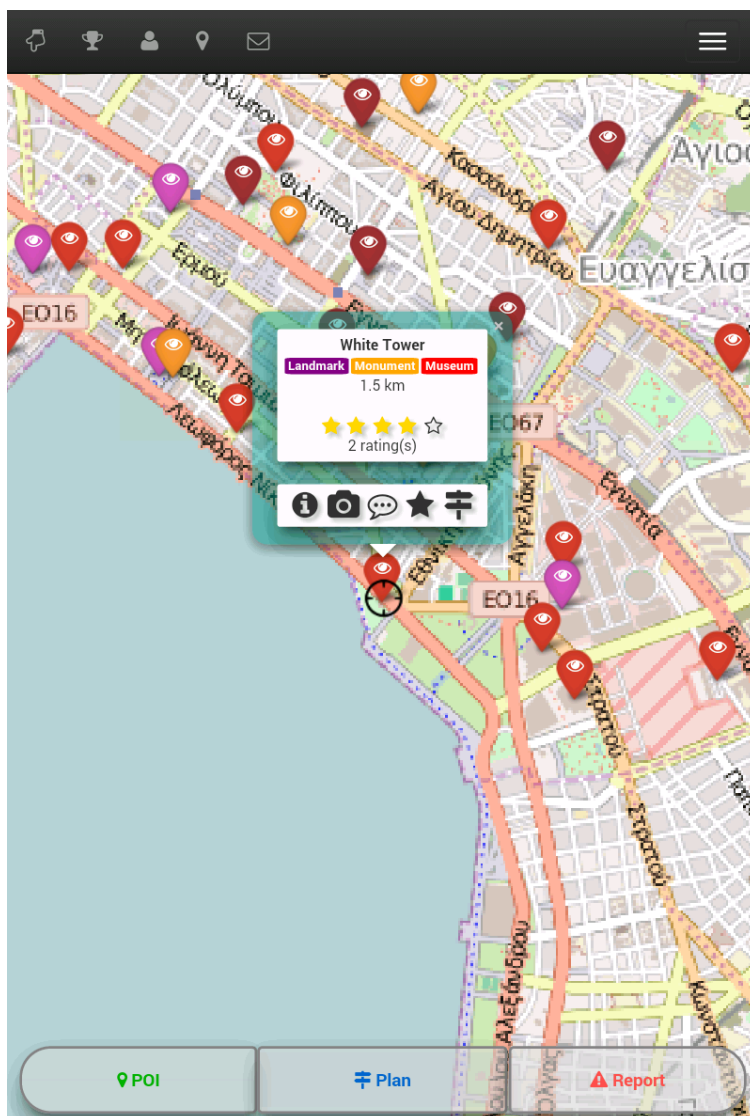


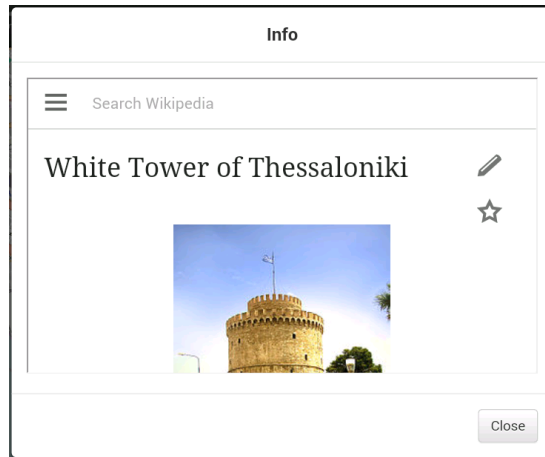
Figure 7. POI details and actions

Details section about the selected POI includes:

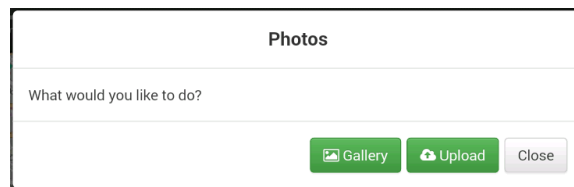
- Name of the POI
- Tags of the POI (e.g. Museum, Monument etc.)
- Distance of the user's current location from the POI
- Rating of the POI (star system from 1 to 5)
- Total ratings of the POI

Actions section about the selected POI includes:

- An “Info” button, which opens a web page providing detailed information about the place.



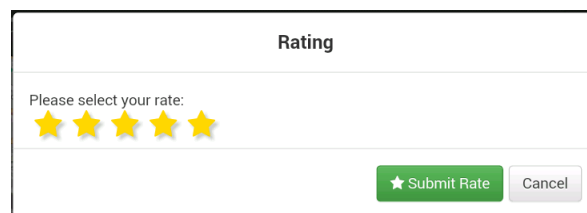
- A “Camera” button, where the user can either see a gallery of photos uploaded by other users or upload their own photo.



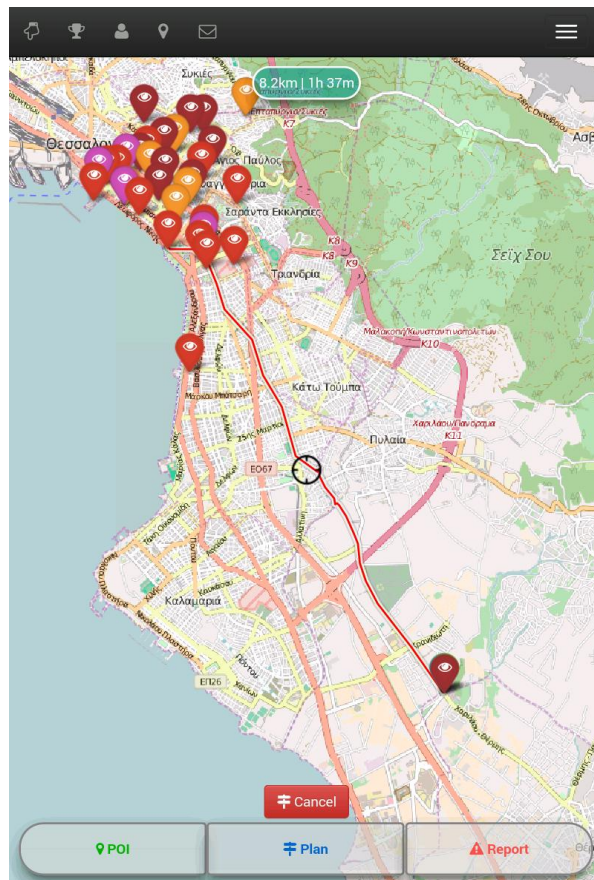
- A “Comments” button, where the user can either see comments posted by other users or post their own comments.



- A “Rating” button, where the user can rate the selected POI.



- A “Get Route” button, where the user can get a route to the selected POI on the map (Routing algorithm takes into account user’s mode e.g. Walk, Drive etc.)



Adding a new POI

ThessOpenTrip, promotes the concept and benefits of **crowd-sourcing** and **Open Data**, by providing users with the ability to create new POIs on the map (see Figure 8) just by clicking on the light-green button at the bottom-left section of the screen. However, to do so their trust level must be equal or above **60%**. Users can see their own trust level by clicking on the trophy icon located at the top panel of the application. In order to increase their trust level value, users must send reports or evaluate reports submitted by other users. If the evaluation algorithm classifies a report as true, then the user who submitted the report receives an increase of their trust level. This procedure is used in order to prevent new users from creating fake POIs or inject malicious information into the system.

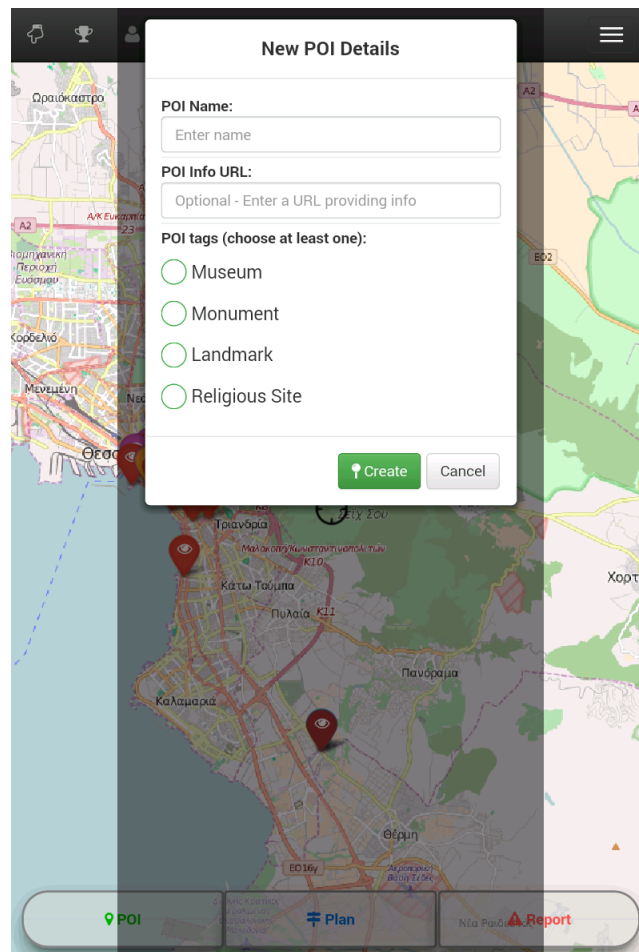


Figure 8. Creating a new POI

Creating a Plan

Users have the ability to create a new Plan (see Figure 9) by clicking on the light-blue button at the bottom-middle section of the screen. A plan is a routing guide, which proposes to the user a list of POIs to visit (Figure 10), taking into account a set of user's constraints and then creates a route through the selected POIs on the map (Figure 11).

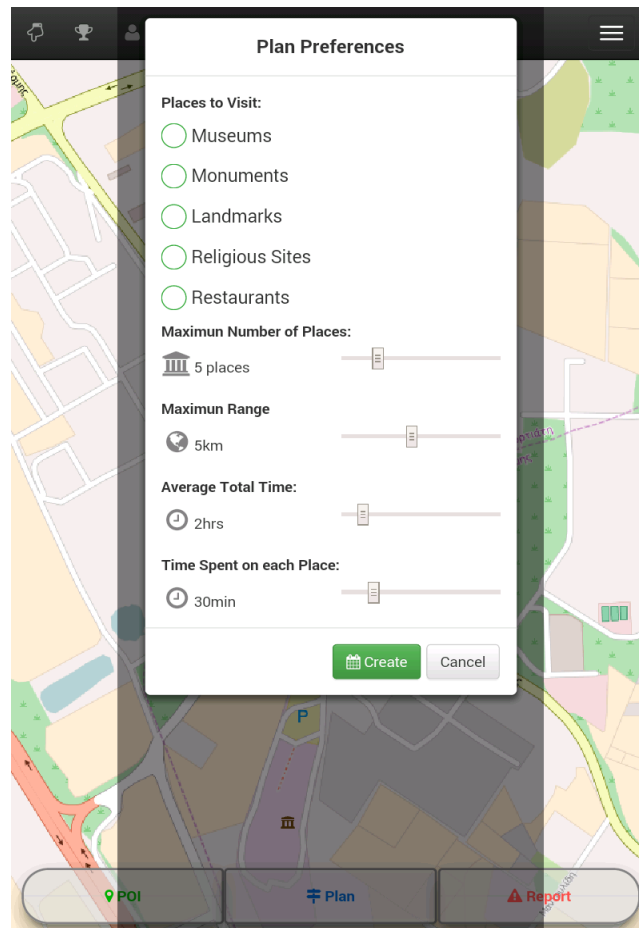


Figure 9. Creating a new Plan

Plan constraints include:

- **Places to visit:** The desired type of places that the user wants to visit (can select more than one).
- **Maximum Number of Places:** The maximum number of POIs that the user wants to visit.
- **Maximum Range:** The maximum area of searchable POIs based on the user's current location.
- **Average Total Time:** Total duration of the proposed plan including transition time between each POI and time spent on it. (Regardless of the user's input, the maximum number of places can be adjusted to fit the total time).
- **Time Spent on each Place:** The time the user plans to spend on each place (affects total time).

Since the user clicks on the “Create” button, a new window appears with a list of the places selected by the plan algorithm (Figure 10). This window includes also the **total distance** of the plan, the **walking/driving/cycling time** of the plan and the **total time** of the plan as calculated by the planning and routing algorithms.

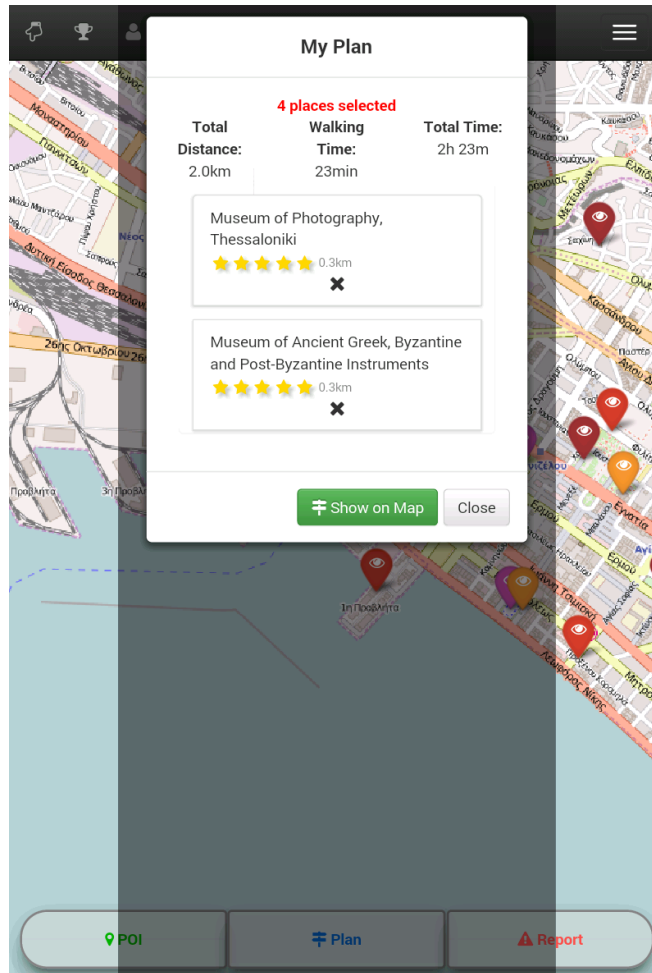


Figure 10. Plan algorithm results

Finally, the user can see the route calculated by the routing algorithm by clicking on the “Show on Map” button. Then, a new window appears (Figure 11) showing the plan on the map and providing instructions to the next POI in the plan. The user can cancel the plan any time by clicking on the “Cancel” button located at the bottom of the screen.

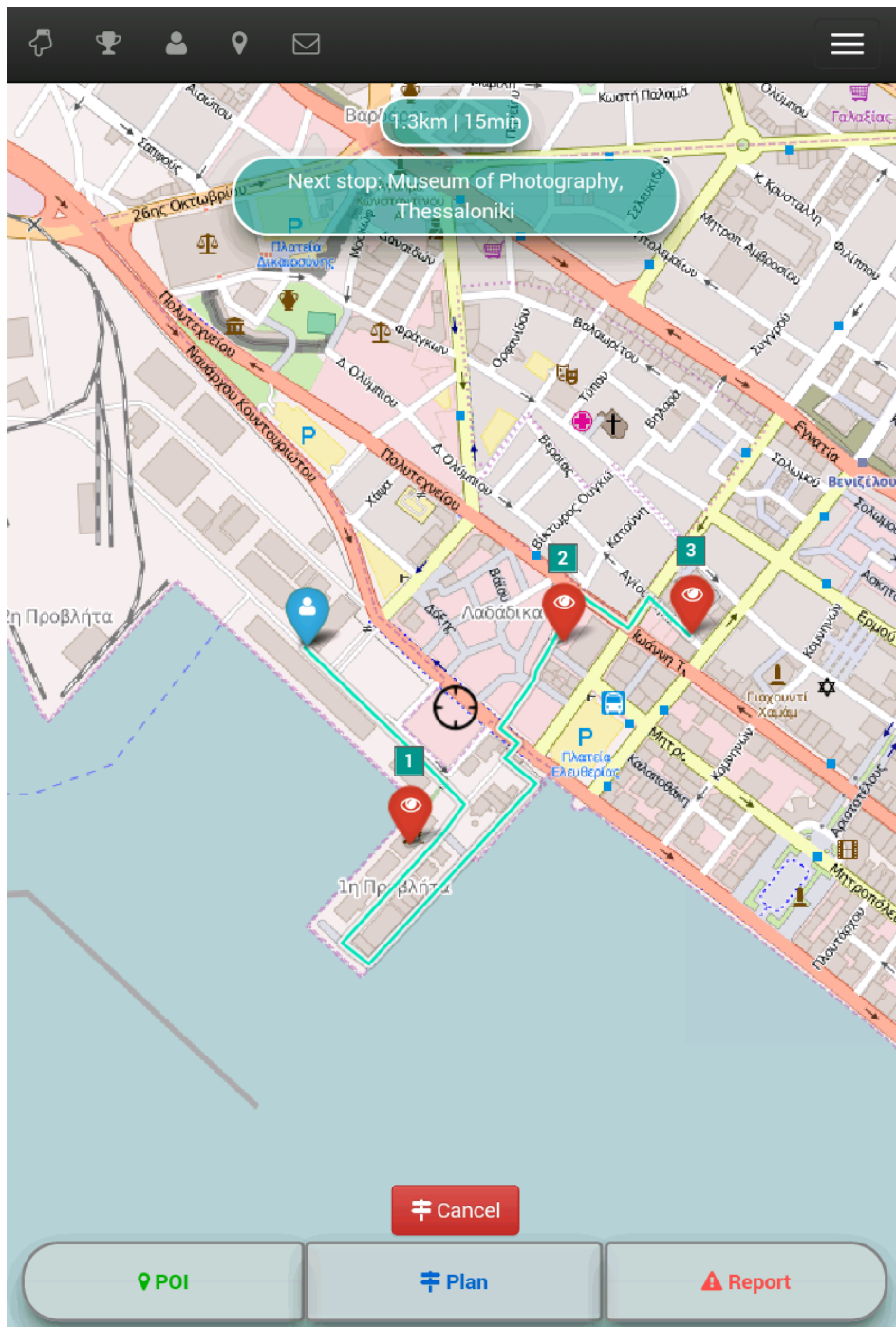


Figure 10. Routing algorithm results

Sending an incident report

When a user wants to send an incident report, s/he scrolls the map in order to move the ‘target’ icon which is by default located at the center of the map. The ‘target’ icon denotes the position of the incident to be reported. By pressing the report button, the window that appears (Figure 11) asks for the user to submit some details. Optionally, the user can add some text to the comments field describing the event/incident (maximum length of message: 160 characters).

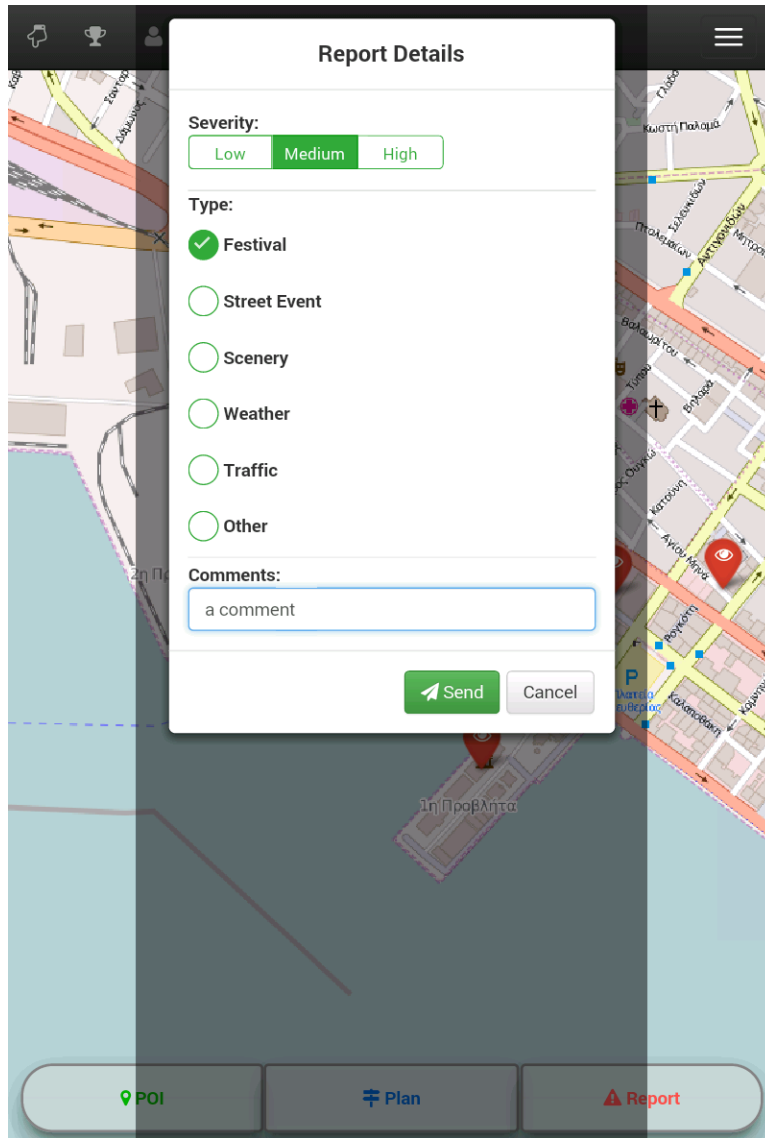


Figure 11. Submitting report details

Evaluating an incident report

Apart from sending incident reports, users can also evaluate the reports that other users have sent, contributing in that way in the **crowd-sourcing** mechanism running behind the scenes of ThessOpenTrip application. For this purpose, each new incident report appears as a new message in the inbox pending user's evaluation (Figure 12). By evaluating a report the user helps on increasing the reliability and hence the quality of the displayed information.

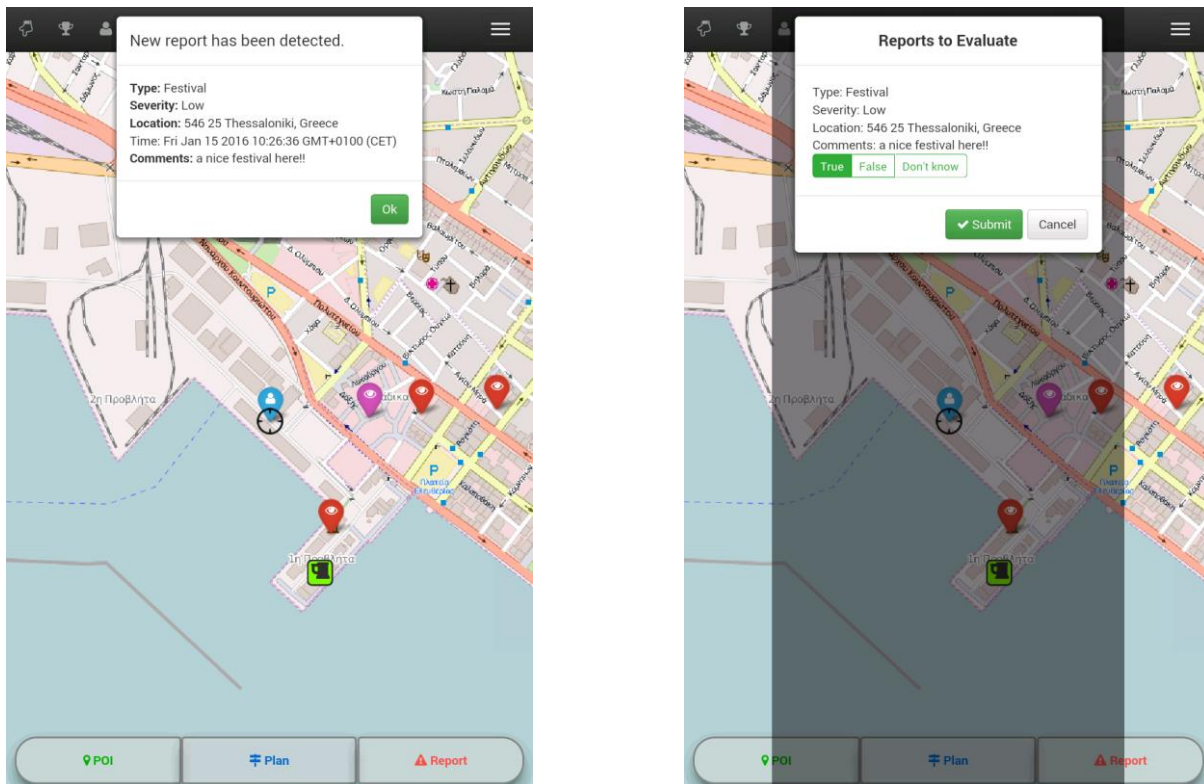


Figure 12. Evaluating Report

Main Menu

The main menu loads through the menu button at the right of the navigation bar (Figure 13). The following options are supported:

- 'Settings' displays the particular settings form (see the following section).
- 'User Manual' displays brief instructions on the main functionalities of the application.
- 'About' displays some general information.
- 'Logout' signs out the user.

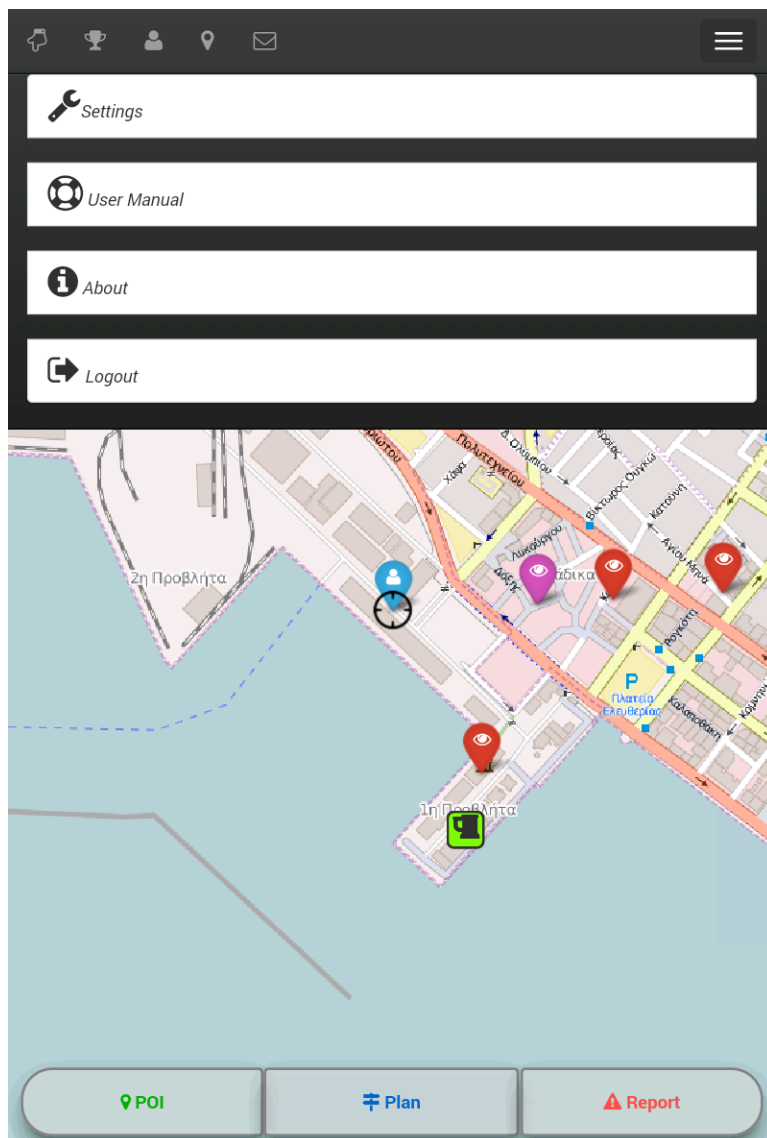


Figure 13. Main menu

Settings menu

The Settings menu consists of a form as illustrated in Figure 14. Through this menu, the following options are available:

- 'Operational mode', drive/walk/transportation/cycle (it can also change from the corresponding button on the navigation bar).
- 'Background color', which is the color of the navigation bar (default for Android is black, default for iOS is white).
- 'Near distance of reports on map (km)', which is the maximum distance from user's position in which report icons are displayed on the map.
- 'Near distance of notifications (km)', which is the maximum distance from user's position in which notifications and messages for incoming reports are showing up on screen for evaluation.
- 'Reports time (minutes)', which is the maximum age of displayed reports (max 24h).
- 'Show notifications' checkbox. When enabled, notifications for new reports are displayed.
- 'Show my reports' checkbox. When enabled, user's own reports are also shown.
- 'Notification sound' checkbox. When enabled, a sound will beep every time the user receives a new notification.

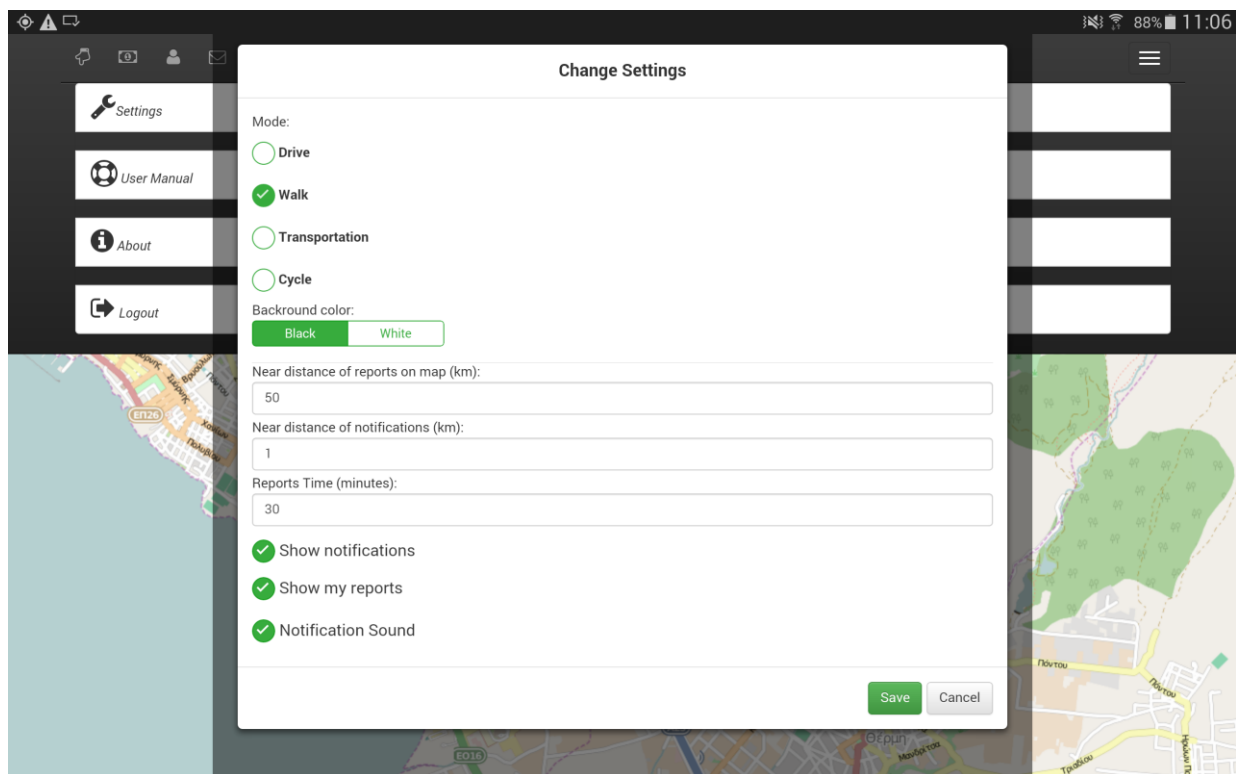



Figure 14. Settings menu


General Notes

- There are four operational modes: drive/walk/transportation/cycle. Use them depending on how


the application is used.

- User's position is represented on the map by four different icons, depending on the operational mode they are using (drive/walk/transportation/cycle). These icons are shown below:

-  *Heading Arrow*: Used in driving mode. Supports rotation based on user's heading.

-  *Walk Pin*: Used in walking mode.

-  *Transportation Pin*: Used in transportation mode.

-  *Cycle Pin*: Used in cycle mode.

- Each time the map changes its position by scrolling and the target icon is moved, the map is not relocated on user's location automatically. For this purpose, the first button on the navigation bar from the left should be pressed in order to relocate the map.
- In driving/transportation mode, there are occasions when user's heading cannot be calculated accurately (e.g. when speed is very low or user is not moving at all). In these cases, heading is measured/ estimated based on previous values.
- The application retrieves the address of each report based on the corresponding coordinates. Nevertheless, sometimes address retrieval takes more time than usual depending on network conditions. Even if the application does not retrieve the address, the server will do so, given that the report is located on proper terrain.
- It is recommended to enable the GPS on the device in order to improve accuracy and avoid delays.
- Each new report inserted into the system appears to all nearby users who are asked to evaluate the incoming report. In this respect, users in drive mode are being prompted to evaluate incoming reports only when they are approaching report's position.
- Application's efficiency depends on network speed and reliability. Therefore, the usage of 3/4G cellular network data access is recommended. Nevertheless, the application can also work on GPRS (2G) data access, however some application parts (e.g. map loading) may be slower.
- In order to increase the quality of the displayed information, the evaluation of other user's reports is important, especially for a short time period after the report is generated.
- The server does not accept reports located on abnormal terrain e.g. over sea or lakes.
- It is recommended to always update your installed application to the latest version.

Report evaluation procedure

1. Each incoming report is firstly evaluated by the report reliability assessment system using the distance between the location of the reporter and the location of the report and the speed of the reporter. Then, the report is shown on the map, with transparency based on the system's first evaluation score, and is available for further evaluation by the users.
2. After 5 minutes the traffic report reliability assessment system reevaluates the incoming report, taking into account this time the users' evaluations.
3. The reports that passed the second evaluation check in step 2, will be visible to nearby users for a specific time period, which can be parameterized by the user in the Settings menu (->Reports Time).

Important notes:

- Reports cannot be evaluated outside the notification distance.
- The reports are available for evaluation for 5 minutes after their first submission.
- The reports that Inbox contains are only these which were submitted in less than 5 minutes and have not been evaluated from the user yet. When a user evaluates a report, or after 5 minutes from the first submission of the report, the latter is removed from inbox as there is no need for further evaluation.
- In drive/transportation mode, report evaluations are only available for approaching reports based on user's heading.
- A user cannot evaluate its own reports.

Application backend

The application backend consists of a database for storing and retrieving reports and POIs and a REST API for accessing the database. The overall system architecture is shown in Figure 15.

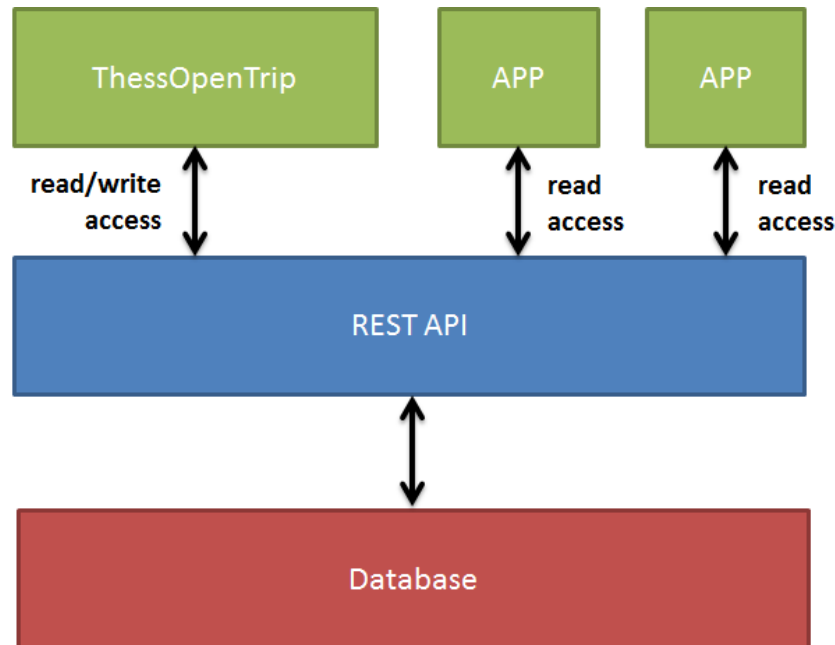


Figure 15. System architecture

We were not able to find an *open* database of POIs for Thessaloniki so we decided to **create a POI database and provided open access** to it. We also provide **open access to the incident reports database**.

After considering a number of alternatives, we selected MongoDB as the system database. MongoDB is an open-source document store with a flexible data model offering amongst others support for handling geospatial data, an essential feature for applications similar to ThessOpenTrip.

We also chose to provide access to the database using a REST API, making it easier for new applications to use the data stored in our database. Note that even though access to data is open, adding or modifying data requires authentication. This occurs for preventing misuse or spam, while enabling our internal algorithms to compute and evaluate the reliability of each report. Therefore, the creation or editing of POIs and reports can only be done using the ThessOpenTrip application, whereas any other application can freely access the ThessOpenTrip database and retrieve information about POIs and incident reports near a selected location.

REST API

In this section we briefly describe the API that external applications can use to access our open database.

Reports

Access to the reports database can be done using the following API:

```
http://109.231.126.241/reportdb/public/report/getByLocTime?latitude=40&longitude=20&max_distance=2&timestamp=12345678&past_window=30
```

which returns reports near the given location, within a distance of 2 km that were created within the given time window. Results are returned in the following JSON format:

```
{
  "reportLatitude" : 40.6375,
  "reportLongitude" : 22.945,
  "address" : "Filippou, 546 31 Thessaloniki, Greece",
  "type" : "traffic",
  "severity" : "medium",
  "timestamp" : "1452542651",
  "comments" : "Watch out!",
  "reliability" : 1
}
```

POIs

Our POI database currently includes museums, monuments, landmarks, religious sites and restaurants. An application can retrieve POIs near a location using the following URL format:

```
http://109.231.126.241/poidb/public/poi/getByLoc?latitude=40&longitude=22&max_distance=2
```

which returns entries of the form:

```
{
  "oid" : "5699202c2bce004f4fca6220",
  "longitude" : "22.962",
  "latitude" : "40.644",
  "name" : "Heptapyrgion",
  "tags" : [ "Landmark" ],
  "ratings": [ "ganton" : 4, "abalask" : 4.5 ],
  "userId" : "ganton",
  "url":"https://en.wikipedia.org/wiki/Heptapyrgion_(Thessaloniki)"
}
```

Applications can also retrieve photos and comments submitted by users. Given a POI id as returned from a call to /poi/getByLoc, an application can retrieve comments for the specific POI using a call similar to the following:

```
http://109.231.126.241/poidb/public/comment?oid=0123456789
```

which returns a list of comments using the following format:

```
{
  "userId" : "abalask"
  "text" : "Really nice museum! A must visit."
  "time" : "1452542651"
}
```

Finally, an application can also retrieve user photos submitted for a specific POI using the following URL:

<http://109.231.126.241/poedb/public/photo?oid=0123456789>

which returns photos in a format similar to the following:

```
{
  "userId" : "abalask"
  "src" : "<photo encoded as a base64 string>"
}
```