Bilkent University
Department of Computer Engineering

Senior Design Project

Analysis Report

Urbscope

urbscope.com

Group Members

Burak Mandıra
Mustafa Motani
Waqaas Rahmani
Syed Sarjeel Yusuf
Abdullah Wali

Supervisor

Mustafa Özdal

Innovation Expert

Veysi İşler

Jury Members

Selim Aksoy
Mehmet Koyutürk

Table of Contents

1. Introduction

As of 2018, people are working on different industries from transportation, accommodation toy engineering, medicine and science. Nonetheless, everyone at some point of their life is a tourist when they go for holiday regardless of whether it is to abroad or domestic. As the technology has been developing, the tourism industry has evolved as well as others. The fast and radical changes due to recent developments has changed people's lives and this, of course, affects how people travel and learn about different holiday locations. Therefore, we propose to develop a mobile application that will serve the people's needs when on holiday with the solutions that use recent technologies. Recent developments in technologies regarding the way human perception, such as augmented reality (AR) and virtual reality (VR), have become very popular and are seen as the future of our lives. They can provide more realistic, descriptive and yet simple solutions to the problems that we have had so far.

When people go to holiday, most of them do not know what kind of places are available to visit or where should they visit for a specific city. They search for museums, galleries, architectural landmarks, historic monuments etc. on the Internet and try to find information as much as they can do before their trip. However, this is not efficient and all the information that has been searched is prone to be forgotten when the trip starts. Therefore, some people prefer doing this information retrieval phase throughout the trip. Nevertheless, within a short time, this can be quite cumbersome because checking everything that surrounds you on the Internet takes a lot of time and finding the decent information can become quite time consuming. Also, one might not want to spend his/her time searching and filtering things on the Internet as most of the time, it is very important to ask the right questions to search engines when one wants to elicit the information- while he/she is visiting some venues that he/she should make the most them. This becomes quite challenging when tourists even do not know the name of the buildings. What is worse can occur when they are unaware of the popular tourist attractions and they skip those attractions as they are just passing by. Therefore, we decided to design a mobile application, which is called Urbscope, that will make acquiring information about the landmarks that people visit fun and easy through AR technology. The system aim to make people more aware of the nearby popular tourist attractions. Tourists will no longer need to search for any buildings on the Internet and rely on and they can have fun of their trips rather than wasting their time on their smart devices searching in order to elicit information thanks to Urbscope.

In this report, we will thoroughly describe the functionality of Urbscope, compare it with the existing products to clarify its value and place among current systems, and provide a preliminary idea of how our product will look like. We will make use of diagrams and UI mock-ups where necessary throughout our report.

2. Current system

The system can be broadly defined as locating landmarks using augmented reality. Considering that augmented reality is relatively a new technology on a commercial scale, especially with the release of Apple's ARToolkit [1], there are not several similar systems. Therefore, while discussing the currently existing products, we will consider those that employ augmented reality and those that do not, and attempt to explain the context of our product in relation to each of those products. This will

be done keeping in mind the basic objectives of the system to be developed. These objectives, which are further elaborated in Sections 3.1 and Section 3.2, can be summarized as below:

- Recommend points of interest to tourist and travellers alike.
- Detect Landmarks.
- Provide information pertinent on points of interest, preferably touristic landmarks.

At the moment there is no single product that provides all these three major features, but some of the features are provided separately by various applications available commercially.

2.1. Information on Points of Interests

The most popular and widely used product in the tourism industry for finding points of interest is TripAdvisor [3]. Googling "Places to visit in" followed by any country/city name, from London to Zimbabwe will always return a link to a Trip Advisor page in one of the first 3 results (as seen in *Fig 2.1)*. TripAdvisor describes itself as a company that provides "hotel and restaurant reviews, accommodation bookings and other travel-related content" **[5]**. In the context of our product, we are only interested in the travel-related content from TripAdvisor. A quick look at their "Things to Do" feature (which is analogous to our feature of recommending places to visit) shows that they rely on on user-generated content and reviews to recommend the places or sites with the highest ratings and interactions. In contrast, our product mainly aims to identify and direct attention to touristic landmarks in the proximity of the user and recommend similar or related landmarks.

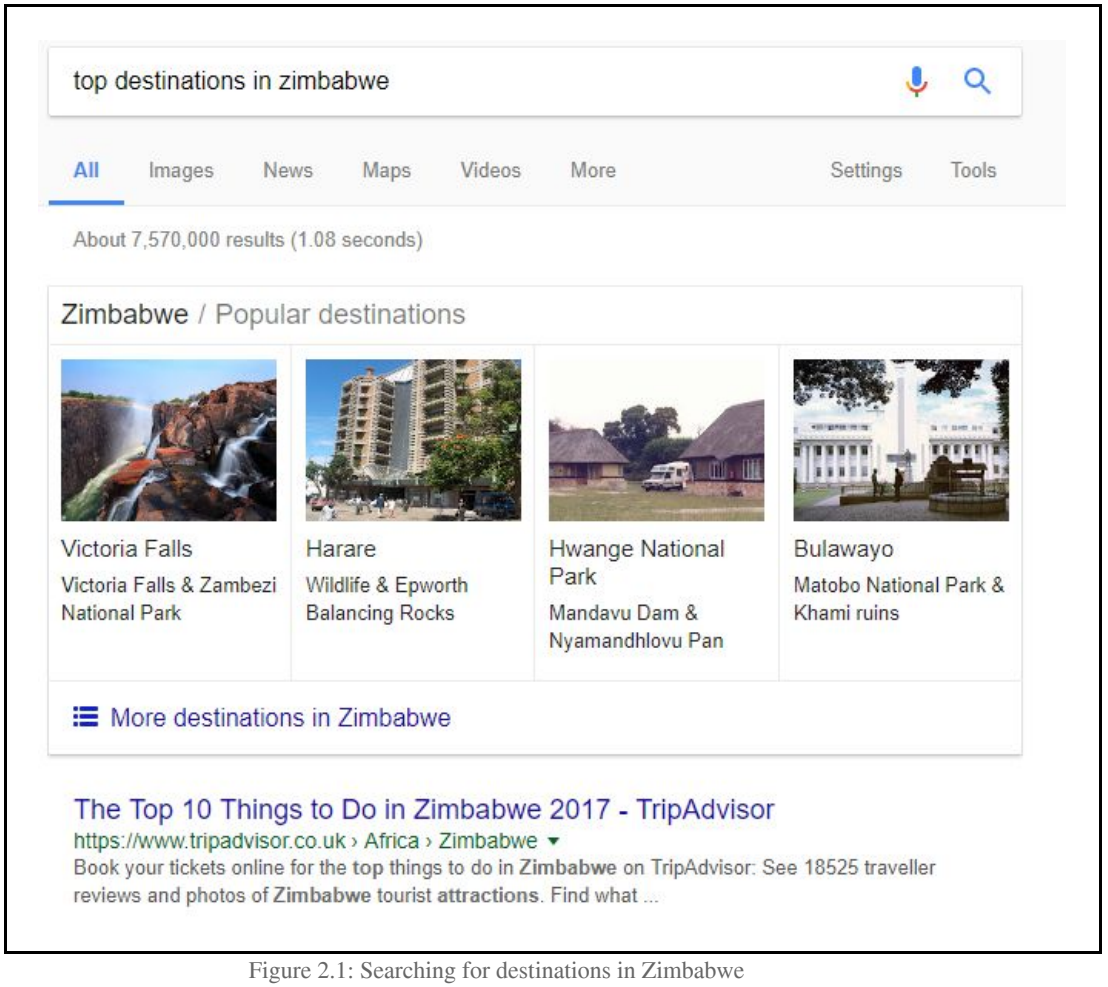

Figure 2.1: Searching for destinations in Zimbabwe

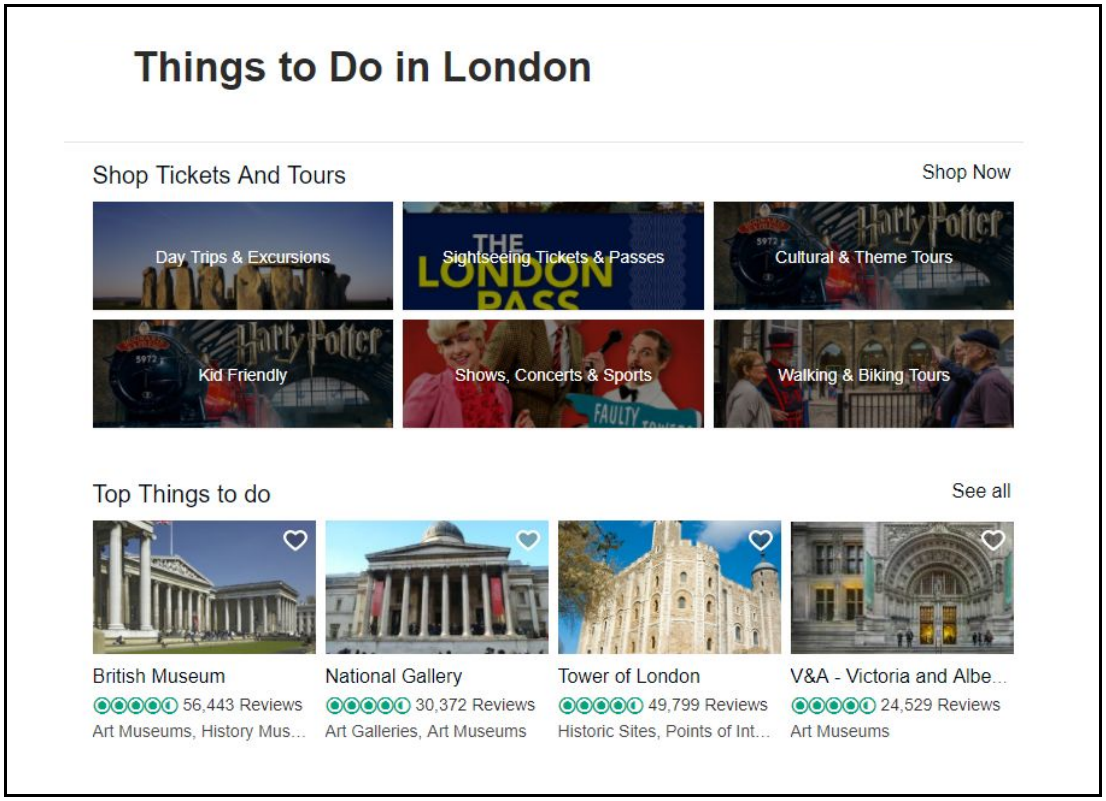

Figure 2.2: Tripadvisor's Things to do feature

2.2. Detecting Landmarks

In terms of augmented reality and landmark detection, the most similar product to the one proposed is called 'Islamic GPS' targeted at the Muslim demographics [2]. The product is very similar to the proposed system in terms of user interface and user functionality. The mobile app allows the user to point his camera to a point of interest, and if that targeted point has any significant relevance to the Islamic faith, the application detects it. It however does not provide information pertaining to the point of interest, unless it is the Mecca. Other than that, the application is also aimed at detecting mosques, but does not provide any historical or cultural information about the mosque.

Nevertheless, it locates the landmarks within a certain radius and puts augmented reality markers onto them. These markers then indicate the location of the points of interest, and the user can click on one from several of the markers that show up. Upon clicking on the marker, the user is redirected to a page with basic information about the point of interest, and has the option to get directions to the selected target. This application is used extensively by Muslim users to locate mosques. Hence as there are many mosques present in the horizon, there shall be many augmented markers on the screen of the phone as the user pans the horizon with his camera. This resembles the objectives of the proposed system, except that instead of only Islamic buildings the system will mark all points of interest with high touristic value.

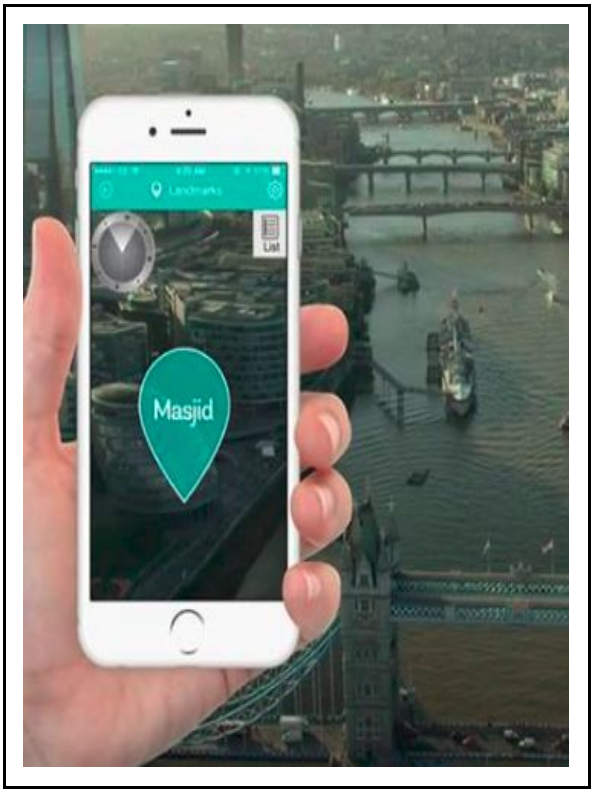

Figure 2.3: Mosque Detection

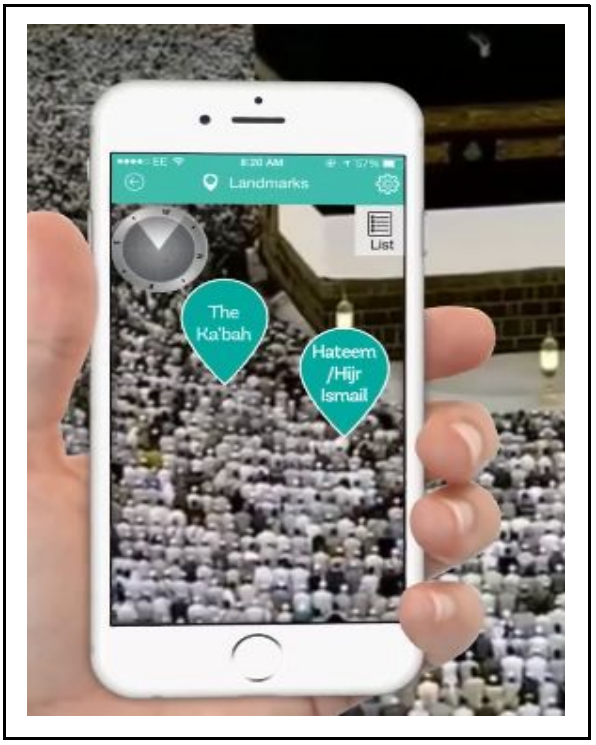

Figure 2.4: Several points of interest

As can be seen from *Fig 2.3* and *Fig 2.4* the current system imitates the required goals of the system. By scanning the horizon, more than one marker pops up and the user can select a single marker to set as the target location. This brings out the directions as seen in *Fig2.5*.

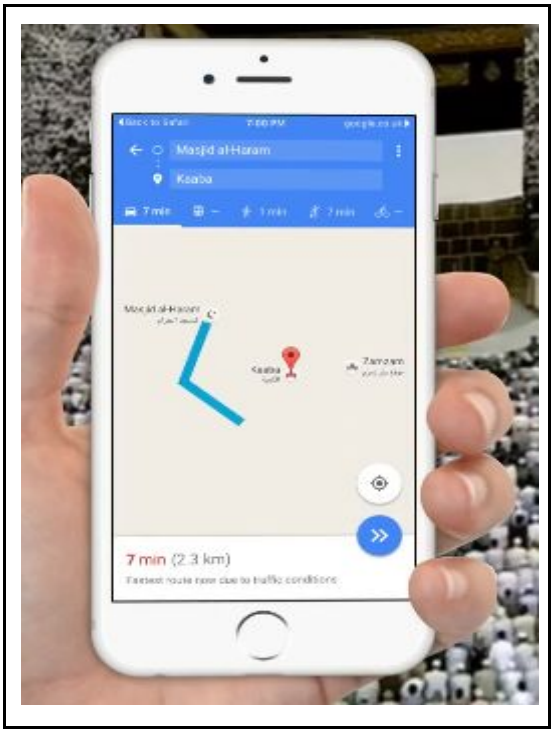

Figure 2.5: Directions

2.3. Recommend Points of Interest

As seen from Section 2.2, there does exist an application that closely resembles the proposed idea, but operates for a different consumer base. However, irrespective of the functionality of the current systems, none of them effectively manage to integrate a recommendation system to target the users personally. There are however, stand alone recommender systems available for the commercial market. It shall thus be these recommender systems that will be the standard of comparison when developing the proposed system's recommender system.

A current system that is successful in recommending all types of culinary sites, is FourSquare. The company is known for its personalized recommendation system related to cuisine and it has proved to be one of the best of such systems. The recommendation engine developed by the company works extensively from the user's data including his or her location, previous choices and time based culinary choices. With all of this information the system manages to even decide what to display to the user when the application is opened on the phone. This can be seen in *Fig 2.6*.

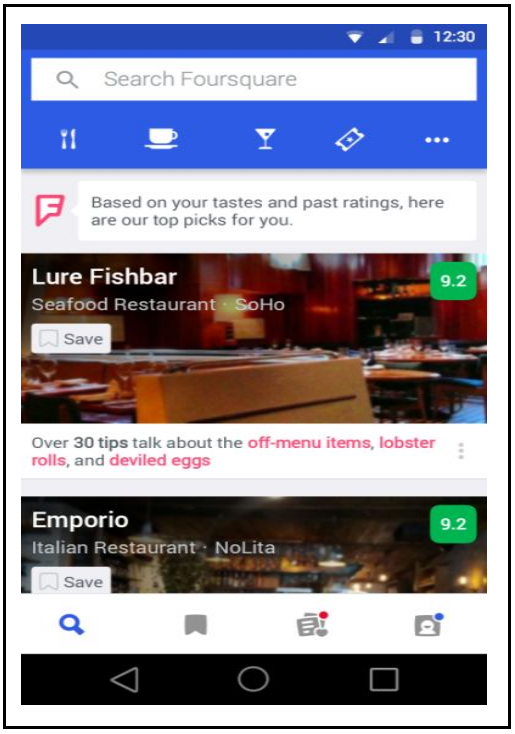

Figure 2.6: Directions

Nevertheless, Foursquare does not operate within the domain that the proposed system wishes to and that is for touristic sites. Infact, till now there are actually no large commercial products available that actually provide the proposed feature. This includes Tripadvisor which also does not provide personalised recommendations. However with they realise the potentials of such a feature, and know their advantage with the large amounts of data they currently have. Hence they are planning to introduce the recommender system soon [4].

3. Proposed system

3.1. Overview

Urbscope is a mobile application that puts the tourist industry into the palms of the user. The user interface of the application is simply the camera of the mobile phone. The user needs to point the camera at the landmark (skyscrapers, historical buildings, museums) and Urbscope, using augmented reality, will display all the information that user needs to know about the landmark. Furthermore, our app employs a recommendation system to suggest nearby tourist landmarks, cafes, restaurants etc. that user might need to know about during his or her travels.

3.2. Functional Requirement

In this section we will talk about the functional requirements of Urbscope. Functional requirements describe the functionalities of the system.

- The application's user interface will be superimposed on the camera view.
- The user will be able to identify landmarks by pointing his camera towards the landmark. This functionality is available in the "Recognition Mode".
- The application draws an augmented reality marker on the landmark at which the camera is pointed.
- Upon selection of that AR marker, the application will display an information box that shows basic and general information about the selected landmark and more detailed information is provided with a direction to outside of the app. This is called as information retrieval.
- The popular nearby attractions around the user will be shown by AR markers. This facility is provided when the user activates the Nearby switch under the Exploration Mode.
- Urbscope maintains a log file that writes down the unique identification of the visited places that is retrieved from the TripAdvisor API. The log will act as a history record of the user's touristic activities, which shall then be used in the recommendation system to provide personal recommendations to the user.
- The users can get recommendations through the recommendation system. This facility can be used if the user activates the Recommendation switch under the Exploration Mode.
- The users can put filters on recommended or nearby landmarks through the side navigation panel.
- The application will navigate the user towards the selected recommended building by making using of Google Maps API in case of the user's demand.
- This navigation facility is available inside the information box only if the user is not in front of the selected landmark.
- The recommendation system will be using Google Compute System.

3.3. Nonfunctional Requirements

- The application should be able to retrieve information of the landmark within feasible time span.
- The landmark detection will be done using Google Vision API.
- User interface should be smooth and lag-free.
- The app will indicate that the landmark is recognised when user points camera towards it.
- As default, only the attractions within the proximity limit will be shown, which can be adjusted through the side navigation panel.
- Depending on the proximity between the user and the surrounding attractions, the colours of the markers will be different, e.g. green for nearer attractions and grey for further ones.

3.4. Pseudo Requirements

- **Devices:** We expect that our application will be heavy on CPU usage and will require high end hardware to run smoothly due to the usage of Augmented Reality (AR) and possibly Image Recognition. Therefore we will be limiting the devices that we will be supporting to iPhone 6S and higher on the iOS spectrum to the latest flagship phones on Android (because wide range of phones running Android, it is not possible for us to specify which devices we will support; but, it is sufficient to say we're targeting the latest devices).
- **Front End:** We plan on using React Native for front-end development
- **Back End:** We will use Django (Python) or NodeJS (Javascript) for backend development
- **APIs:** We will utilise Google Vision API for the recognition of landmarks; TripAdvisor API for acquiring information about the detected landmarks; Google Maps API for navigation.

3.5. System Models

3.5.1. Scenarios

- Oliver tries to identify a cathedral in front of him

  Oliver is a tourist in Paris, and as he is walking through the streets, he sees a cathedral that he cannot identify. Oliver opens Urbscope to learn more about the cathedral. When the app is on, which is by default in Recognition Mode, and he points his mobile phone's camera towards the cathedral so that it is clearly in the view. The application will display an AR marker on the cathedral with "Notre-Dame de Paris" written on it.

- Oliver wants to learn more about the cathedral he has identified

  Oliver is interested in history and architecture, and wants to learn more about the Notre-Dame. He clicks on the marker on his mobile phone's screen, and an information box marker appears with some information about the cathedral. From the information he reads, Oliver learns that the Notre-Dame is of a height of 69 m, and that it was constructed in the year 1163 AD, following the French Gothic architectural style and so forth. At the end of these basic information, there is a link to an external source where he can acquire more detailed information.

- Oliver wants to learn about nearby landmarks

  Oliver is not appeased with just the cathedral and wants to learn about other touristic landmarks in the vicinity. He clicks on the button on bottom right corner labelled "Exploration Mode". Then, he activates the Nearby switch, which is on the top of the exploration mode screen, and sweeps the view with his camera. As he sweeps his camera slowly, all nearby sites of interests pop up, given that they satisfy the filters that can be adjusted through the side navigation panel. It also shows how far these landmarks are from the user by making the colours of the AR markers different depending on the distance The color shade of the marker will be aimed at giving an abstract idea of how far the landmark is. For example, the darker , e.g. red for non-nearby landmarks and green for nearby ones.

- Oliver wants to see the recommended landmarks for him

  Oliver didn't like the nearby landmarks so he decided to check recommended landmarks because he know that they are recommended based on his previous visits. Therefore, in the Exploration Mode, he switches to recommendation screen by activating the Recommendation switch. He can put some filters on the results such as categories of the landmarks and set the proximity limit according to his desire using the side navigation panel. Then, only the ones that satisfy these filters will be shown to him. He has to slide his mobile phone's camera towards different directions to see which landmarks are in which direction.

3.5.2. Use Case Model

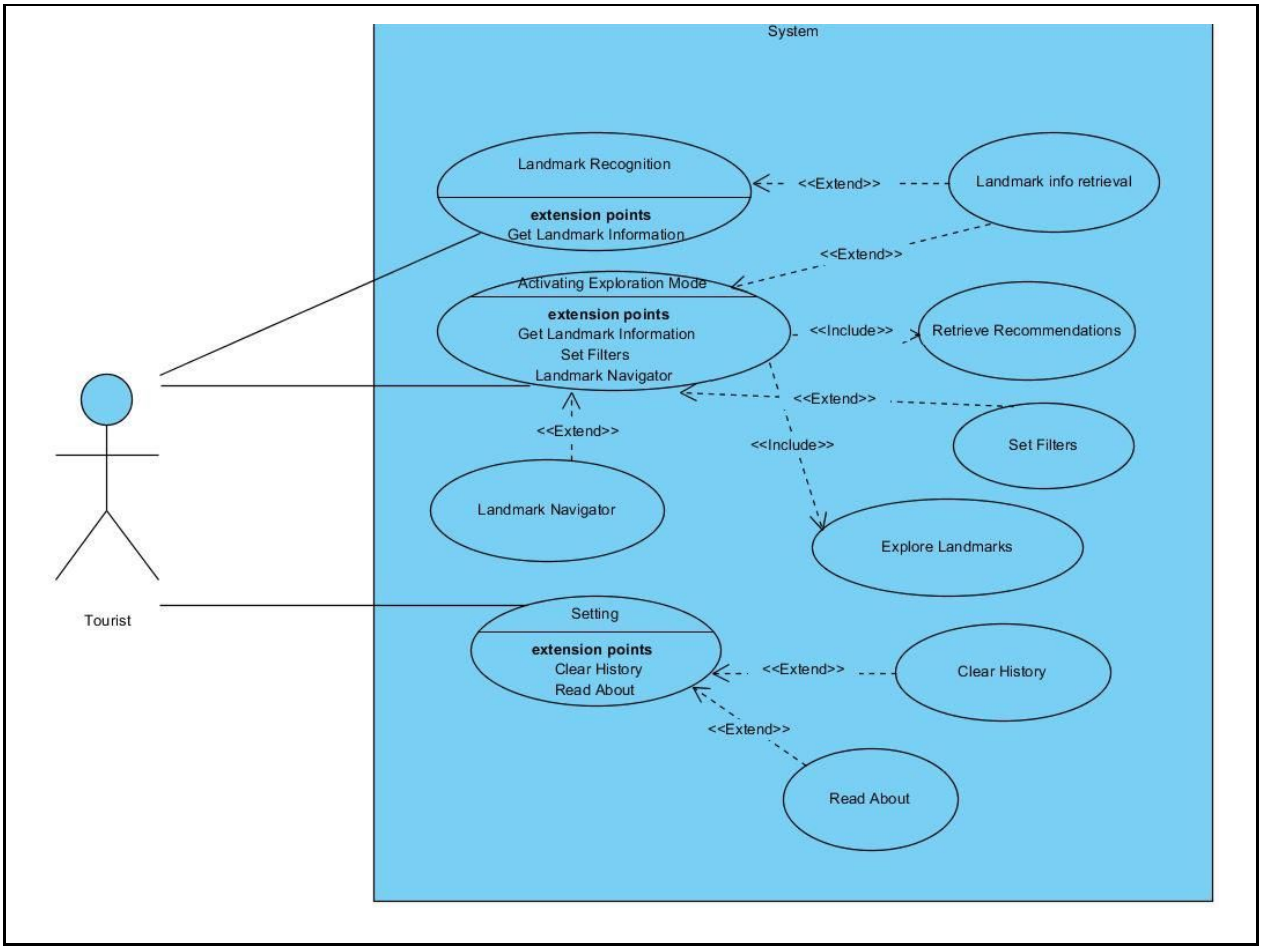

Fig 3.5.2.1

- Tourist uses the landmark recognition feature

Use Case Name: Landmark Recognition

Participating Actors: Tourist

Entry Condition:

- Tourist is connected to the Internet.
- Urbscope is running, and is currently in Recognition Mode.

Exit Condition:

- Tourist closes Urbscope.
- Tourist changes the current mode from Recognition to Exploration.

Main Flow of Events:

- Tourist points his phone towards the landmark.
- Tourist ensures that the landmark to be recognised is clear in the camera frame.
- Tourist holds the phone steadily for a few seconds for the detection to take place.
- Urbscope sends the current view to the server to for the landmark to be identified.
- Tourist sees a marker with the landmark's name overlaid over the landmark.

Alternative Flow of Events 1:

- Tourist loses the Internet connection.
- Urbscope fails to recognise the landmark.

Alternative Flow of Events 2:

- Tourist defectively points towards to another direction in which no popular touristic landmark is available.
- Urbscope tries to detect a landmark within the camera's frame.
- Urbscope cannot detect any popular landmark in the frame and nothing extra shows up in the screen.

- Tourist clicks on the info marker

Use Case Name: Landmark Information Retrieval

Participating Actors: Tourist

Entry Condition:

- Tourist is connected to the Internet.
- Urbscope is running, and a marker is currently displayed on the screen.
- Tourist has clicked the marker.

Exit Condition:

- Tourist closes Urbscope.
- Tourist clicks somewhere outside the marker.
- Tourist closes marker by clicking the close button.
- Tourist is redirected to external source for more information.

Main Flow of Events:

- A marker appears that shows information about the selected landmark.
- Tourist reads the information.

Alternative Flow of Events 1:

- A marker appears that shows information about the selected landmark.
- Tourist reads the information.
- Tourist wants to learn more about the landmark and clicks the redirection button at the end of the information provided.

Alternative Flow of Events 2:

- Tourist loses the Internet connection.
- Urbscope fails to fetch information about the landmark.

- Tourist activates Exploration Mode

Use Case Name: Activating Exploration Mode

Participating Actors: Tourist

Entry Condition:

- Tourist is connected to the Internet.
- Urbscope is running, and is in Recognition Mode.

Exit Condition:

- Tourist closes Urbscope.
- Tourist switches back to Recognition Mode from Exploration Mode.

Main Flow of Events:

- Tourist clicks on Exploration Mode button which is available in the bottom right corner.
- Urbscope switches to Exploration Mode with the Nearby switch activated by default.

Alternative Flow of Events:

- Tourist loses the Internet connection.

- Tourist wants to learn recommended landmarks for him/her

Use Case Name: Explore Recommended Landmarks

Participating Actors: Tourist

Entry Condition:

- Tourist is connected to the Internet.
- Urbscope is running, and user is in Exploration Mode.
- The Recommendation switch is activated.

Exit Condition:

- Tourist closes Urbscope.
- Tourist activates the Nearby switch.
- Tourist changes the mode from Exploration to Recognition.

Main Flow of Events:

- Optionally, tourist specifies the filters pertaining to landmarks.
- Tourist points slides his/her phone around himself/herself.
- For each recommended landmark that satisfies the filters, different markers pop up as tourist is rotating.

Alternative Flow of Events 1:

- Optionally, tourist specifies the filters pertaining to landmarks.
- Tourist points slides his/her phone around himself/herself.
- No landmarks present that satisfies the filters selected by the tourist. Therefore, no markers is shown on the screen.

Alternative Flow of Events 2:

- Tourist loses the Internet connection.
- No markers is shown on the screen.

- Tourist explores nearby landmarks

Use Case Name: Explore Nearby Landmarks

Participating Actors: Tourist

Entry Condition:

- Tourist is connected to the Internet.
- Urbscope is running, and is in Exploration Mode.
- The Nearby switch is activated.

Exit Condition:

- Tourist closes Urbscope.
- Tourist activates the Recommendation switch.
- Tourist changes the mode from Exploration to Recognition.

Main Flow of Events:

- Optionally, tourist specifies the filters pertaining to landmarks.
- Tourist points slides his/her phone around himself/herself.
- For each recommended landmark that satisfies the filters, different markers pop up as tourist is rotating.

Alternative Flow of Events 1:

- Tourist loses the Internet connection.
- No markers is shown on the screen.

Alternative Flow of Events 2:

- No landmarks present that satisfies the filters selected by the tourist.
- No markers is shown on the screen.

- Tourist navigates to a location

Use Case Name: Landmark Navigator

Participating Actors: Tourist

Entry Condition:

- Tourist is connected to the Internet.
- Urbscope is running, and is in Exploration Mode.

Exit Condition:

- Tourist closes Urbscope.
- Tourist halts the navigation by clicking the stop button.

Main Flow of Events:

- Tourist selects the marker of the location that he/she wants to go.
- Navigation button shows up beneath the information displayed inside the information box.
- Tourist clicks the navigation button.
- Tourist is directed to the map with directions.

Alternative Flow of Events:

- Tourist loses the Internet connection.
- Navigation stops working.

- Tourist sets landmark filters

Use Case Name: Setting Filters

Participating Actors: Tourist

Entry Condition:

- Tourist is connected to the Internet.

- Urbscope is running, and is in Exploration Mode.

Exit Condition:

- Tourist closes Urbscope.
- Tourist switches back to Recognition Mode from Exploration Mode.

Main Flow of Events:

- Tourist opens navigation bar from the left side of the screen.
- Navigation Panel slides into the view.
- Tourist specifies filters for landmarks.
- Tourist closes the navigation bar.

- Tourist deletes his/her visit history

Use Case Name: Clear History

Participating Actors: Tourist

Entry Condition:

- Urbscope is running and it can be in one of the modes (Detection or Exploration).

Exit Condition:

- Tourist closes the settings menu by clicking the settings button again or somewhere else rather than settings menu.
- Tourist closes Urbscope.

Main Flow of Events:

- Tourist open the settings menu by clicking the settings icon which is shown in the top right corner.
- Tourist clicks on the clear history button.
- A message appears indicating the success.

- Tourist checks the About section

Use Case Name: Read About

Participating Actors: Tourist

Entry Condition:

- Urbscope is running, and it can be in one of the modes (Detection or Exploration).

Exit Condition:

- Tourist closes the settings menu by clicking the settings icon which is shown in the top right corner.
- Tourist closes Urbscope.

Main Flow of Events:

- Tourist open the settings menu by clicking the settings icon which is shown in the top right corner.
- Tourist clicks the About button.
- A view with information about the developers and the project is displayed.
- Tourist is astonished with the developers' skills and follows them on Github.

3.5.3. Object and Class Model

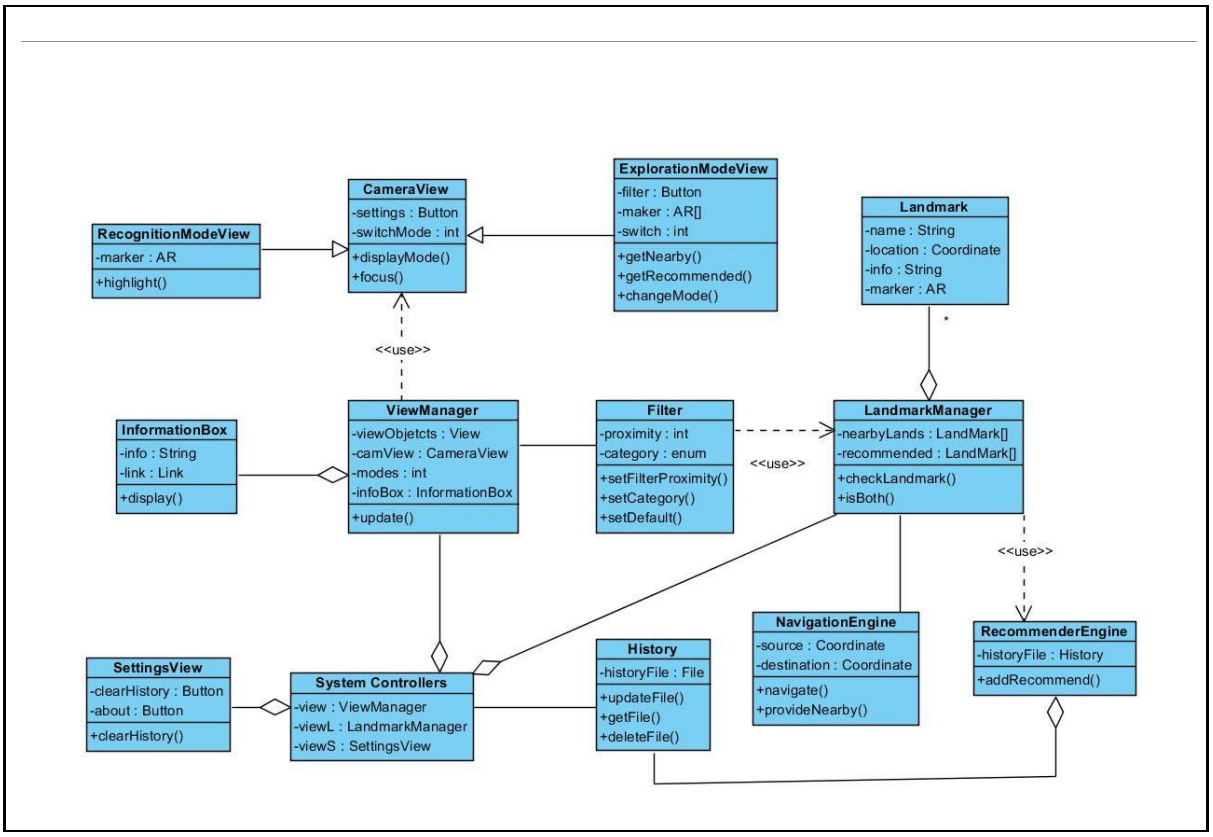

Fig 3.5.3.3.1

3.5.4. Dynamic Model

- State Diagram

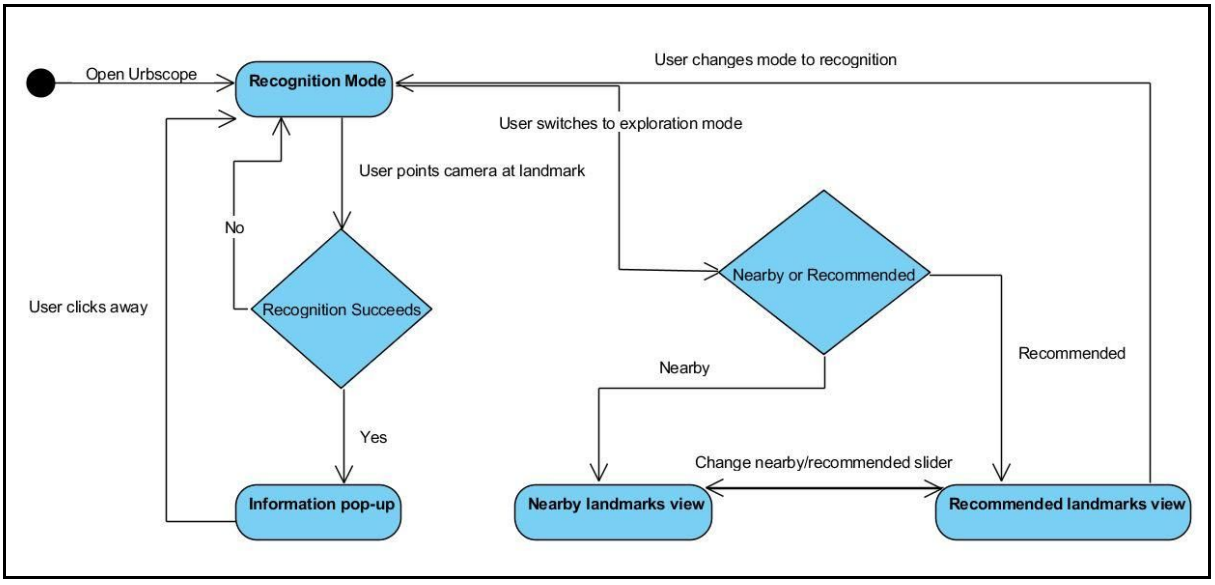

Fig 3.5.4.1

- Sequence Diagrams

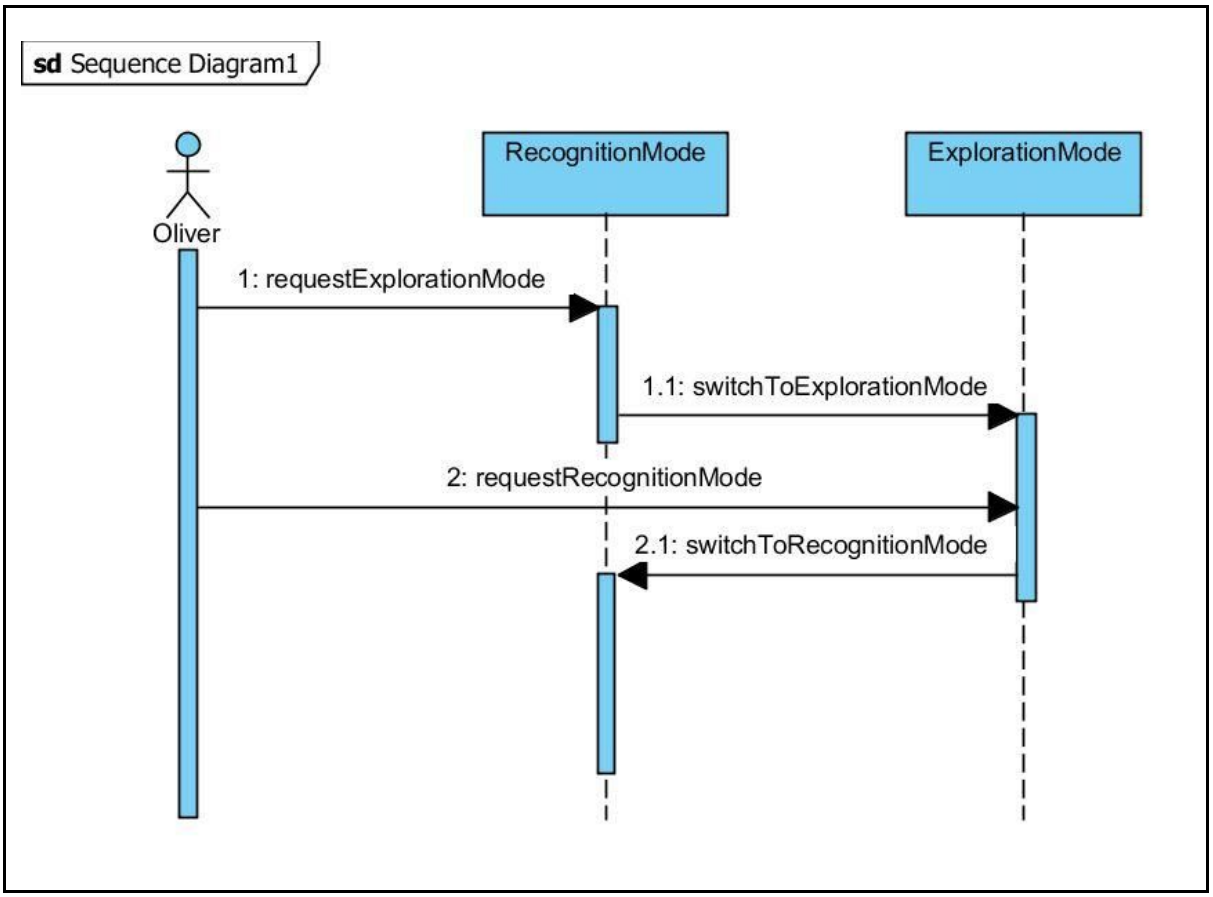

Fig 3.5.4.2

Sequence diagram 1, as seen in *Fig 3.5.4.2*, illustrates the user switching between the two main modes of Urbscope. When the user opens Urbscope, the default mode is the Recognition Mode. The user can request Exploration Mode and the system switches to the Exploration Mode. The user can then request to go to the Recognition Mode again, resulting in a switch back to the Recognition mode. As can be seen no mode's creation depends on the other mode. Both modes are initially created.

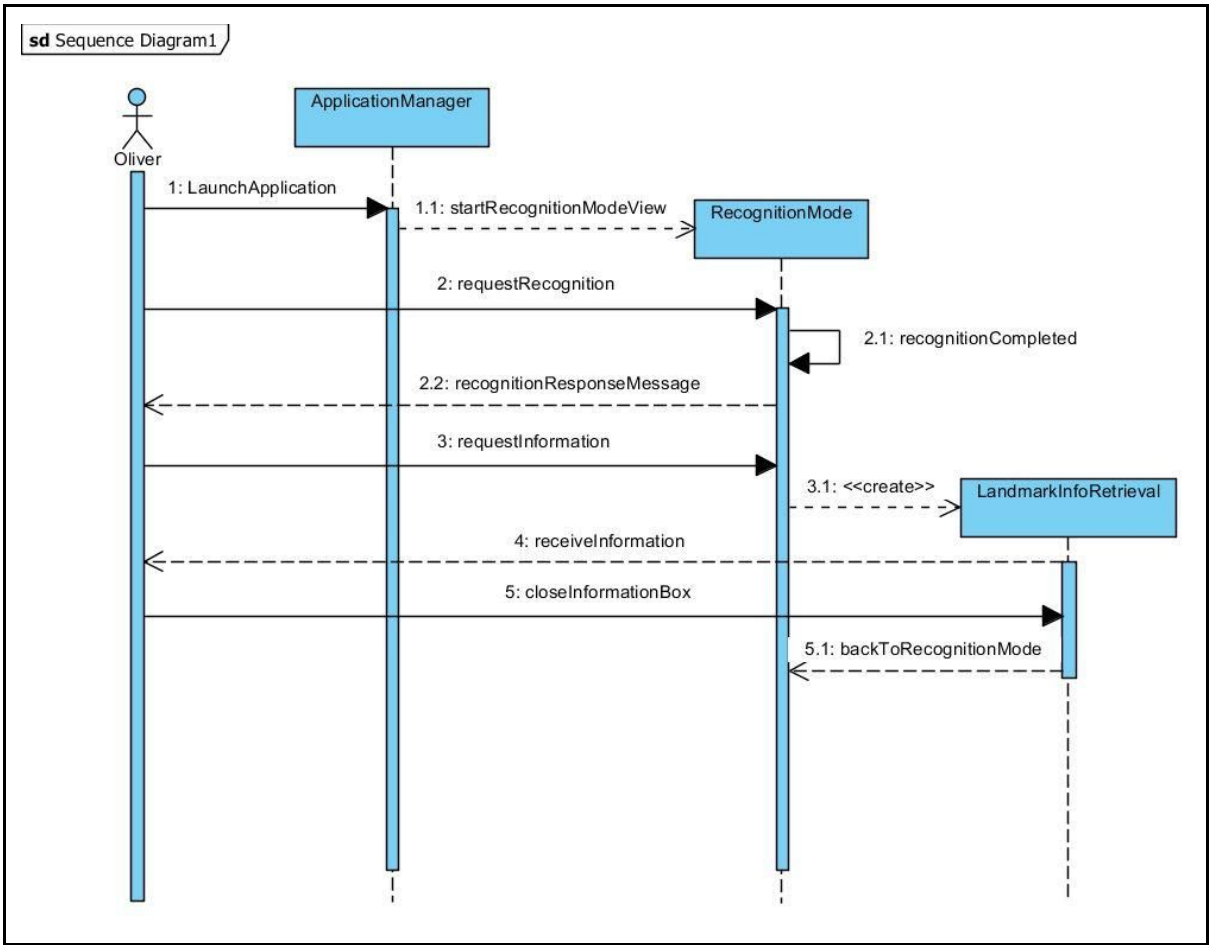

Fig 3.5.4.3

The Sequence diagram in *Fig 3.5.4.2* illustrates the way the user navigates through the Recognition Mode. As can be seen, the Recognition mode is the initial mode when the user launches Urbscope. The Application Manager is responsible for creating the Recognition Mode. Thereafter the user can request landmark recognition and is then sent back a response after the system recognises the landmark. Once the user retrieves the information, a request can be then sent for information. This then creates the Landmark Info Retrieval that procures the correct information and sends it back to the user. The user can then close the information box which results in the system then returns back to Recognition Mode.

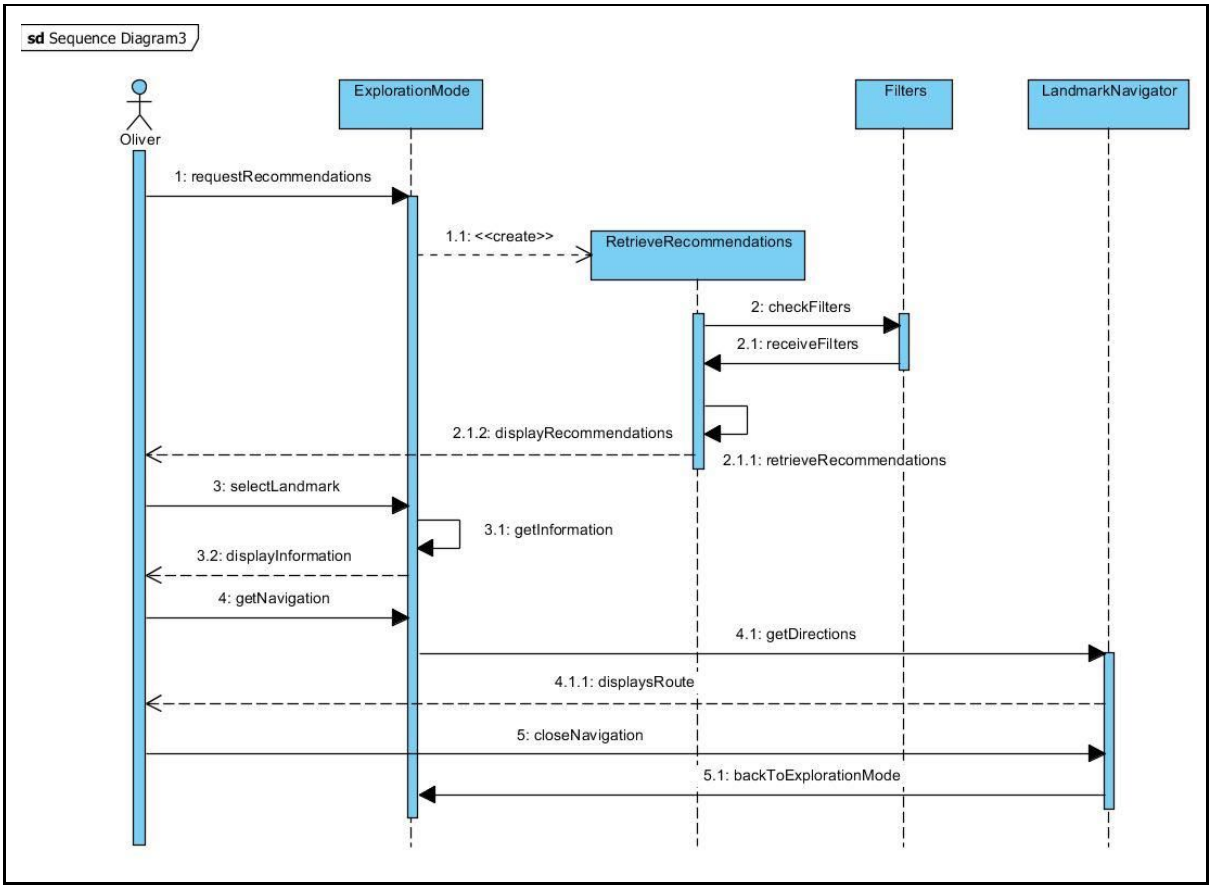

Fig 3.5.4.4

The sequence diagram shown in *Fig 3.5.4.4*, illustrates the sequence of the user in the Exploration Mode while he uses the Recommendations feature. Initially the user requests for recommended landmarks, and this creates the Recommendation Engine to retrieve recommendations. The Recommendations Engine then checks the filter parameters along with the information in the user's log file and displays respective landmarks. The User can then select a marker which then causes the Exploration Mode to get information pertinent to the selected landmark. The information is then displayed and the user can ask for directions which then results in the creation of the Landmark Navigator. The newly created Landmark Navigator then displays the routes back to the user. The user can close the navigations options and this sets the system back to the Exploration Mode view.

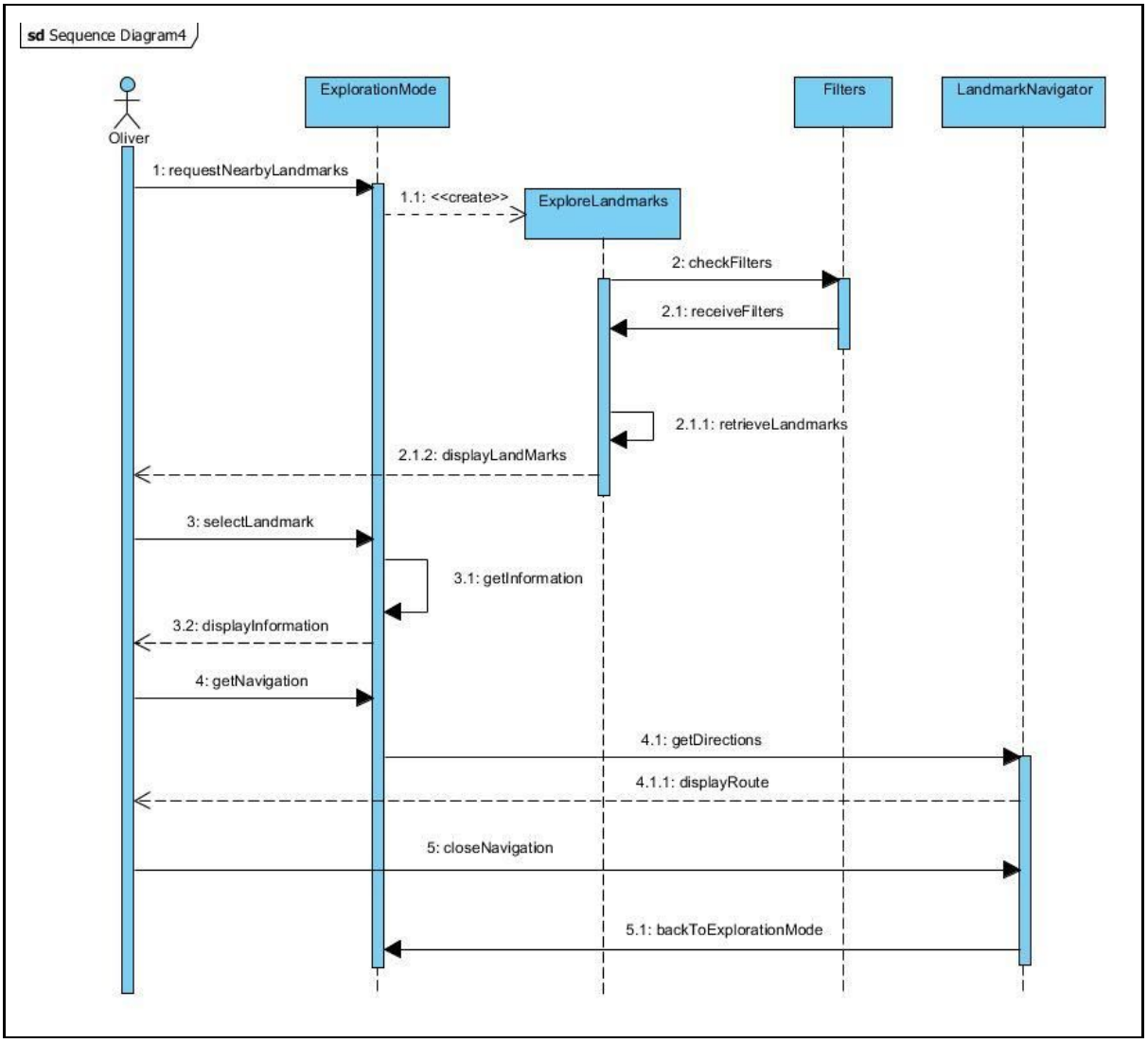

Fig 3.5.4.5

The sequence diagram shown in *Fig 3.5.4.4*, illustrates the sequence of the user in the Exploration Mode while he uses the Nearby Landmarks feature. Initially the user requests for nearby landmarks, and this creates Explore Landmarks. The filter parameters are checked and the landmarks which fall under these parameters are displayed. The following sequences then resemble the flow of sequence described in *Fig 3.5.4.4*.

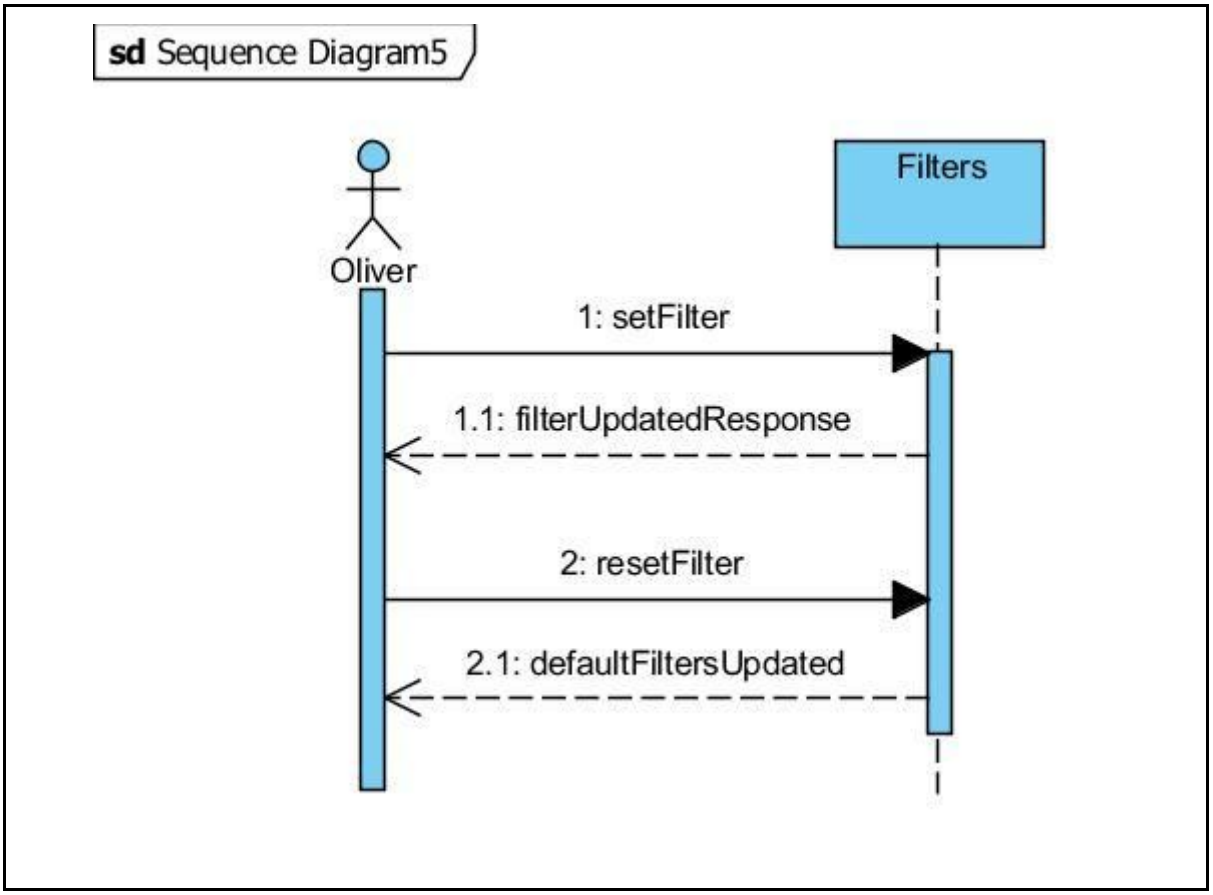

Fig 3.5.4.6

The final sequence diagram, as seen in *Fig 3.5.4.6*, illustrates the way the user would set the filter parameters. As can be seen the user can also reset the parameters back to default.

3.5.1. User Interface - Navigational Paths and Screen mock-ups

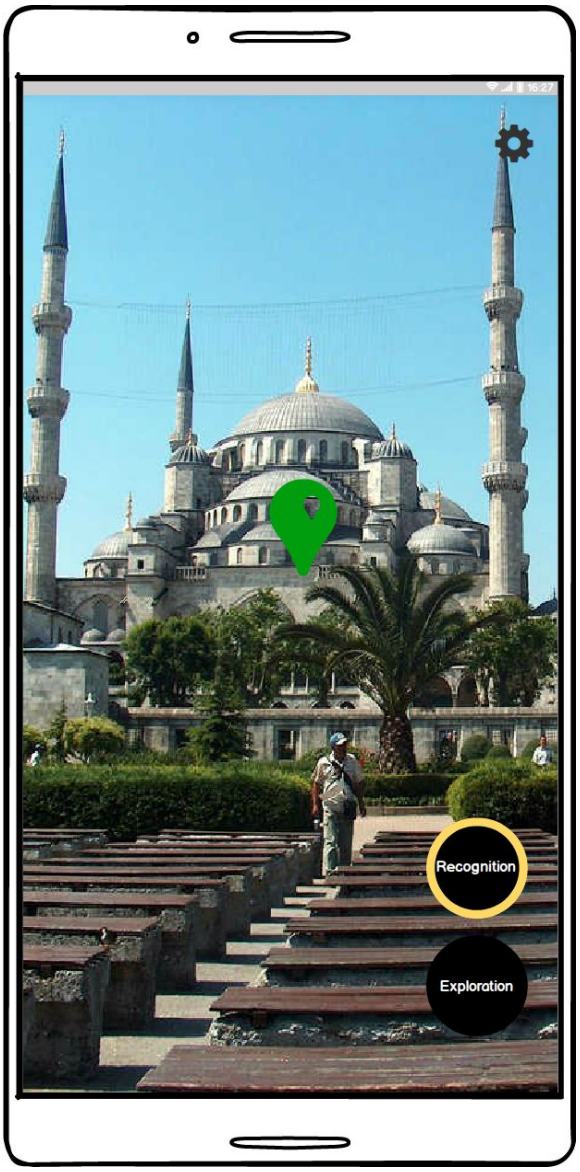

Figure 3.5.5.1: The main screen when the user opens the application.

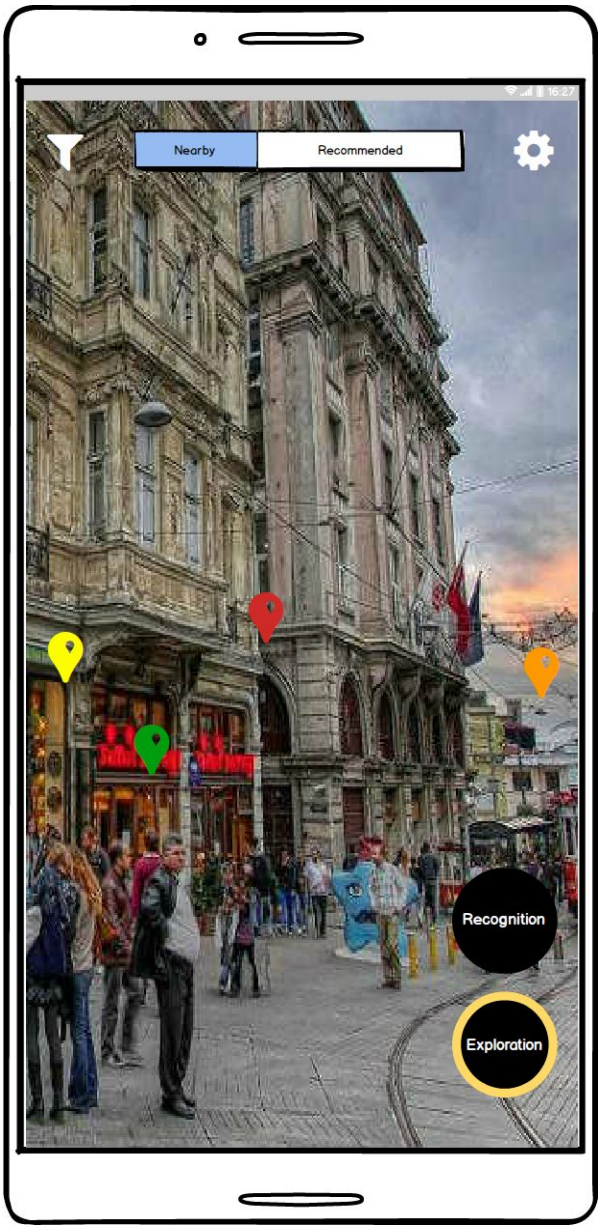

Figure 3.5.5.2: This is the Exploration mode in the application that shows the nearby and recommended locations.

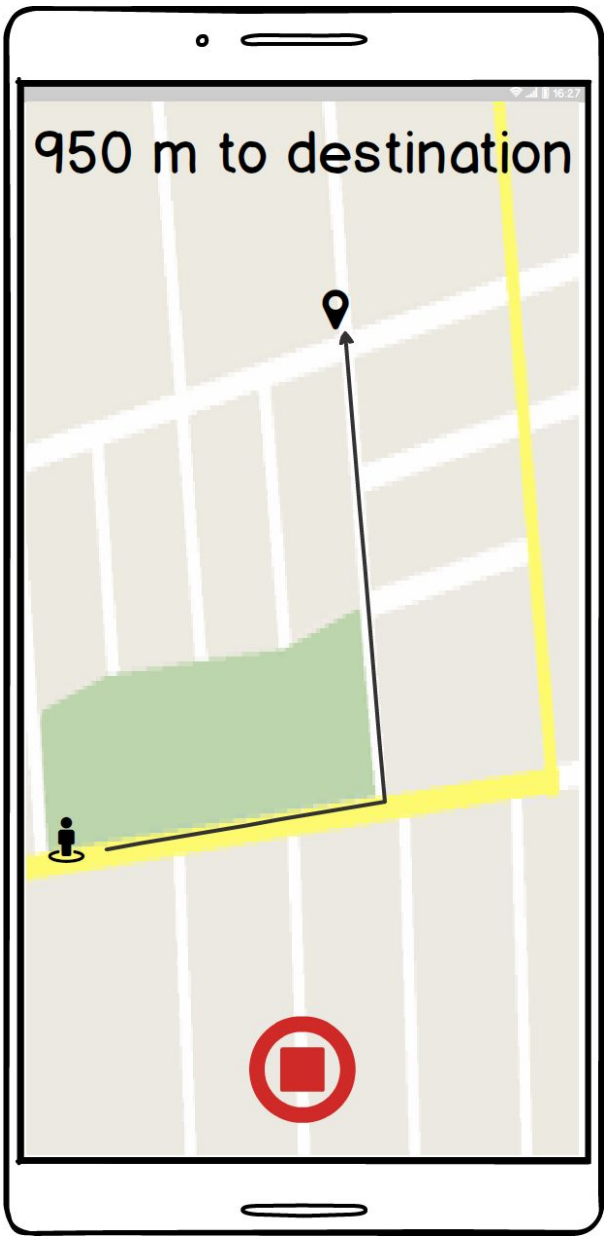

Figure 3.5.5.3: This is the map screen that shows the directions from the user's current location to the location of the landmark, which user selects from exploration screen.

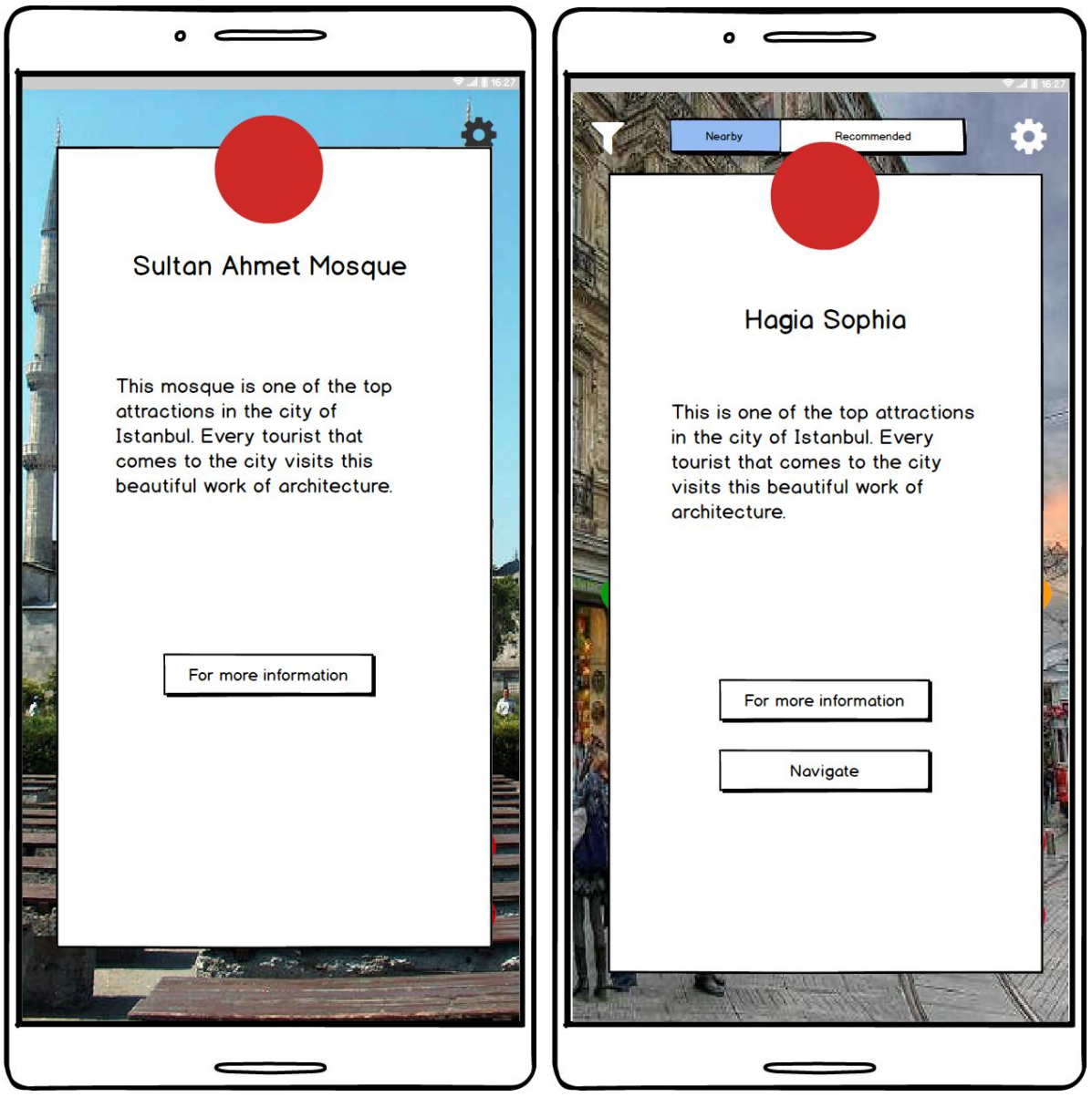

Figure 3.5.5.4: The information boxes that open up upon the selection of a landmark..The left one is from detection mode and the right is from the exploration mode.

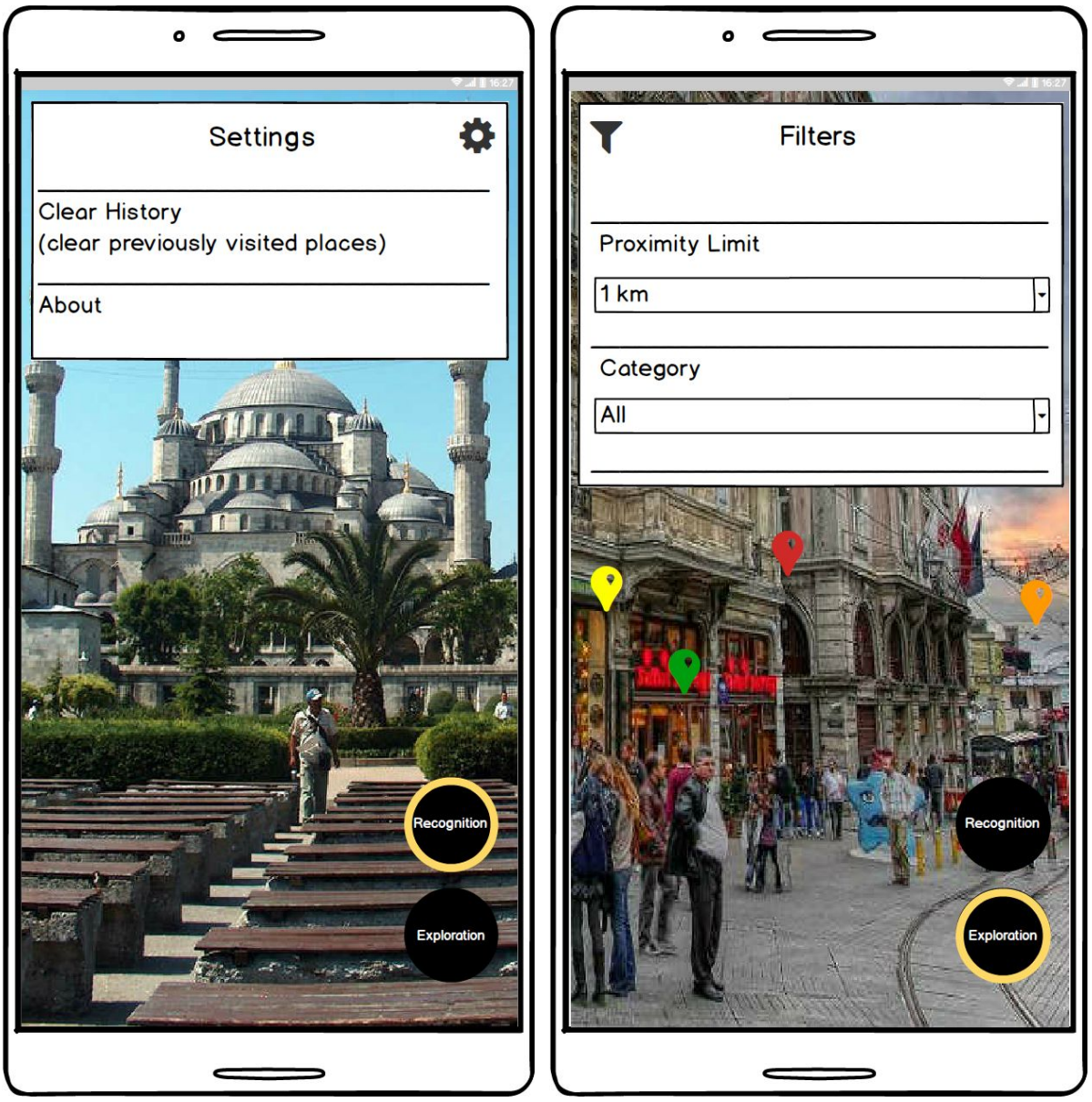

Figure 3.5.5.1: These are the settings and filter Panel to customise user experience

4. Glossary

AR Marker: This is basically a bubble on the screen. However, this bubble is on the AR layer and it is stationary with respect to users. Users have to turn towards the AR marker in order to see it on the screen.

Exploration Mode: One of the two modes that Urbscope has. This mode is used when users want to see the nearby or recommended landmarks' AR markers on the screen.

Filters: Filters are in essence the options that users choose in order to restrict the recommended or nearby landmarks. They include category types such as historical monuments, museums, sights, towers, religious landmarks etc. and proximity limit.

Information Box: This is a popup box shown when the user wants to acquire more information about a specific landmark and clicks on the corresponding landmark's AR marker. Inside the box, the relevant information is displayed. At the end, there is a link to an external source for more information. There may be a navigation button after the link if user is not in front of the landmark.

Landmark: It is essentially a well-known touristic place.

Nearby Switch: One of the two options that the switch which is shown at the top of the screen has when Urbscope is in Exploration Mode. It is activated when nearby landmarks are sought to be explored by the users.

Proximity Limit: This limits determines the diameter of the circle which is used in the Exploration Mode. Only the landmarks that are inside of this circle are shown in that mode.

Recognition Mode: This is the default mode of Urbscope. It is used for recognising the landmarks that user points out via the mobile phone's camera.

Recommendation Switch: One of the two options that the switch which is shown at the top of the screen has when Urbscope is in Exploration Mode. It is activated when user wants to see recommended landmarks for him/hem.

Side Navigation Panel: This a slide out navigation bar available on the left of the screen when Urbscope is in the Exploration Mode. It contains filters and proximity limit settings.

5. References

1. T. Claburn in San Francisco 8 Jun 2017 at 06:01 tweet_btn(), “The harsh reality of Apple's augmented reality toolset ARKit: It's an incredible battery hog,” *The Register® - Biting the hand that feeds IT*. [Online].Available: https://www.theregister.co.uk/2017/06/08/apple_arkit/.
2. “Augmented Reality Islamic App Mosque Finder,” *Islamic GPS*. [Online]. Available: https://www.islamicgps.com/.
3. “Read Reviews, Compare Prices & Book,” TripAdvisor. [Online]. Available: https://www.tripadvisor.com/.
4. M. van Rijmenam, “Three Ways TripAdvisor Uses Big Data To Become The World's Largest Travel Site,” *Datafloq - Connecting Data and People*. [Online]. Available: https://datafloq.com/read/three-ways-tripadvisor-uses-big-data/246.
5. “15 things you didn't know about TripAdvisor,” *The Telegraph*, 04-Feb-2016. [Online].Available: http://www.telegraph.co.uk/travel/lists/15-things-you-didnt-know-about-TripAdvisor/