Tour Genie: Tourist Guide Application

R. A. M. Lahiruni

Department of Computer Science and Engineering, University of Moratuwa
Sri Lanka
malsha.13@cse.mrt.ac.lk

Abstract—'Tour Genie', is a flexible and customizable travel partner for individuals that takes delight in travelling. It automates the task of locating the best places to visit by providing recommendations based on user location. The application is a combination of web and mobile applications that lets tourists access updated information while travelling. The software consists of a user friendly interface and a comprehensive database, and employs data from Google APIs in supplying the user with information. The user can use GPS services in mobile phones to track their current location and acquire information of recommended places using multimedia such as images and maps. The application is beneficial for tourists and local travelers who are unfamiliar with their places of visit. The comprehensive information provided offers tourists with a secure and comfortable setting in their places of visit. Thus, Tour Genie is an effective and efficient solution that is easy to use for tourists while travelling.

Keywords—*Google API; GPS; MVC; HTML; CSS; Travel; Android*

I. INTRODUCTION

Travelers and tourists often travel to locations that are not familiar and known to them. Once they are in a foreign environment, it becomes a difficult task for them to decide on the places to visit or obtain refreshments. As the choice of a wrong place would end up costing them both money and time, picking a place worth their expense is a challenging task. Thus, having an application that could be a travel guide which recommends them the best places to visit, would be of much convenience for them.

The application 'Tour Genie' was developed as a solution to the above problem. The need for a personal travel guide that could give custom recommendations to the traveler, was the motivation behind the development of the application. The travelers were in need of a system that was flexible and could be used while traveling. Thus the system had to be developed to meet such demands in the target user segment.

'Tour Genie' is a combination of web and mobile applications that use location intelligence to provide the user with meaningful information using internet connection. The application lets tourists access updated information while travelling. It consists of a user friendly interface and employs data from Google APIs in supplying the user with information. The user can use GPS services in mobile phones to track their current location. The application then provides a detailed description of the user's current setting and provides other places of interest to visit, and recommendations for where to eat, stay, shop and re-fuel.

The information is provided using multimedia such as images and maps. The application automatically records places of visit and save the details as user's history of visited places. Users can also save tours covering several locations as a trail, and easily view the route they have taken in different tours. User can search for places of interest and save them for future reference.

This paper describes how Tour Genie application was designed and developed. Section II surveys related work in building similar travel guide applications. Section III provides an overview of the system models. System design, implementation and testing of the application are described in latter sections respectively. Final section of the paper describes conclusion and future work.

II. LITERATURE REVIEW

This section provides an overview of existing applications with similar functionalities.

Foursquare [1] is a mobile and web application that provides user with location based information, Foursquare community and several other features. The user must have location services enabled (in mobile) to use the system. It gives recommendations for places such as places with food, coffee, fun and shopping. User can access the details of these places. User can also join its community and save places of interest, keep a history of visited places and create lists of places. User can also give reviews, tips and comments about a place. It uses its own Foursquare API in providing the information.

Yelp [2] is a popular mobile and application similar to Foursquare. Services are available in only a limited list of countries, and is not available in Sri Lanka. Other than what is available in Foursquare, it provides events near the user and other news alerts about nearby places. Tourias Sri Lanka [3] is a mobile application for tourists in Sri Lanka, providing users with location based information about a selected list of places in Sri Lanka. This contains a detailed description about the place and other useful information. Restaurant Finder [4] is a mobile application to find restaurants nearby and get related details. User can also search for restaurants by typing in an address or zipcode. It provides details about restaurants such as available menus and other options such as meal takeaway.

The benefit of using Tour Genie rather than using the above mentioned applications is that, while providing almost similar services, Tour Genie lets you manage your tour details. It lets you save the tours the user has been on as a

collection of places, with the details of tour route and visiting dates. The user can view their past tour details when planning a new tour, and choose the best route that would take them to new places. Furthermore, Tour genie can be used in almost all the countries, unlike Yelp, which is available for a selected list of regions.

III. SYSTEM MODELS

The features in Tour Genie are of two categories; Basic and Additional features. The basic features are available to any user, while the additional features are available only to the registered users. The Basic features include: viewing current location, getting recommendations based on a selected location, searching for locations and creating user accounts. The additional features include: viewing user history, saving places and creating tours that consist of several places. These functional requirements are depicted in the use case diagram are shown in Figure 1.
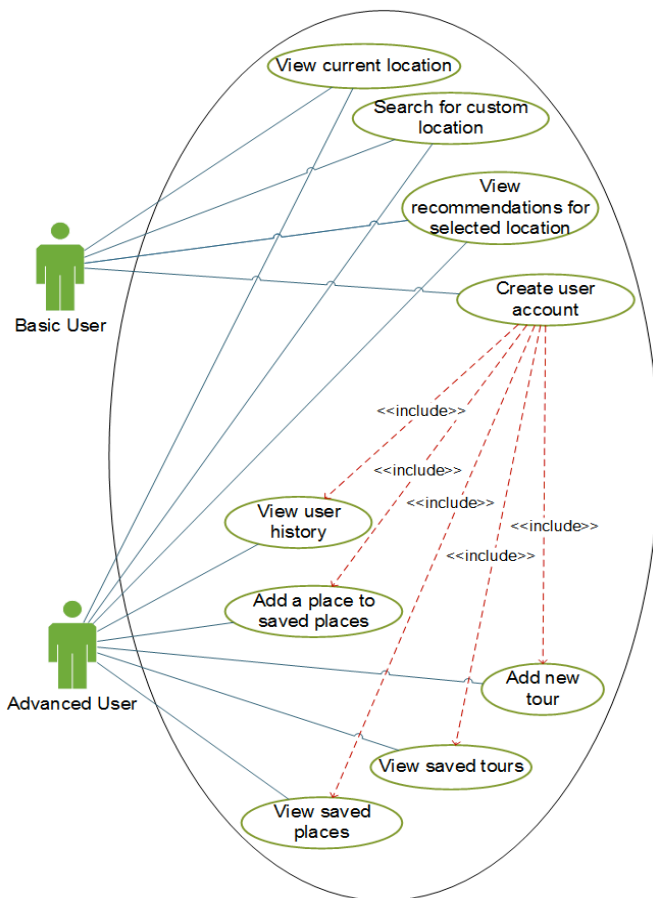

Fig. 1. Use Case Diagram

Apart from these functional requirements, there were several non-functional requirements for the system. To enhance the usability of the system, it was provided for both web and mobile platforms, with an intuitive user interface

and access to basic features without having to register. The application also had to be high in reliability, thus Google APIs were used to ensure the accuracy of the provided information. As the database stores user travel details, much attention was paid to the security of the database. The user passwords were stored using MD5 hashing, and the database queries were embedded out of the application code, so that the stored data are not vulnerable to SQL injection.

IV. SYSTEM DESIGN

A. Architecture of the System

The architecture of Tour Genie is based on the 4+1 view model, depicted in Figure 2.
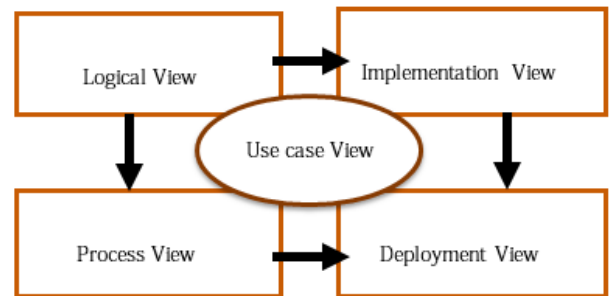

Fig. 2. 4+1 View Model

The use case view captured the functional and non-functional requirements of the system. The logical view focused on the elements in the system: the packages and classes. The process view described the flow of operations in the functionalities in the system. The implementation view focused on the different packages in the system and the deployment view described the distribution of the system.
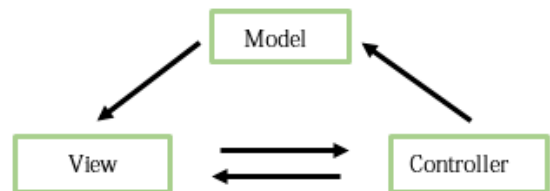

Fig. 3. MVC Pattern

The packages in the system were created according to the Model-View-Controller (MVC) architectural pattern (Figure 3). The elements in the application were arranged separately in these packages. Using this pattern allowed independent development and testing of the individual components and easy integration among them. The components in the application were arranged into packages as in Figure 4.
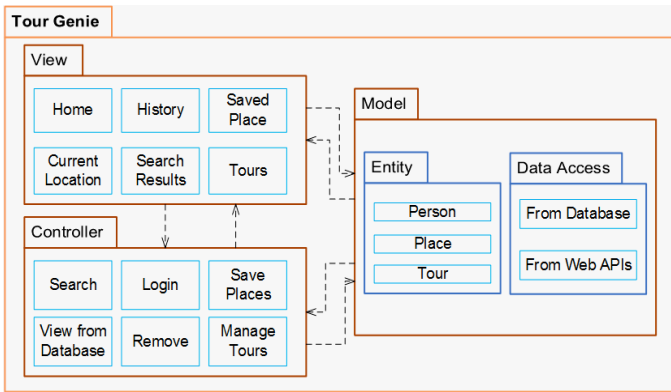
Fig. 4. Package Diagram

D. Data View

The database for Tour Genie stores information of users and user related information including user history, saved locations and tours. The Entity Relationship diagram for the database is shown in Figure 7.

The database is used to store the information of registered users. The registered user details are stored in User table where encryption is used to secure the data. If the user saves a place, add a place to history or creates a tour, the relevant tables are updated along with the place table. Place table stores the place ids that are retrieved from the Google APIs. This id is used in retrieving the place information in future references. The tour_stop table contains the details of each place in a tour.

B. Logical View

The class diagram in Figure 5 shows the entity classes that were created in the Model package of the application. The User, Tour and Place are the main classes in the system. The SavedPlace, VisitedPlace and TourStop classes extend the Place base class. A user has saved places, visited places and tours, shown by aggregation in the diagram. A Tour is a collection of TourStop objects, and is therefore shown using composition.
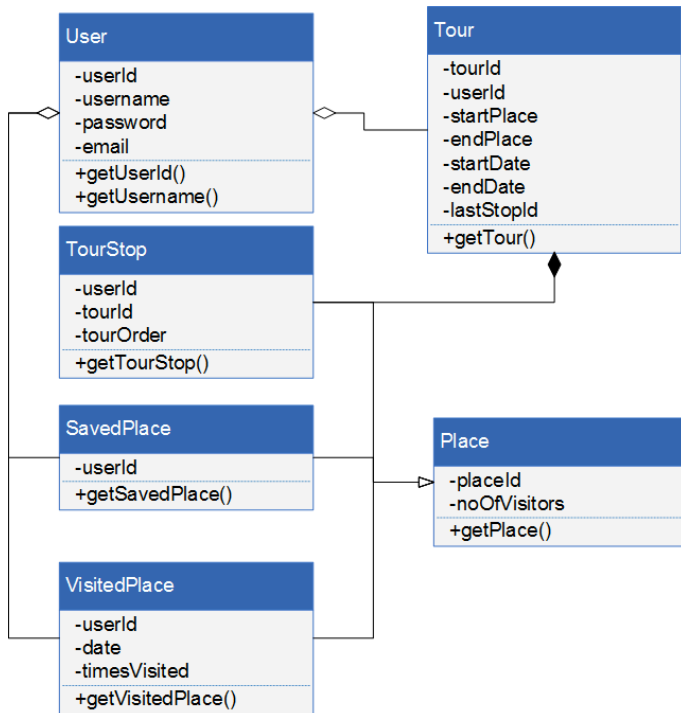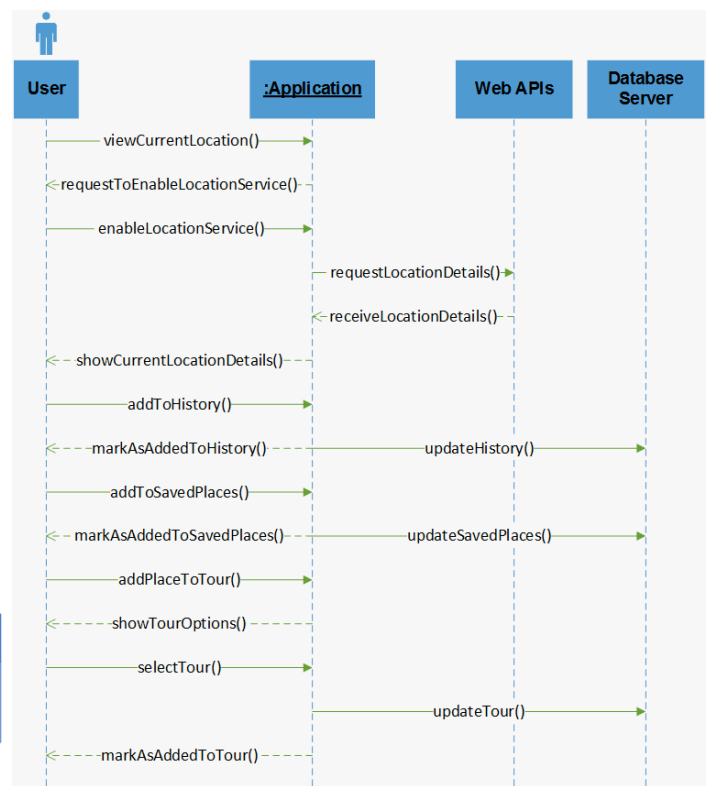

Fig. 6. Sequence Diagram

Fig. 5. Class Diagram

C. Process View

The main sequence diagram for viewing current location is shown in Figure 6. This involves adding the current location to user history, saving the place and adding it to a tour, covering the main functionalities.
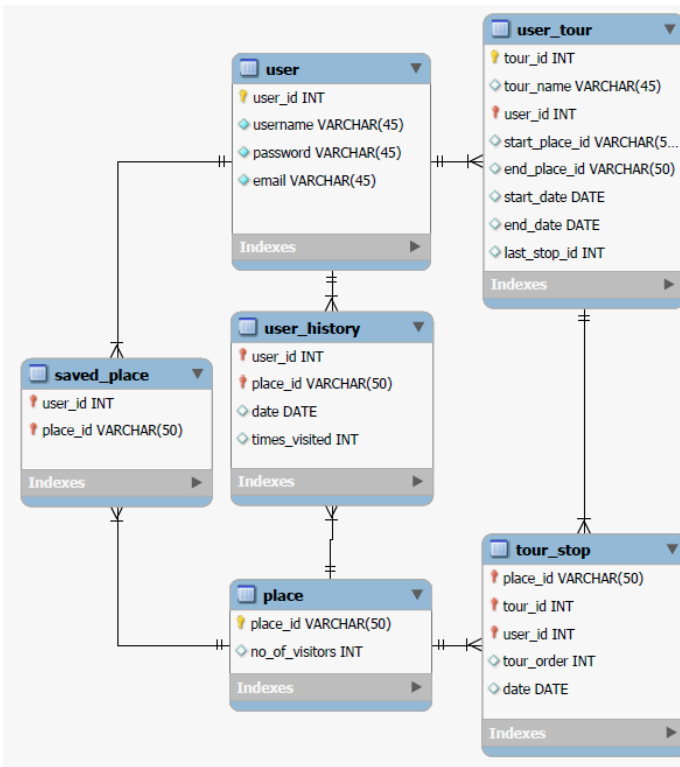
Fig.7. Entity Relationship Diagram

V. SYSTEM IMPLEMENTATION

A. Implementation Procedure

In implementing the system, various technologies and tools were used in combination. PhpStorm was used as the development environment for the web application, and Android Studio was used for the mobile application. MySql Workbench and phpMyAdmin were used to manage the database.

The web application is developed using languages such as HTML, CSS Bootstrap, JavaScript, JSON, Php. Further, Java and XML were used for the development of the mobile application. In obtaining the information of places and related details, a set of APIs were used; Google Places JavaScript Library [5], Google Maps API [6] , Google Places API for Android [7], Google Places Web Services API [8], Google Geocoder API [9], Google Street View Static Image API [10] and Google Knowledge Graph API [11].

The Google Knowledge Graph API was used in obtaining descriptions for places. Volley Library [12] was used in accessing Google Web Services API in the mobile application.

B. The Algorithm

Obtaining current location of user and getting recommendations for places nearby involves the use of location service functionalities and several APIs. Explained below is the algorithm used in web application for the above

functionality (Figure 8). While displaying the current location interface, the application is connected to the relevant API.

Providing recommendations for places was done according to the distance from user's current location to the place. Upon the selection of a place category to search, the search function filtered place types using the available types in the Places API.

```
function initMap(){
pos ← current location obtained using GPS
    map ← create Google map element
placeID ← get placeID of current location
        using Geocoder API
place ← get place object from Google Places
        API using the placeID
  placeName ← get place name from the API
     if (place.photo != undefined)
        placePhoto ← from places API
                   else
      placePhoto ← from StreetView API
carousel ← set images to carousel modal
                 window
      display details in interface
                   }
```

Fig.8. Algorithm for Obtaining Current Location

C. Main Interfaces

The main interfaces of the web application are as shown below. Figure 9 shows the home page of the application, from which user can search for places or log in. The Current Location button opens the current location interface, shown on Figure 10. Here user can view their current location and related details. User can search for places near them using the Look Around Me option. If the user searches for a place, the search results page is displayed (Figure 11). The related places and their location is shown on the map, while information about is place is also displayed. A logged in user can view the places they have saved, or places they have visited (Figure 12). User can further view details about tours they have saved (Figure 13).
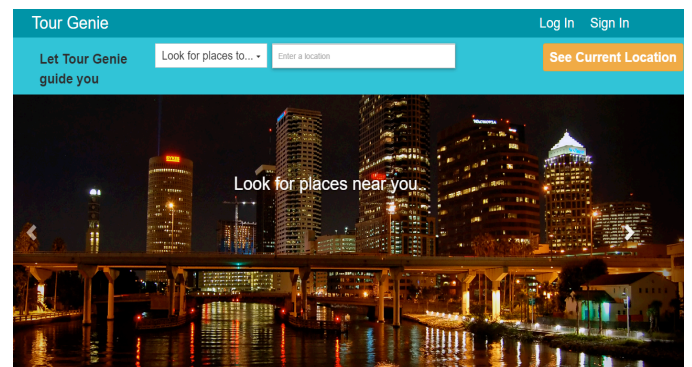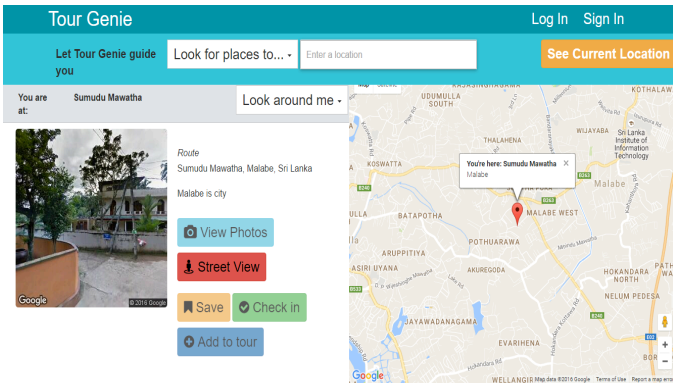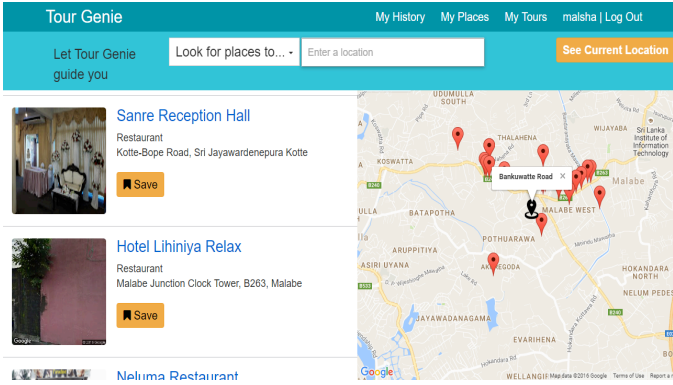

Fig.9. Home Page

Fig.10. Current Location Interface

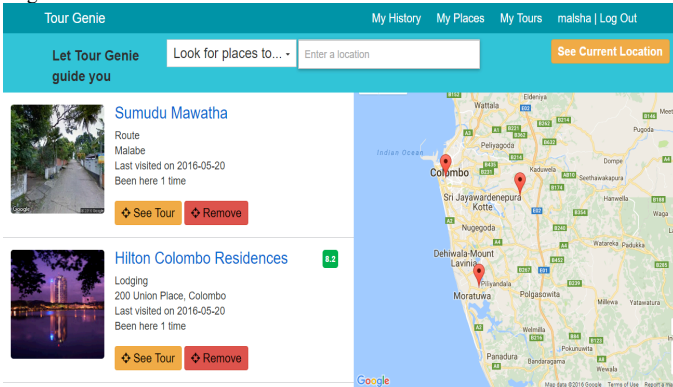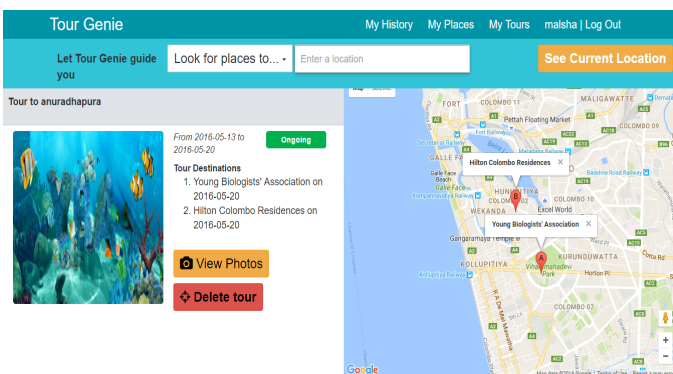Fig.11. Search Results Interface

Fig. 12. My History Interface

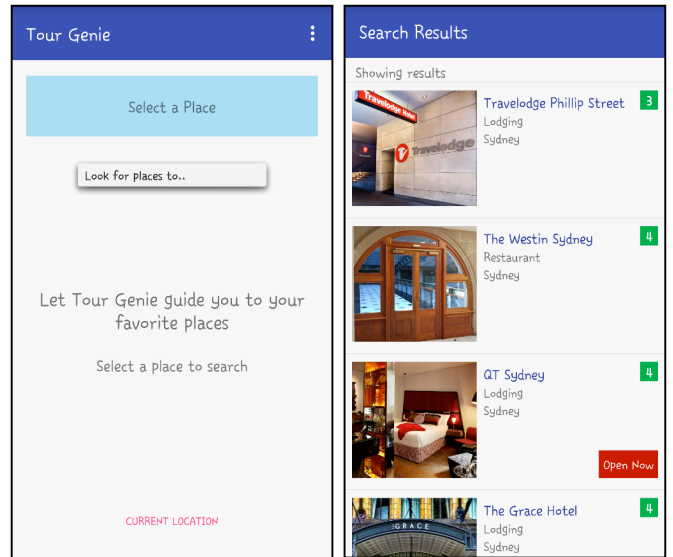Fig. 13. Detailed Tour Interface

Fig. 14. Home and Search Results View

The interfaces in the mobile application are shown as follows. Figure 14 shows the home view, and the search results view. The current location and detailed location interfaces are shown in Figure 15.

Fig.15. Current Location and Detailed Location

VI. SYSTEM TESTING AND ANALYSIS

There were several objectives of testing Tour Genie application. One is to verify that the software meets its set of requirements specified earlier in development phase. These requirements were expected by the user, thus it was important to verify that they are available in the application. Testing also ensured correct execution of each functionality in the application by executing unit testing. It helped to discover and fix bugs and check the compatibility of the product across different browsers and devices. It also helped in discovering and fixing the edge cases where users might

use the software in an irregular way leading to bugs in the application. Testing further ensured that system performs well when several users using the application at the same time.

Various tools were used for the testing of the application. For interface testing, the Selenium IDE tool for Firefox was used and for unit and functional testing, PhpUnit [13] was used. Interface testing was done for each interface in the system using Selenium IDE in Firefox browser. Unit testing was done for each data access function using PhpUnit, to verify that the expected results are generated by the function. This ensured that all database access functions perform well and without errors. Functional testing was done using PhpUnit, for processes that involved using several data access functions at one time.

VII. CONCLUSION AND FUTURE WORK

Tour Genie was developed to be a flexible and customizable travel partner for travelers and tourists. Currently, the application functions well in the web and Android platforms as mentioned, and performs the functionalities defined in above sections. It has automated the task of finding the best places of visit and provides information using multimedia. It also saves user preferences and provides an interface to retrieve these data at any time. Thus it can be said that Tour Genie is an effective and efficient solution for travelers to use while traveling.

Tour Genie can be improved in several ways in future. It can be extended to a community of travelers, where every user can share their tours and ideas about places. The recommending feature could be enhanced to consider user interests in visiting a place. It can also be enhanced with other related data to places such as weather of a place, distance to places, traffic conditions on reaching a place and available facilities in a selected place; such as facilities in a hotel or items for sale in a shop. Furthermore, it can incorporate an 'Events Nearby' option, where users will be able to view a set of events such as exhibitions or musicals near them and get related information. This would, however, require a comprehensive and updated data source of such events. Tour Genie can also be developed into a tour planner, where user can pick places from their search results and create a travel plan. Thus, Tour Genie can be extended into a comprehensive and complete tour guide by including such features and functionalities in future.

REFERENCES

[1] Foursquare Inc., "Foursquare," [Online]. Available: https://foursquare.com/. [Accessed 19 June 2016].

[2] Yelp Inc., "Yelp," [Online]. Available: http://www.yelp.com/. [Accessed 19 June 2016].

[3] Google, "Google Play - Tourias Sri Lanke Travel Guide," [Online]. Available: https://play.google.com/store/apps/details?id=com.tourias.android.guide.sri_lanka. [Accessed 19 June 2016].

[4] Google, "Google Play - Restaurant Finder," [Online]. Available: https://play.google.com/store/apps/details?id=com.akasoft.topplaces. [Accessed 19 June 2016].

[5] Google, "Google Places Javascript API," [Online]. Available: https://developers.google.com/maps/documentation/javascript/places. [Accessed 19 June 2016].

[6] Google , "Google Maps API," [Online]. Available: https://developers.google.com/maps/. [Accessed 19 June 2016].

[7] Google, "Google Places API for Android," [Online]. Available: https://developers.google.com/places/android-api/. [Accessed 19 June 2016].

[8] Google , "Google Places Web Services API," [Online]. Available: https://developers.google.com/places/web-service/. [Accessed 19 June 2016].

[9] Google, "Google Geocoder API," [Online]. Available: https://developers.google.com/maps/documentation/geocoding/start. [Accessed 19 June 2106].

[10] Google, "Google Street View Static Image API," [Online]. Available: https://developers.google.com/maps/documentation/streetview/. [Accessed 19 June 2106].

[11] Google, "Google Knowledge Graph API," [Online]. Available: https://developers.google.com/knowledge-graph/. [Accessed 19 June 2016].

[12] androidhive, "Android working with Volley Library," [Online]. Available: http://www.androidhive.info/2014/05/android-working-with-volley-library-1/. [Accessed 19 June 2016].

[13] PhpUnit, "The Php Testing Framework," [Online]. Available: https://phpunit.de/. [Accessed 19 June 2016].