Department of Informatics, University of Leicester

CO7501 Individual Project

City Tour Planner – Android Application

Author Name: Muhammad Sohaib Furqan

University Email: msf7@student.le.ac.uk

University ID: msf7

Preliminary Report

Word Count: 2157

Supervisor: Dr. Rayna Dimitrova

Second Marker: Prof. Reiko Heckel

Submission Date: 07 March, 2019

DECLARATION

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people s work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people s work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Muhammad Sohaib Furqan

Signed:

Date: 25 February, 2019

Abstract

This document aims to provide a concise insight into the project. It is hoped that the reader of this document will develop an appreciation of the motivation behind the project and the main challenges that may arise over the course of implementation. A tentative list of requirements (both high-level and detailed), has been laid out. The technical specifications of the project, including an initial description of the system architecture and the technologies to be used are detailed. The foundations of the project rest upon a survey of the background material involved (including the study and analysis of applications of a similar nature), and hence made a part of this report. The report concludes with a timetable that provides a crude first estimate of the duration and order of tasks under standard circumstances, and a risk plan indicating anticipated events, their impact on the project and suggested mitigation strategies to address the risks if needed.

Table of Contents

Motivation and challenges

Why is this worth doing?

Ever since the use of the internet has become widespread and the computing industry has experienced a fundamental paradigm shift towards the use of mobile applications instead of traditional desktop software that were prevalent in the past, we see a raft of mobile applications across platforms that address a wide range of user needs. These mobile applications span almost all areas where desktop programs were customarily deployed and have changed the fundamental way people go about doing their daily tasks. Where a user would have had to bind themselves to a computer screen for carrying out their tasks, the ease of access and portability offered by smart phones have given a new dimension to routine computing.

The flip-side of this increasingly technology driven era though, is that there is hardly any spare time for people to attend to their hobbies. It would be good if this technology dominance can be molded and used to facilitate users in a way that allows them an efficiently managed break from their schedules, without impacting daily timetables in a negative way.

There are many applications that enable users to plan and organize trips, and all of them offer distinct features to facilitate a greater user experience. Redesigning the wheel and coming up with a similar application would amount to a waste of valuable resources and time. Hence, this project is directed towards building upon the offerings of existing applications, albeit with a different objective. The aim is to provide smartphone users with a software-driven solution which allows them to plan a one-day tour of a city. Success in doing so will be of value to users in that they will not be required to set aside a dedicated time period for trips, but instead will be able to organize short recreational trips around their daily workflows. While similar applications work around organizing and managing traditional, long-duration trips, our aim in this project is for the user to be able to explore everything that a city has to offer while adhering to individual time constraints and work commitments.

Challenges:

This project poses two fundamental challenges:

Integration of relevant information from various sources:

The application will use information from various available sources (e.g. Google Maps, foursquare, tripadvisor etc.) to generate a customized trip based on user preferences. This will require filtering out available information from and using it in an efficient and effective manner to provide maximum value for the user. Furthermore, integration of filtered (relevant) information from the various sources to generate the best available trip deals for the user will be a challenge. Time-constraints set by users will also have to be taken into account while developing the application.

Provide an estimate of the cost of a generated trip:

Costs of leisure trips can escalate very quickly because of the large number of factors that can be involved. Users may opt for all kinds of different facilities and experiences, and the choices can cause costs to vary dramatically. Also, different geographical locations have different prices for the same thing. What might be cheaper in one area might be relatively expensive in the others. The application should provide a close approximation of the costs. Given the diversity involved, such a task can pose a significant challenge in the development of the actual software.

List of requirements:

High-level requirements:
1. Allow users to set up a profile
2. Allow users to customize trip experience
3. Retrieve and integrate relevant information from different sources
4. Generate tour suggestions based on user interests and constraints
5. Optimize search results based on user preferences
6. Provide an estimate of the cost incurred by a specific trip
7. Include pictures of venues along with user reviews of the most popular spots
8. Incorporate user feedback and suggestions for improving the user experience of the application

Detailed Requirements:
1. **Allow users to set up a profile:**
   The application should allow each user to enter profile information including personal information such as name, age, gender, contact number etc. and also information pertaining to trip preferences and time constraints. This will also include the places and the type of activities that the user is interested in.
2. **Allow users to customize trip experience:**
   Users should be able to customize settings for individual trips based on various search filters .e.g. budget, location, type of trip, activities, food choices, specific facilities etc. If no available trip fulfils the customized criteria, the user should be asked to change the search criteria and try again.
3. **Retrieve and integrate relevant information from different sources:**
   The application will strive to provide the best user experience by using different sources to gather information. The information will be retrieved based on the preferences and search criteria specified by the user. All sources will be queried against the same criteria and the results of the search will be combined in the best possible way to ensure that the user gets the best suited customized trip. The search results should then be displayed in a user-friendly manner.

4. **Generate tour suggestions based on user interests and constraints:**
   The application will allow for storing user preferences and time constraint settings. These should be used to suggest the users regarding available trips. These trip suggestions will be generated automatically and be visible to users if they select the option of "view suggested trips".
5. **Optimize search results based on user preferences:**
   An appropriate level of detail should be presented to the user initially, with the choice to view more. This is especially important, as too little or too much detail is likely to have a negative impact on the user. Also, only the most relevant information should be retrieved and used by the application.
6. **Provide an estimate of the cost incurred by a specific trip:**
   The application should calculate the estimated cost of the selected trip with the selected settings. This estimate is liable to change based on changes to the user's search criteria and preferences.
7. **Include pictures of venues along with user reviews of the most popular spots:**
   The application should provide a complete user experience by displaying pictures and descriptions of the famous spots in a city, and should also have the option for tourists to post reviews about their experience at the specific venue. These reviews should then be available for other users to see so that they can make a more well-informed choice.
8. **Incorporate user feedback and suggestions for improving the user experience of the application:**
   Users should be allowed to provide feedback regarding their experience of using the application. This could include comments or requests for add-ons or new features to be made part of the application in subsequent releases.

Technical Specifications:

This section of the report discusses briefly the system architecture and the technologies that will be used for the development of the different components of the system. The discussion of the architecture is divided further to aid the understanding of the reader.

Architecture Overview:
The application is intended for use on the Android OS and will run on almost all android-powered devices. The target audience is any user interested in automated trip-planning. The aim of the application is for users to be able to quickly and efficiently find and customize trip experiences to their liking. Users will be able to set custom search criteria, and retrieve related information on the basis of these settings from multiple sources. Users will be asked to set preferences and these preferences will be taken into account during the search. The information will then be presented in a user-friendly way.

Technologies to be used:

This application targets the android platform and therefore, a significant part of it will be developed in Java using the Java Platform Standard Edition (Java SE). The database technology for the purposes of this application will mainly be SQLite but Object-Relational Mapping (ORM) systems such as ROOM might also be used if needed.

The application will make use of various APIs made available by different vendors. This includes (but is not limited to) Google Maps, Trip Advisor and Foursquare. These technologies will be used as deemed appropriate at the time of development so as to enhance the user experience of the application.

Other specifications:

The application will run on any device running Android 4.0.3 (IceCreamSandwich) or higher.

Background material:

Similar android applications including TripIt, RoadTrippers and Sygic Travels were studied as part of the literature review for this project. The following is a brief description of the main features offered by each of these applications. The reading list section indicates resources used to study android programming required for this project.

TripIt:

TripIt is a travel planning application which allows for creating an itinerary for each trip. Among other features, this application allows users to forward trip confirmation emails to a dedicated email address (plans@tripit.com). It then creates an itinerary out of the forwarded emails, separately for each trip. Users also have access to a master itinerary where they can access every trip. These plans can then be added to a calendar or shared with friends.

RoadTrippers:

This is a web and mobile based application intended to help users in planning road trips. Focusing specifically on areas in the US and Canada, users of this application can choose from over 5 million locations and can use category filters to customize preferences. The application also provides users with turn-by-turn navigation of their desired location.
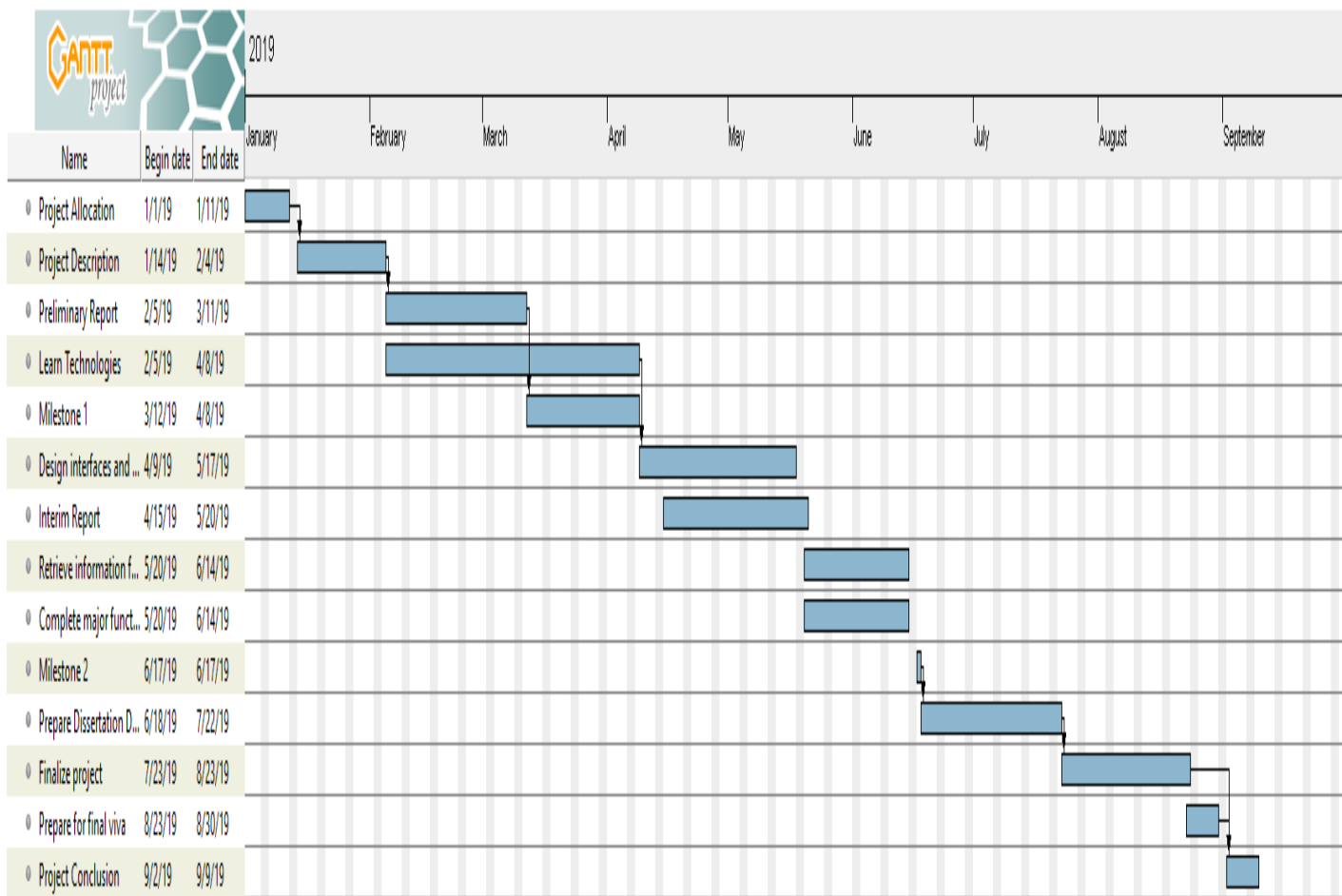
Sygic Travels:

A GPS navigation software that works along the lines of Google Maps but was the first to offer offline maps. Though Google has also started offering this feature now, there is still a huge difference. Sygic maps are far more memory efficient compared to google and also allow for storing larger maps for offline usage.

Reading List:

1. Google Codelabs – android fundamentals
2. Android developer guides
3. Material design guidelines

Project Schedule:

Name	Begin date	End date
Project Allocation	1/1/19	1/11/19
Project Description	1/14/19	2/4/19
Preliminary Report	2/5/19	3/11/19
Learn Technologies	2/5/19	4/8/19
Milestone 1	3/12/19	4/8/19
Design interfaces and ...	4/9/19	5/17/19
Interim Report	4/15/19	5/20/19
Retrieve information f...	5/20/19	6/14/19
Complete major funct...	5/20/19	6/14/19
Milestone 2	6/17/19	6/17/19
Prepare Dissertation D...	6/18/19	7/22/19
Finalize project	7/23/19	8/23/19
Prepare for final viva	8/23/19	8/30/19
Project Conclusion	9/2/19	9/9/19

Task Details:

The following table lists each task depicted in the Gantt chart above along with the respective start and end date.

Task	Start Date (DD-MM-YYYY)	End Date (DD-MM-YYYY)
Project Allocation	01-01-2019	11-01-2019
Project Description	14-01-2019	04-02-2019
Preliminary Report	05-02-2019	11-03-2019
Learn Technologies	05-02-2019	08-04-2019
Milestone 1	12-03-2019	08-04-2019
Design Interfaces and generic functionality	09-04-2019	17-05-2019
Interim Report	15-04-2019	20-05-2019
Retrieve information from online sources	20-05-2019	14-06-2019
Complete major functionality	20-05-2019	14-06-2019
Milestone 2	17-06-2019	17-06-2019
Prepare dissertation draft	18-06-2019	22-07-2019
Finalize project	23-07-2019	23-08-2019
Prepare for final viva	23-08-2019	30-08-2019
Project Conclusion	02-09-2019	09-09-2019

References:

Android Developer Guides, available at http://developer.android.com/guides (Accessed: 10 February, 2019)

Material Design Guidelines, available at https://material.io/design/guidelines-overview/ (Accessed: 23 February, 2019)

Google Codelabs – Android Fundamentals, available at https://codelabs.developers.google.com/android-training (Accessed: 10 February, 2019)

The Java Tutorials by Oracle, available at https://docs.oracle.com/javase/tutorial/ (Accessed 10 February, 2019)