A GPS-BASED MOBILE APPLICATION FOR TOURISTS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF APPLIED SCIENCES

OF

NEAR EAST UNIVERSITY

BY

KATHY KEFAS

In Partial Fulfillment of the Requirements for

the Degree of Master of Science

in

Computer Information Systems

Nicosia, 2015

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Kathy, Kefas

Signature :

Date:

ABSTRACT

Global Positioning System (GPS) is a space-based satellite system used for navigations that offers time information and location where there's a clear line of vision to four or higher GPS satellites. The research on human space–time behavior has rapidly increased in recent times due to the availability and development of cost-effective and reliable tracking devices which resulted in the use of GPS and other tracking technologies in various fields. In view of this and considering that tourism is a relevant subject today, as it serves as a great source of revenue to any economy, especially in Cyprus, there is need to take advantage of the GPS and develop a mobile application in order to aid tourists in their decision making and acquisition of information, without going through the stress of carrying heavy and bulky guide books. Many tour guide applications have been developed over the years, most of which are recommender systems that only recommend points of interest to the tourist without guiding them to such places, until now, no system has been developed to address the issue of navigation for tourist and also bearing the in mind the limitations of storage capacity in mobile phones. Hence this thesis aims to develop an android GPS-based mobile application that combines both a cloud-based platform and multimedia features for tourists, to enable them get valuable information on POIs and serve as a guide for tourists/visitors to find the desired places in Lefkosa, Cyprus with ease. A cloud-based platform was used for this development, so as to avoid a heavy and bulky application that can take most of the users' mobile phone storage and also because it combines the advantages of both mobile computing and cloud computing, thereby providing optimal services for mobile users. With the help of the cloud system tourists can reach this data from anywhere in the world.

Keywords: Android, global positioning system (GPS), mobile cloud systems, mobile application, tourist

ÖZET

Küresel Konumlama Sistemi (GPS) dört veya daha çok GPS uyduları sayesinde zaman, yer ve yükseklik bilgisi vermektedir. Navigations için kullanılan bir uzay tabanlı uydu sistemidir. İnsan uzay-zaman davranışları GPS sistemleri sayesinde çok ilerlemiş ve gerçek zaman izleme de dahil olmak üzere birçok sistemlerde GPS kullanılmaktadır. Turizm Kıbrıs'ta ekonomik açıdan son derece önemli bir konu olduğu için bu alanda turistlere yardımcı olmak amacıyla GPS kullanımı da önemli bir konudur. Halen turistler için çeşitli yardımcı kitaplar ve broşürler bulunmakatdır. Fakat bunların hiçbiri bir turiste navigasyonda yardımcı olmamaktadır. Bu tezin amacı Android cep telefonu tabanlı ve bulut kullanan multimedya içerikli turistlerin kullanabileceği bir uygulama geliştirmektir. Bu uygulama sayesinde turistler hem genel navigasyonu kullanabilecek ve hem de Lefkoşa'nın tarihi ve turistik yerlerine kolaylıkla GPS sayesinde gidip görebileceklerdir. Bu tezde bulut tabanlı bir sistem geliştirilmiştir. Bu şekilde bütün turistik yerler hakkındaki bilgiler sınırlı belleği olan cep telefonu yerine bulutta depolanmakta ve bu optimum sistem tasarımı sayesinde dünyanın herhangi bir yerinden bu bilgilere kolaylıkla erişmek mümkün olabilmektedir.

Anahtar Kelimeler: Android, küresel konumlama sistemi (GPS), mobil bulut sistemleri, mobil uygulama, turist

ACKNOWLEDGMENT

This thesis would not have been possible without the efforts, support and patience of my supervisor, Assoc.Prof.Dr. Nadire Çavuş, May God bless you ma.

I would also like to sincerely thank Prof.Dr. Dogan Ibrahim who has been very helpful through the duration of my thesis.

Not forgetting Near East University for providing me the opportunity to do this thesis. I also wish to thank The Graduate School of Applied Sciences.

I wish to express my sincere and humble thanks to God Almighty, for the sheer tenacity of purpose and dogged determination he has bestowed on me, to my parents Mr. and Mrs. K. I. Ambair for their immense love and support and also to my husband Lt. G.P. Henry for his patience and love.

Lastly, I wish to thank all my friends and those who contributed one way or the other in the fruition of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

3G: Third Generation

4G Fourth Generation

AAA Authentication, Authorization, Accounting

API Application Programing Interface

BTS Base Transceiver Station

CC Cloud Computing

CSP Cloud Service Provider

GPS Global Positioning System

HA Home Agent

IaaS Infrastructure as a Service

LBS Location Base Service

LTE Long Term Evolution

MC Mobile Computing

MCC Mobile Cloud Computing

NIST National Institute of Standards and Technology

PaaS Platform as a Service

PDA Personal Digital Assistant

PSiS Personalized Sightseeing Tours Recommendation Systems

POI Places of Interest

RTP Real-time Transport Protocol

S3 Simple Storage Service

SaaS Software as a Service

CHAPTER 1

INTRODUCTION

There has been extraordinary increase in the collection and use of high resolution geographic data in movement of objects and people due to the various technological developments like the development of online maps and the earth images at no cost, the creation of remote sensing and easy to use, affordable, small and accessible tracking technologies. Researches in human space-time behavior has led to the increase in the utilization of GPS and other tracking devices (Bluetooth, WI-FI, etc.) in different fields like medicine (especially in cases of Alzheimer's disease),environmental health, cardiology, physiology and tourism (Goodchild, 2007; Richardson et al., 2013; Shoval et al., 2014). In recent times researchers have greatly profited from the development and spread of smartphones with tracking capabilities which transfers retrieved locations easily and at affordable rates and also has abilities to examine the large space-time database generated by these tracking devices (Shoval et al., 2014).

Tourism is of great relevance to the global economy and the use of mobile phones has been on the rise in this sector in the recent years due to their mobility, portability and availability which has also led them to become the prevailing platform for ubiquitous computing (Rodriguez-Sanchez et al., 2013). The mobile phone has improved in terms of battery life and in its abilities to support complex tourists' applications that controls information about way finding functionalities (Rodriguez-Sanchez et al., 2013). These functionalities are to be adapted to the environment at great levels of heterogeneity (physical connections, network topology, and user preferences) which is to be regularly re-estimated (Lee et al., 2009).

Tourism activities have a particular geographical and temporal pattern. Several research lay emphasis on the visitors behavior and supply-demand balance as it regards how they affect the physical environment (Hall & Page, 2006). In order to carry out analysis of trends, geography and perspectives for academic purpose or otherwise, tourism studies still needs quantitative datasets to observe flows in the tourism sector irrespective of the various detailed qualitative approaches available. Surveying methods in tourism and geography studies are making head ways due to the recent development trends in Information and Communication Technology (ICT) like the digital databases and geographical Information Systems (GIS) which have so many applications in the tourism sector (Cheng, 2001; Buhalis & Licata, 2002; Frihida et al., 2004; Lew & McKercher, 2006; Wang & Ahas et al, 2008). Mobile phones location-based services (LBS) and positioning datasets are some of the major subject areas emerging in geographical studies (Spinney, 2003; Ahas & Mark, 2005; Ratti et al., 2006; Ahas et al., 2007). The study of tourism geography has the potential to address mobile positioning data applications in space-time behavior studies (Ahas et al., 2008).

In this thesis, we will present the design and implementation of a mobile application called GPS-Based Mobile Application for Tourists, with which mobile users can get valuable information on different landmarks of Lefkosa city and guide tourists/visitors to find the desired places in the city that is nearest to them with more ease anytime and anywhere.

By GPS-Based Mobile Application for Tourists, users can get detailed information about important landmarks in text, maps and pictures.

1.1 Problem Statement

- Considering the impact of tourism to the economy of Cyprus as an island, the development of a GPS-based mobile application will help to boost and foster the rate of tourist coming into the country whereby increasing the revenue coming into the economy via the tourism sector but it is obvious that there has not been any developed GPS-based mobile application for tourism in Cyprus so far.

- There is a need to develop a GPS-based mobile application that combines both cloud and multimedia features, but until now there has been no such development in the tourism sector.

1.2 Aim of Study

The aim of the study is to develop an android GPS-based mobile application that combines both a cloud-based platform and multimedia features for tourists.

1.3 Significance of Study

GPS presents a comprehensive on accurate observation of routes and trip speeds and in recent times it has also been used as an accurate tool in route choice analysis which are needed to provide and describe alternative routes and in collecting travel data. A toolkit is needed to design such features in a one stop manner because the choice sets can get very huge. Hence, this study of the development of an android GPS-based mobile application which combines both a cloud-based platform and multimedia features for tourists will help facilitate all variables in a one stop manner.

1.4 Limitations of Study

This study is limited to the Android platform at the moment, and also Lefkosa city. It also is limited by other possible functionalities that could make the app much efficient, of these are, weather conditions of the intending Point of Interest (POI), and options of writing a review, a search button and holidays of a particular locality.

1.5 Overview of Thesis

The thesis is divided into six (6) chapters and is summarized as follows:

Chapter one discusses the aim, importance and limitations of the study.

Chapter two gives an insight on some Related Research that has been done on the development of GPS-based mobile application and on the reviews on mobile recommender systems.

Chapter three looks at the materials and methods used in the thesis.

Chapter four describes the functionality and requirements of the mobile application.

Chapter five displays the implementation of the developed application.

Chapter six gives concluding statements and future work plan.

The Reference section shows the literature cited in the whole thesis.

CHAPTER 2

RELATED RESEARCH

There has been a rise in the use of mobile devices in certain fields like the tourism sector in recent years, consequent to the fact that mobile phones are presently the existing platform for ubiquitous computing due to their accessibility and handy nature (Rodriguez-Sanchez et al., 2013). The recent increase in mobility of tourists in the world, has led to great interest in tourism flows. In recent times, most researchers emphasizes on balancing the needs and behavior of tourists and also the general surrounding (Hall & Page, 2006).

Several international bodies like the World Travel Organization (WTO), World Travel Tourism Council (WTTC) and Eurostat has brought out quantitative datasets and methodologies for analyzing tourism flows since the common quantitative method for analyzing the international tourism flows are inadequate and does not provide sufficient explanations as regard to the questions and concerns raised (Ahas et al., 2008). With the evolution of digital databases and geographical information systems in Information and Communication Technologies (ICT), surveying techniques in the study of tourism geography has improved, since the GIS and other important visualization techniques have relevant functions in the tourism sector (Lew & McKercher, 2006).

2.1 Related Research

The research presented below is arranged according to their publication dates in an ascending order;

Umlauft et al. (2003) developed a location-based multimedia Universal Mobile Telecommunication System (UMTS) application called “LOL@” (local location assistant) that aims at supporting tourists. The application offers maps, speaker-independent speech input and localization and routing functionality. They also offered a tour diary to stay in contact with family and friends back home.

Moiseeva and Timmermans (2010) shows how Bayesian belief networks can be used to established automatic data imputation and how land use of retail location GPS tracers.

Wilson, Bertlotto and Weakliam (2010) designed and developed a prototype of geographic information system called “MAPPER” (MAP PERsonalization) that provide only the relevant and sufficient map information to fit the preferences of user interaction and whereby addressing spatial information overload, thus by simplifying and completing tasks performed using interactive maps. In order to shape map content to exact user preferences and significantly reduces the size of vector data that are sets necessary to transmit and render maps on mobile devices.

Kabasi (2010) reported the relevance of both handheld devices and computers using tourism recommender systems. He provided some tourism recommender systems development steps by making a high tech personalized electronic tourism services both in computers and handheld devices. The author also reviews the presentation techniques and user modeling used in this system. They also reported the theories apply for the improvement of the personalization process in tourism recommendation systems, their applications and evaluation.

Mathkour (2011) proposed and developed a GPS-based mobile service locator system to help individuals in different walks of life locate addresses, emergency services and other services of interest using their mobile devices. The distance between user and the locations of the desired service was able to be determined by the proposed system and it is also flexible and extendible to easily incorporate additional mobile service providers and new services in the system.

Sugimoto (2011) extracted a desired landscape and the photographed locations by participant using a digital camera and GPS logger. The author also applied kernel density assessment to estimate photo-taking locations density.

Gavalas and Kenteris (2011) broaden the idea of recommender systems for travel utilizing filtering collaborative techniques and taking into account also the contextual information (such as the time, weather conditions, user’s current location and places the user already visited) for deriving improved recommendations in a pervasive environment. Installations Wireless sensor networks (WSN) around tourist sites to enable specific localization and also provide mobile users convenient and inexpensive means for uploading tourist information and ratings about points of interest (POI) via their mobile devices was also proposed by the authors. The idea of "context-aware rating", thereby ratings of users uploaded through WSN infrastructure are weighted higher to differentiate the users that rate POIs using the mobile guide application for tourist while on site and those away from POI using the internet was also introduced.

Papinski and Scott (2011) route choice analysis (RCA) and GIS toolkit developed was used to generates suite of over 40 variables describing route characteristics such as; travel time, speed statistics, distance, number of turns, number of stops signs/stop lights, number of intersections, and a measure of route circuits, etc. from their work. Single or several routes which can be gotten from the global positioning system data or some other means like shortcut paths are taken as input by the toolkit. The designed toolkit support route choice modeling by producing variables that have been tested in previous modeling efforts. They also demonstrated its utility by testing the hypothesis that workers choose certain routes to minimize either travel time or distance between home and work.

Savage et al. (2011) designed a location-based context aware recommendation system that generates a person's social network profile by learning user preferences. A decision making model, which considers the physical constraints learned and the current feeling of the individual was also defined by the authors. Their recommendation algorithm is based on a text classification problem and was implemented on a server which communicates with Nokia N900, and also the form of transportation detection and the user interface was implemented in the phone.

Gavalas et al. (2011) studies handle the problem personalized recommendations derivation for daily tourism routes for tourists visiting any destination. Places of interest selected by traveler that they would potentially wish to visit and desires a near-optimal route for daily visit, potential interest places are selected based on stated or implied user preferences was considered in their approach. Planning of daily customized personalized tourist routes considering time available for visiting sights on a daily basis, opening days of sights, user preferences and average visiting sights for these sights was also enabled in their method.

Baltrunas et al. (2012) reported a new method they took for modelling and assessing the relationship between item ratings and contextual factors rather than using the traditional approach of data collection. In this system, recommendations are rated with respect to real situations as participants go about their lives as normal. Regarding on how the context influences ratings of users, they simulated contextual situations to more easily capture data. In this regard, they proposed an approach whereby users are asked to judge whether a contextual factor (e.g season) influences the rating. They built a context-aware mobile recommender system (ReRex) that uses the context influences shown to be important based on the analysis of the data they obtained.

Kenteris, Gavalas and Mpitziopoulos (2012) broaden the concept of travel recommender systems that uses filtering collaborative techniques for improved recommendations derivation. Installation of Wireless sensor networks (WSN) around tourist sites for providing mobile users inexpensive means and convenient, for uploading tourist information and ratings about points of interest (POI) via their mobile devices was also proposed in their study.

Noguera et al. (2012) presented a novel mobile recommender application that combines mobile 3D GIS architecture and a hybrid recommendation engine. Their application allows tourists to benefit from features such as real-time location-sensitive recommendations and 3D map-based interface.

Wolf, Stricker and Hagenloh (2013) reported that the GPS tracking proved to be an efficient and resourceful tool to determine three performance measures for interpretive media, especially the holding, attracting and distracting powers of interpretive media. The study also reported that the GPS navigation tour performed well compared to traditional media in achieving an intermediate highest distracting power, the attracting power and the highest holding power. It was highly rated for overall experience with the medium and for facilitating fun in respect to the audio tour and visitors were more willing to provide word-of-mouth recommendation for the GPS navigation tour. The study finally reported that media inbuilt system achieved highest satisfactory ratings for discovery and learning and were most efficient at facilitating factual learning.

Emmanouilidis, Koutsiamanis and Tasidou (2013) reported that mobile guide applications increasingly employed portable devices in a mobile range. A mobile guide taxonomy considering several criteria was provided in their work. The various aspects of the mobile application space including client architectures, mobile user interfaces, and context awareness as well as offered functionalities, technological, highlighting functional architectural and implementation issues was also considered in the taxonomy.

Anacleto et al. (2013) stated from their study that tourists' recommendation systems have been growing primarily because of the most relevant systems on the field over the last few years and presents "PSiS mobile" which is a mobile recommendation and planning application designed to support a tourist during his vacation. It provides recommendations about POIs to visit based on the tourists preferences and on the user and sight context. Also, they suggested a visit planning which can be dynamically adopted on current user and sight context. They developed tool works also like a journey dairy since it records the tourists' moves and task to help them remember how the trip was like.

Rodriguez-Sanchez et al. (2013) study shows that the developed application GAT, enables users to generate way-finding application for both outdoor and indoor environment without the need for programming skills, assisted by a system of automatic generation and update of points of interest through a web form.

Yang and Hwang (2013) stated that recommender systems in mobile tourism have attracted significant attention in the past decade. They proposed facilitation of attracting recommendation task by looking at other tourists' ratings on their visited attractions. The proposed approach uses mobile peer-to-peer communications for exchanging ratings via their mobile devices. They also developed a cost effective travel recommender system “iTravel” to provide tourists with on-tour attraction recommendation. Three data exchange methods that allow users to effectively exchange their ratings toward visited attractions was also proposed. They performed simulated experiments to assess the proposed data exchange methods and a usage study was conducted to validate the usability of the proposed iTravel system.

Panahi, Woods and Thwaites (2013) reported from their research that mobile tourism is relatively a new trend in tourism and it involves mobile devices like electronic tourist guides. Their paper describes the design and development of a location-based mobile tourism application for cultural tourism in Malaysia by using a cloud-based platform. The application consists of three levels; front-end which is a location-based mobile tourism application for apple mobile devices and it offers a web service to generate extensible markup language (XML) output from the relational database to exchange data between the mobile application and servers in the cloud. Finally the backend level that is built on AWS cloud platform offers services such as Apache Web Server on Amazon relational database service (RDs) and a cloud storage that uses amazon simple storage service (S3). The application they developed is also caching images on the local storage and storing some data in SQLite database on the mobile device itself to provide some offline activities and also to decrease the number of requests to the sever in order to save some data traffic.

Shoval et al. (2014) reported from their study that GPS, smartphone and other tracking technologies for high resolution space-time data collection and implementation has created dramatic growth over the past decades. Only a small proportion of the journal articles that report findings using tracking technologies and data were published in geography journals or involve geographers as collaborators was found in a meta-analysis conducted. They also discuss various feasible reasons on this trend and see this negligence of such highly useful geographical tools by geographers as a missed opportunity. They also encouraged geographers to pay more attention to the new possibilities offered by these technologies in light of their immense potential for the advancement of geography in the future.

Gavalas et al. (2014) reported that recommender system represent an interesting and fast growing field of software systems that have found specific success in web environments. Their review also followed an organized method based on a classification scheme that include three different view angles in the examination of existing mobile tourism recommender systems, their chosen architecture, the degree of user involvement in the delivery of recommendations and the criteria taken into account for deriving recommendations. They also described mobile tourism as a privileged application for mobile recommender systems with opportunities to provide accurate and effective tourists recommendation and they also listed the following tasks commonly offered by existing mobile recommender systems such as; attraction sites (POI) recommendations, tourist services recommendations, collaborative user generated content, personalized multiple-day tour planning, social networking services for tourists, routes and tours recommendations etc.

It appears that nothing can slow the growth of tourism. Despite political instability, declining economies and the proliferation of terrorism, the lure of the overseas adventure continues to draw millions of people every year. One thing that is making tourist travel more attractive is the growing number of location-based services that use GPS and act as a traveler’s very own tour guide. A mobile tourist guide system for Android Mobile Phones that is able to provide tourism information to the mobile users conveniently and efficiently is a plus to any individual, organization or country**.** This system takes advantage of light-weighted mashup technology that can combine more than one data sources to create value-added services, while overcomes the limitations of mobile devices.

CHAPTER 3

THEORETICAL FRAMEWORK

This chapter provides the basic concepts on GPS-based mobile application and the overall mobile application development. It also provides some basic concepts on cloud computing and tourism. Finally it provides information about GPS-based mobile application for tourists.

3.1 Mobile Application Development

The information and communication technology (ICT) business and application was gradually improved owing to the development of Electronic applications like electronic commerce, electronic learning, internet culture and tourism guides, internet-gaming, electronic health and electronic sensing and data acquisition environment since the mid-1990s. With wireless technologies and mobile devices invoke, a new trend of mobile application has emerged in line with their electronic predecessors, which include mobile commerce (Ngaia & Gunasekaranb, 2007), mobile learning (Cavus & Ibrahim, 2009), mobile gaming (m-gaming) (Ballagas et al., 2007), mobile guides (Emmanouilidis, Koutsiamanis & Tasidou, 2013), mobile health (Akter, D'Ambra & Ray, 2013), etc.

These applications like the above mentioned are making waves in practice and in the market already (Brodt & Verburg, 2007). A forecast stated that the global mobile applications market were to reach $25billion by the end of 2015 (www.marketsandmarkets.com).

Mobile applications differ from desktop applications by their common usage characteristics; an example is the frequent requests from usage pattern of the common mobile application which has a short duration and service requests. It is unacceptable to have a response time exceeding a couple of seconds while initiating an application. A practical application does not get the user confused with a great number of options like applications designed for desktop does instead it is required to assist the user to concentrate on certain tasks (Salmre, 2005). The main advantage of mobile application is its ability to present 24/7 multi-connectivity (such as delivering precise information and services, etc.) coupled with its mobility functionality among others (Spaccapetra et al., 2005). The context-aware computing has mobile application positioned at its center due to the flexibility provided by user and mobile devices which is also important to non-mobile application too (Dey, 2001; Seffah & Javahery, 2004).

3.2 Global Positioning System (GPS)

According to United Nation (2010) Global Positioning System (GPS) is a space-based satellite system used for navigations that offers time information and location where there's a clear line of vision to four or higher GPS satellites in all weather conditions on or around the Earth, it also aids commercial, civil and military users all over the universe (United Nations, 2010).

The research on human space–time behavior has rapidly increased in recent times due to the availability and development of cost-effective and reliable tracking devices which resulted in the use of GPS and other tracking technologies in various fields such as environmental health (Zenk et al., 2011), tourism (Shoval & Isaacson, 2010), transportation (Boohte & Maat, 2009) and different aspects of medicine especially in subject areas like Alzheimer's disease (Miskelly, 2005; Richardson et al., 2013), cardiology (Le Faucheur et al., 2008), etc.

3.3 Tourism

Tourism is a relevant subject today, as it serves as a great source of revenue to any economy. However tourism is a seasonal event because of the difference in climate and weather around the world, and also due to the fact that the tourists' native land has traditional holidays and seasons (Higham & Hinch, 2002; Ahas et al., 2007). Furthermore, tourists require guide books and maps to find their way around in a strange land but these tools are often too bulky to carry around, the current rise in GPS-based mobile application has however made information more portable and user-friendly also direction seeking is made less stressful (Anacleto et al., 2014).

3.4 GPS-Based Mobile Application for Tourist

Mobile recommender systems have been very helpful in terms of aiding tourists in making decisions or acquiring information (Huang et al., 2004). In recent years, great attention has been attracted in both practice and research by Mobile tourism services (Gavalas et al., 2014).

Information systems have grown to imitate the services carried out by tourists agents which is to offer travel recommendations to tourists putting the tourists budget and time availability into consideration (Ricci, 2002). Mobile tourism applications provide interactive maps, audio and video content, and location-based services to aid a wonderful tourism experience (Kenteris et al., 2010). Context of guides is a recommendation technique that is often used to provide recommendations to tourists based on content. (Ricci, 2012).

Example, Personalized Sightseeing Tours Recommendation Systems (PSiS) offers recommendations to city visitors considering both their environmental contexts and personal interests/preference (Anacleto et al., 2014); "DailyTrip and Mtrip" provides recommendations according to the users travel preferences, users available time and user location (Gavalas et al., 2012; http://www.mtrip.com); "ReRex" uses criteria influencing context-aware recommendations as configured by the user to make its recommendations (Baltrunas et al., 2012); "I'm feeling LOCO" provides recommendations based on user location, users mode of transportation, user mood and preference (Savage et al., 2011); "SPETA" offers recommendation similar to the services given by a real tour guide i.e. based on the tourist's preferences, location, locations visited (Garcia-Crespo et al., 2009).

Certain approaches were proposed by Ge et al. (2010) and Kisilevich et al. (2010) that can be used in examining various routes of tourists in order to recommend travel patterns and also a client-server prototype was presented to recommend various travel contents. In the client-server, the sever part uses ratings and evaluations of tourists with similar preferences to make recommendations this is also known as collaborative filtering techniques (Gavalas & Kenteris, 2011).

3.5 Cloud Computing

To get a common understanding on cloud computing, let's start with the basics: the essential characteristics and service and deployment models. The National Institute of Standards and Technology (NIST), defined Cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This cloud model is composed of five essential characteristics, three service models, and four deployment models as the below picture depicts;

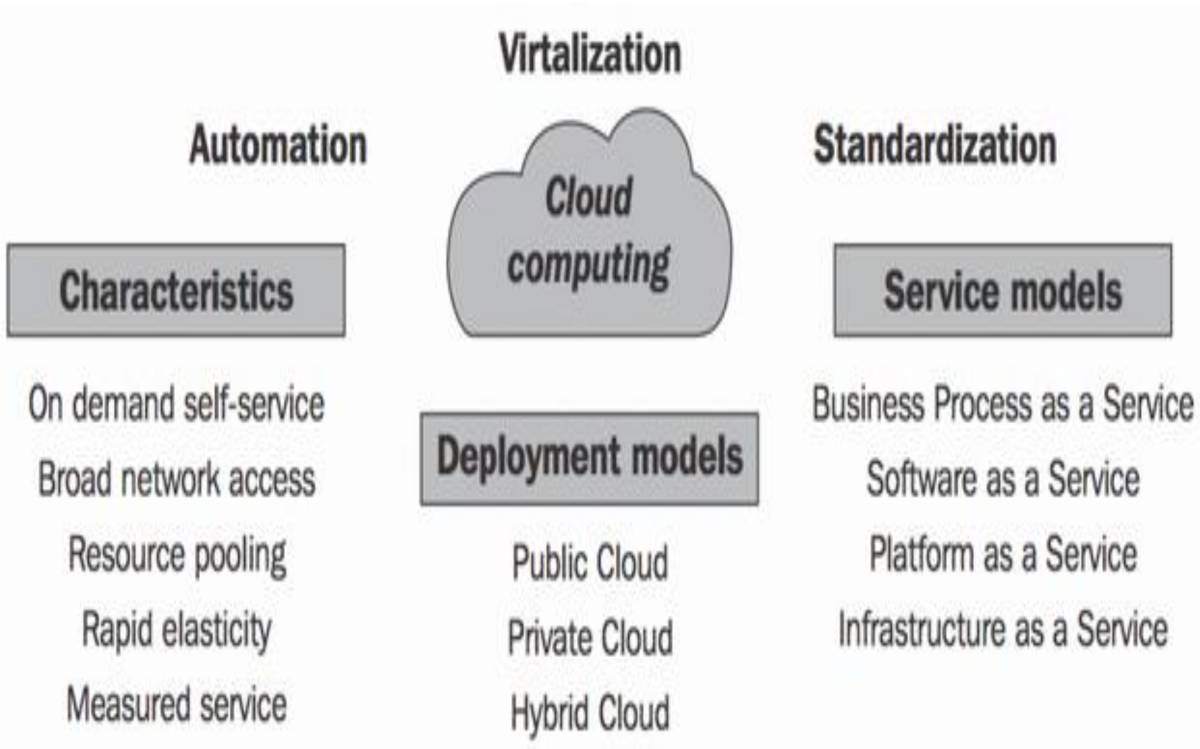

Figure 3.1: The Cloud Model (Edwin Schouten IBM SmartCloud Essential, 2014)

3.5.1 Cloud Computing Characteristics

Cloud computing is much more than just virtualization. It's really about utilizing technology "as a service". Users need little to no knowledge on the details of how a particular service is implemented, on which hardware, on how many CPU's, and so on. All that's important for a user is to have good understanding of what the service offers—and what it does not—and how to operate the self-service portal.

- On-demand self-service: Users are able to provision cloud computing resources without requiring human interaction, mostly done though a web-based self-service portal (management console).

- Broad network access: Cloud computing resources are accessible over the network, supporting heterogeneous client platforms such as mobile devices and workstations.

- Resource pooling: Service multiple customers from the same physical resources, by securely separating the resources on logical level.

- Rapid elasticity: Resources are provisioned and released on-demand and/or automated based on triggers or parameters. This will make sure your application will have exactly the capacity it needs at any point of time.

- Measured service: Resource usage are monitored, measured, and reported (billed) transparently based on utilization. In short, pay for use.

3.5.2 Cloud Computing Services

There are three service models according to the National Institute of Standards and Technology (NIST) namely:

- Infrastructure (IaaS),
- Platform (PaaS); and
- Software as-a-service (SaaS).

Interestingly enough, IBM and other major IT and analyst firms have added a fourth service model, namely Business Process as a Service (BPaaS). To get a better understanding on what each of the service models comprises, refer to the below image that depicts the layers of which atypical IT solution consists:

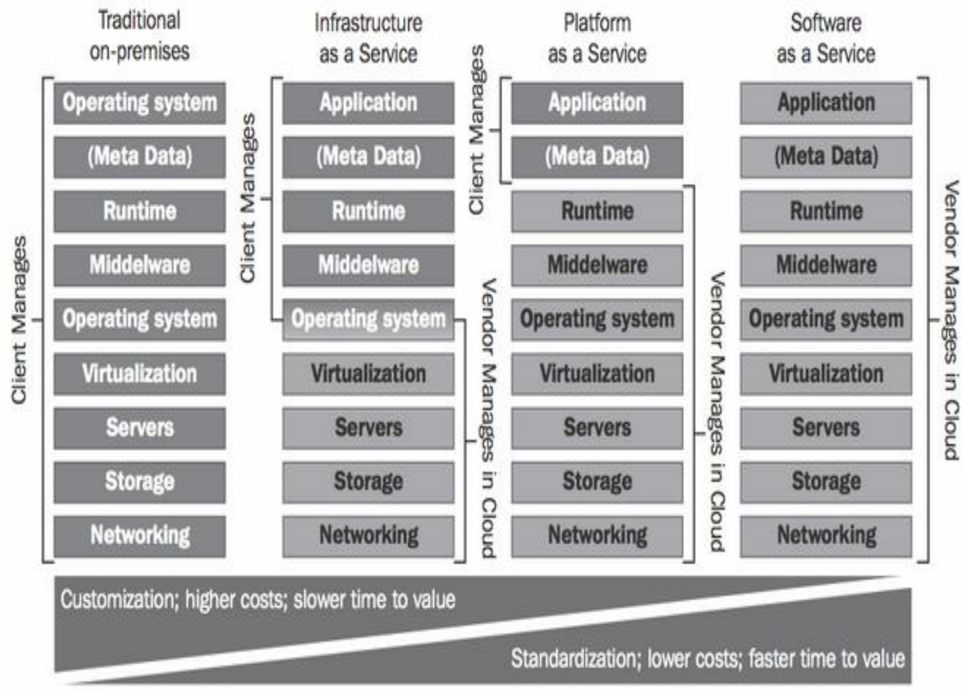

Figure 3.2: Cloud Computing Service Model (Edwin Schouten IBM SmartCloud Essentials, 2014)

3.5.3 Cloud Computing Deployment Model

A cloud deployment model represents a specific type of cloud environment, primarily distinguished by ownership, size, and access. There are four common cloud deployment models, namely;

- Public Clouds
- Community Clouds
- Private Clouds; and
- Hybrid Clouds

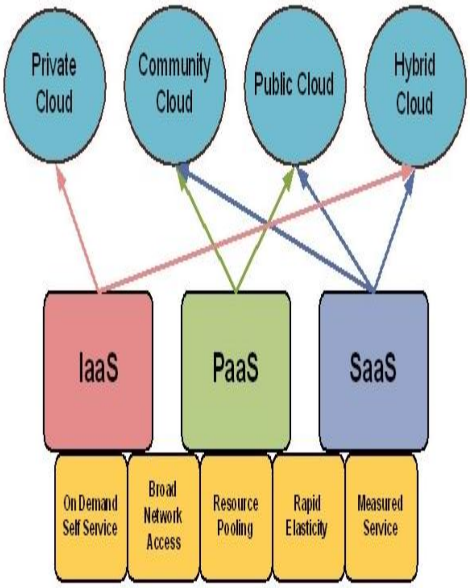

Figure 3.3: Cloud computing deployment model
(http://blog.thehigheredcio.com/2011/02/22/cloud-deployment-models/)

- **Private cloud:**

  A "private" cloud infrastructure is operated solely for a single organization or agency: the Cloud Service Provider (CSP) dedicates specific cloud services to that agency and no other clients. It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

- **Community cloud:**

  A "community" cloud infrastructure is procured jointly by several agencies or programs that share specific needs such as security, compliance, or jurisdiction considerations. The agencies or CSP may manage the community cloud and may keep it on-premises or off-premises.

- **Public cloud:**

  A "public" cloud infrastructure is available to the general public and is owned by a third party cloud service provider (CSP). In a public cloud, an agency dynamically provisions computing resources over the Internet from a CSP who shares its resources with other organizations.

- **Hybrid cloud:**

  A "hybrid" cloud comprises two or more clouds (private, community, or public) with a mix of both internally and externally hosted services that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

3.6 Mobile Application

A mobile app is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function. Mobile apps are designed with consideration for the demands and constraints of the devices and also to take advantage of any specialized capabilities they have. A gaming app, for example, might take advantage of the iPhone's accelerometer.

A mobile application also may be known as an app, Web app, online app, iPhone app or smartphone app.

3.6.1 Types of Mobile Application

Mobile apps are sometimes categorized according to whether they are web-based or native apps, which are created specifically for a given platform. A third category, hybrid apps, combines elements of both native and Web apps.

As the technologies mature, it's expected that mobile application development efforts will focus on the creation of browser-based, device-agnostic Web applications.

- **Native Apps**

  Native apps live on the device and are accessed through icons on the device home screen. Native apps are installed through an application store (such as Google Play or Apple's App Store). They are developed specifically for one platform, and can take full advantage of all the device features like camera, GPS etc. — they can also the device's notification system and can work offline.

- **Mobile Web Apps**

  Web apps are not real applications; they are really websites that, in many ways, look and feel like native applications, but are not implemented as such. They are run by a browser and typically written in HTML5.

- **Hybrid apps**

  Hybrid apps are part native apps, part web apps. (Because of that, many people incorrectly call them "web apps"). Like native apps, they live in an app store and can take advantage of the many device features available. Like web apps, they rely on HTML being rendered in a browser, with the caveat that the browser is embedded within the app.

3.7 Mobile Cloud Computing (MCC)

MCC can be defined as a combination of mobile web and cloud computing, which is the most popular tool for mobile users to access applications and services on the Internet. (Christensen, 2009).

Briefly, MCC provides mobile users with the data processing and storage services in clouds. The mobile devices do not need a powerful configuration (e.g., CPU speed and memory capacity) since all the complicated computing modules can be processed in the clouds.

Figure 3.4: Three types of Mobile Application (Javapraskash, 2013)

3.7.1 Mobile Cloud Computing Architecture and Advantages of Mobile Cloud Computing

From the concept of MCC, the general architecture of MCC can be shown in the Figure below.

In the diagram above, mobile devices are connected to the mobile networks via base stations (e.g., base transceiver station (BTS), access point, or satellite) that establish and control the connections (air links) and functional interfaces between the networks and mobile devices.

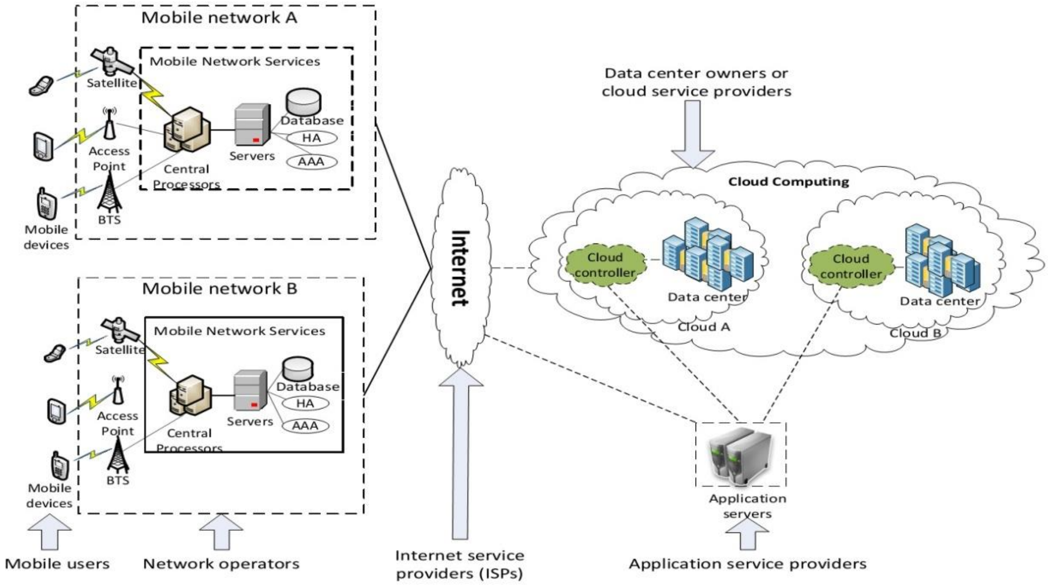

Figure 3.5: Mobile Computing Architecture (Javaprakash, 2013)

Mobile users' requests and information (e.g. ID and location) are transmitted to the central processors that are connected to servers providing mobile network services. Here, mobile network operators can provide services to mobile users as AAA (for authentication, authorization, and accounting) based on the home agent (HA) and subscribers' data stored in databases. After that, the subscribers' requests are delivered to a cloud through the Internet. In the cloud, cloud controllers process the requests to provide mobile users with the corresponding cloud services. These services are developed with the concepts of utility computing, virtualization, and service-oriented architecture (e.g., web, application, and database servers) (Hoang et al., 2013).

The details of cloud architecture could be different in different contexts. For example, a four-layer architecture is explained in "Cloud Computing and Grid Computing 360-Degree Compared," (Foster et al., 2009), to compare cloud computing with grid computing. Alternatively, a service- oriented architecture, called Aneka, is introduced to enable developers to build .NET applications with the supports of application programming interfaces (APIs) and multiple programming models presents an architecture for creating market-oriented clouds, and proposes an architecture for web- delivered business services.

Advantages of Mobile Cloud Computing

Cloud computing is known to be a promising solution for mobile computing due to many reasons (e.g., mobility, communication, and portability). In the following, we describe how the cloud can be used to overcome obstacles in mobile computing, thereby pointing out advantages of MCC.

- **Extending battery lifetime:** Battery is one of the main concerns for mobile devices. Several solutions have been proposed to enhance the CPU performance and to manage the disk and screen in an intelligent manner, to reduce power consumption. However, these solutions require changes in the structure of mobile devices, or they require a new hardware that results in an increase of cost and may not be feasible for all mobile devices.

  Computation offloading technique is proposed with the objective to migrate the large computations and complex processing from resource-limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds). This avoids taking a long application execution time on mobile devices which results in large amount of power consumption.
- **Improving data storage capacity and processing power:** Storage capacity is also a constraint for mobile devices. MCC is developed to enable mobile users to store/access the large data on the cloud through wireless networks. A good example is the Amazon Simple Storage Service (Amazon S3) which supports file storage service.
- **Improving reliability:** Storing data or running applications on clouds is an effective way to improve the reliability since the data and application are stored and backed up on a number of computers. This reduces the chance of data and application lost on the mobile devices.

  In addition, MCC can be designed as a comprehensive data security model for both service providers and users. For example,

  The cloud can be used to protect copyrighted digital contents (e.g., video, clip, and music) from being abused and unauthorized distribution.

The cloud can remotely provide to mobile users with security services such as virus scanning, malicious code detection, and authentication.

Such cloud-based security services can make efficient use of the collected record from different users to improve the effectiveness of the services.

- **Dynamic provisioning:** Dynamic on-demand provisioning of resources on a fine-grained, self-service basis is a flexible way for service providers and mobile users to run their applications without advanced reservation of resources.
- **Scalability:** The deployment of mobile applications can be performed and scaled to meet the unpredictable user demands due to flexible resource provisioning. Service providers can easily add and expand an application and service without or with little constraint on the resource usage.
- **Multi-tenancy:** Service providers (e.g., network operator and data center owner) can share the re- sources and costs to support a variety of applications and large number of users.
- **Ease of Integration:** Multiple services from different service providers can be integrated easily through the cloud and the Internet to meet the users' demands.

3.8 Mobile (Android) GPS-Based Application for Tourists

Android is an open source mobile operating system based on Linux with java support. It comes under free and open source software licenses. As per first quarter Report of the year 2012, 400 million people are using Android based devices worldwide and 59% of smart phone market is occupied by android based smart phones (AndroidOperatingSystem,http://en.wikipedia.org/wiki/Android_(operating_system)). Android provides the support of mobile map and GPS localization. Android based mobile tour-guide application can provide valuable information on different landmarks of a city or geographical location and guide tourists/visitors to find the desired places in the city/geographical location with more ease. In this thesis we are proposing a tour guide application called GPS-Based Application for Tourists on android based mobile platform Lefkosa city.

3.8.1 Mobile GPS-Based Application for Tourists system challenges

a) ***Power consumption:*** GPS consumes a lot of battery power on a mobile device. It was verified that in a continuous use it can consume all the battery power, of a current top range smart phone, in less than 7hours (Gaonkar et al., 2008; Kjasrgaard, 2012).

b) ***Network availability:*** GPS-based mobile application requires a mobile network to aid its functionality and to assist the mobile device in making position estimate determinations (Mathkour, 2011). Some connections are too slow they cannot support a GPS-based application and some interesting tourist sites might not have mobile network.

c) ***Usability:*** Most applications in the market are not user friendly and does not provide precise data, nor allow multiple ways to access the data, such as map access, application loading and real time feed back to the requester (Anacleto et al., 2014).

3.8.2 Mobile GPS-Based Application for Tourists system advantages

a) ***To the Economy***: Tourism generally brings money to every country but it depends on the publicity, accessibility and affordability of tour trip and information (Ahas et al., 2008). With the aid of GPS-based mobile application the tourists tends to save more as there will be little or no need to employ the services of a tour guide, not to talk of the publicity a mobile application creates for tourists. Generally GPS-based mobile application attracts the tourists due to the affordability and publicity it gives and the more the tourists the more the revenue for the country

b) ***To Tourists***: GPS-based mobile application provides important travel information like accommodation, emergency centers, Point of Interest (POI), etc., to tourists, which can also be given according to user specifications as most systems requests for the user preferences (Horozov, Narasimhan & Vasudevan, 2006; Ricci & Nguyen, 2007). It is also a good tool that offers alternative routes and tour recommendations to the nearest Point Of Interests (POIs) for tourists.

The chapter explains how a GPS based tour guide application using cloud computing will provide succor in tour guidance, majorly from the fact that it is on an android platform, which is becoming one of the world's most used OS. With the help of mobile cloud computing, we now have a promising solution for mobile computing due to many reasons, ranging from mobility, communication, and portability efficiency. These are to help show tourist current location via GPS on the map and helps to find the nearest spots to current location of tourist.

CHAPTER 4

SYSTEM ANALYSIS

For developing a GPS-Based Mobile Application for Tourists, this thesis elected Java programming language, which is used for Android platform application. Eclipse is the editor environment for this project, Android SDK was also used to provide an Android emulator with requirement libraries to build Android application. Finally, this thesis dealt with some APIs. The Google Map API is one of the most important APIs in this application. In addition, it helps to show tourist current location via GPS on the map and helps to find the nearest spots to current location of tourist.

4.1 System Architecture

Proposed Mobile GPS-Based Application for Tourists application will work on smart phones on Android platform not less than version 4.1**.** It can be used by visitors/ tourists. The diagram below represents proposed architecture of the tour guide application.

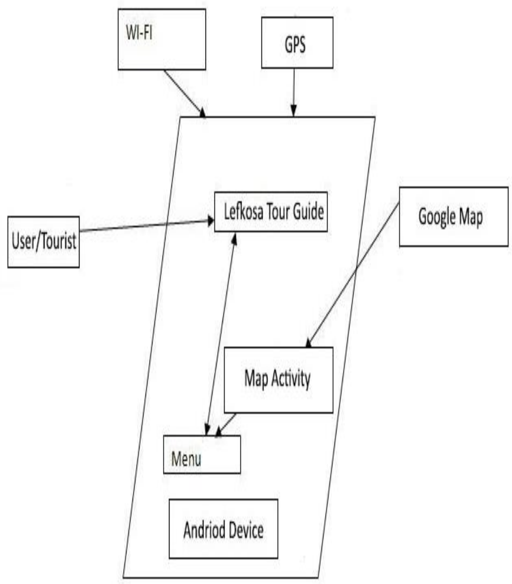

Figure 4.1: Architecture of the proposed system

4.2 System Description

The Lefkosa Tour Guide Application will be as user friendly as possible for efficiency and effectiveness. Considering the diagram above, users/tourists will interact with Lefkosa Tour guide application which can be activated from menu bar of android phone. It will use the GPS feature and Google Map functionality of the smart phone. It also has directory for points of interest, and picture and video snippet for the tourist. This tour guide application has practical significance.

This application include basic functionality such as showing the present location of the user on the map and display the route from users current location to a location chosen by the user. Tourists/Users are able to search and view the all-important landmarks based on categories e.g. Hotel, Police Station, Restaurants, etc. as shown in Figure4.2;

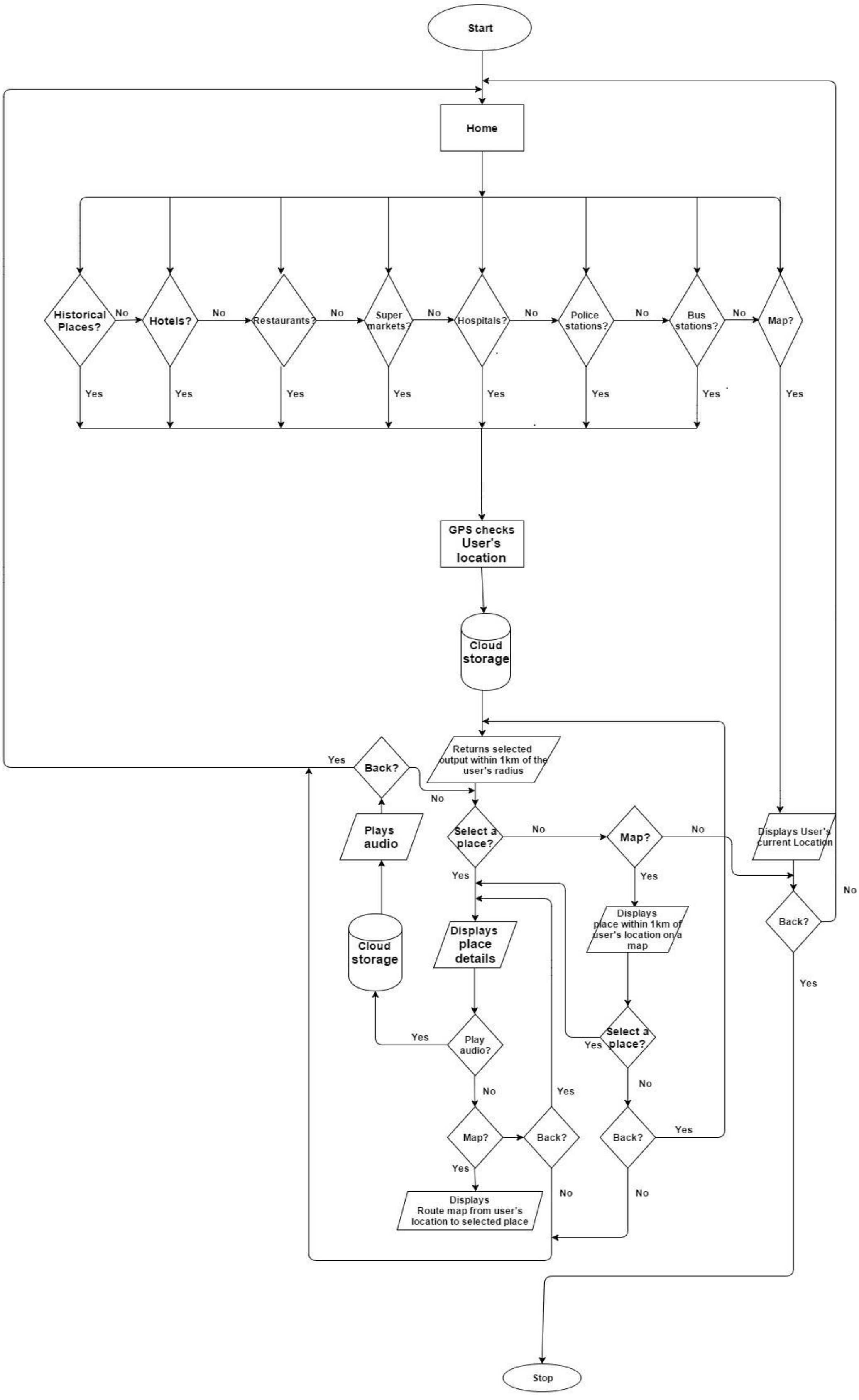

Figure 4.2: Flowchart

The significant feature of this application is the implementation of Mobile Cloud Computing (MCC), which provides mobile users with the data processing and storage services in clouds. The mobile devices do not need a powerful configuration (e.g., CPU speed and memory capacity) since all the complicated computing modules can be processed in the cloud.

.

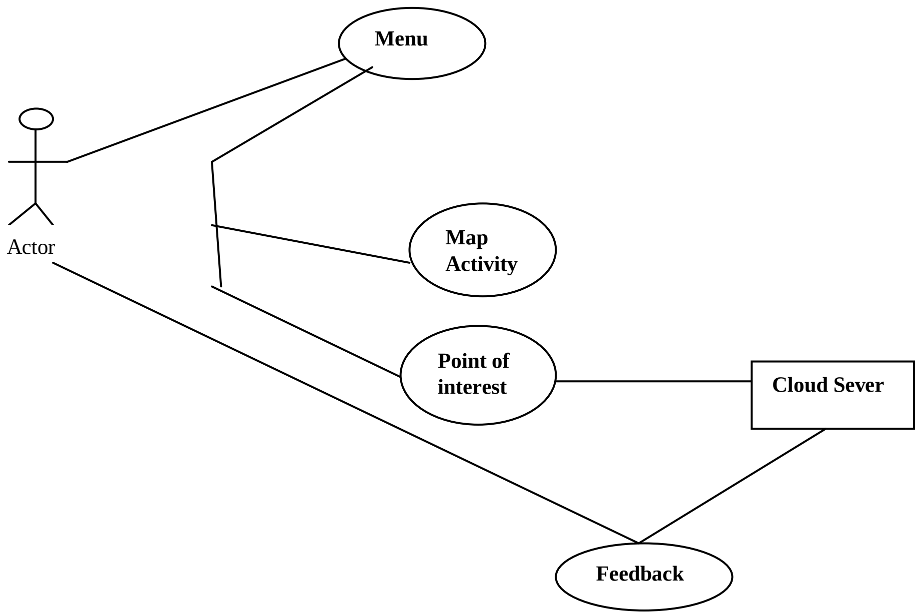

Figure 4.3: Use case diagram of the developed mobile application

In this scenario, the actor is the Tourist. The application starts at the component Menu, then the Map Activity displays a map with the current location of the user. When a user selects a certain POI, the application calculates the distance and returns only the places within 1km radius of the user’s current location in a list view. The user can select any item on the list view and upon selection it will display the details of that location with a map button which when selected can display the route from the users current location to the selected location.

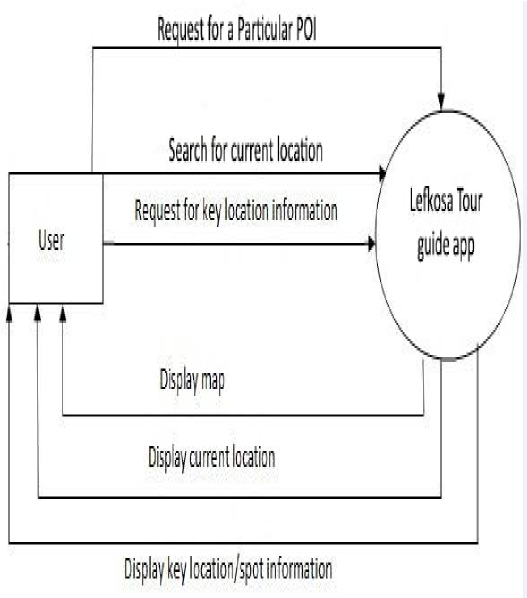

Figure 4.4: DFD Level 0 of the developed system

Here, user can request for spot information to any POI like Hospitals, Hotels, etc.

- Application displays brief description of any POI.
- User can request to display the map of any POI

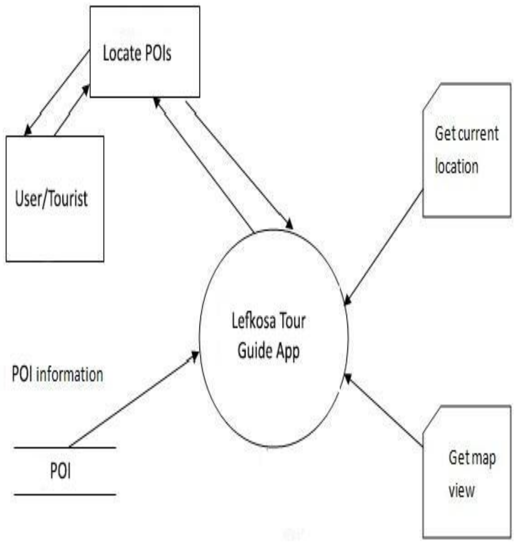

Figure 4.5: DFD Level 1 of the developed system

4.3 System Technologies

4.3.1 Cloud Computing

Cloud computing is known to be a promising solution for mobile computing due to many reasons (e.g., mobility, communication, and portability). It has been attracting the attentions of entrepreneurs as a profitable business option that reduces the development and running cost of mobile applications, of mobile users as a new technology to achieve rich experience of a variety of mobile services at low cost, and of researchers as a promising solution for green IT.

It is known that mobile devices are constrained by their processing power, battery life and storage. However, cloud computing provides an illusion of infinite computing resources. Mobile cloud computing is a new platform combining the mobile devices and cloud computing to create a new infrastructure, whereby cloud performs the heavy lifting of computing-intensive tasks and storing massive amounts of data. In this new architecture, data processing and data storage happen outside of mobile devices.

Certain factors foster the adoption of mobile cloud computing, these include;

- **Trends and Demands:** Users expect the convenience of using companies' websites or application from anywhere and at any time, and mobile devices can provide this convenience. Enterprise users require always-on access to business applications and collaborative services so that they can increase their productivity from anywhere, even when they are on the commute.
- **Improved and Increased Broadband Coverage:** 3G and 4G along with WiFi, femto-cells, fixed wireless and so on are providing better connectivity for mobile devices.
- **Enabling Technologies:** HTML5, CSS3, hypervisor for mobile devices, cloudlets and Web 4.0 will drive adoption of mobile cloud computing4.4.2 Mobile Application Technology.

A set of mobile development technologies were used to aid the development of the mobile application in this thesis. The following are the important tools used in the development of the system;

4.3.2.1 Java Development Kit (JDK)

This is a program development environment used for the development of application and applets using the Java programming language. It consists of the Java Runtime Environment (JRE) which has the Java Virtual Machine (JVM) and APIs as its components. This application is written in Java code, therefore making it necessary for the JDK to be installed on the System on which the application is to be developed. The JDK compatible with windows has been downloaded and installed.

4.3.2.2 Android Development Tool (ADT) Bundle

The ADT comprises of bundle of tools which is used for the development of applications. To develop applications, developers need to download and install a Software Development Kit (SDK), which comes in different versions depending on the phone. The ADT is a plugin for Eclipse Integrated Development Environment (IDE) which enables developers set up new android projects, debug the applications using the SDK, provides User Interface (UI) for the application, uses the Android

Framework API to add packages, and export .apk files in order to distribute the application.

In this thesis, the following were used for the development of the application;

- Eclipse IDE
- Android SDK
- Java programing language

4.3.2.3 Database

This system used cloud database for data storage so as to conquer low memory challenge in mobile devices. The system used the following cloud database for storage;

- **Parse.com:** Parse.com which is cloud based and provides a free (for low usage)/paid (for high volume usage) platform for application developers, they also provide their own APIs, and in that way NO backend programming/SQL programming is required for general database operations. It is very convenient and easy to maintain/update the database at the Parse.com website – everything can be done on the UI, no programming is needed. By calling the Parse.com APIs on Android applications, the APIs handle the network connections and database queries. The API also handles the distance filtering in the developed system. Since the database query is over the network, the callback method is called when the data is returned from the server. Therefore one can see from the callback methods that the Application will update the UI again when data is returned from the server.
- **Dropbox:** This is also a cloud database which the developed system uses for storage of data.

4.4 Mobile Operating System

A mobile operating system, also called a mobile OS, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smartphones, PDAs, tablet computers and other handheld devices. The mobile operating system is the software platform on top of which other programs, called application programs, can run on mobile devices.

4.4.1 Android

Android is now over six years old and despite its seeming popularity, there are still those who don't know what it is or what it's all about.

Android is the name of a mobile operating system (OS) based on the Linux kernel and currently developed by an American company Google. It most commonly comes installed on a variety of smartphones and tablets from a host of manufacturers offering users access to Google's own services like Search, YouTube, Maps, Gmail and more.

4.4.2 Android advantages

Google's Android Operating System in Mobile phones are still relatively new, however, it has been progressing quite rapidly. This is majorly due to certain features it can boast of that gives it an edge over other OS.

The main advantage of Android is that it is an Open Source Operating System so almost anyone can create apps for it. The number of apps grows faster, other advantages are as below, but not limited to

- *Multitasking*: Android phones can run many applications, at the same time. This means I can be surfing the net, while listening to music and recording too.
- *Ease of Notification*: Any SMS, Email, or even the latest articles from an RSS Reader, will always send a notification on the Home Screen Android phone, some devices have LED for notifications.

- *Ease of access*: Easy access to hundreds of thousands of applications via the Google Android App Market.
- *Phone options are diverse*: Talk Android phone, it will feel ‘different’ than the IOS, if the IOS is limited to the iPhone from Apple, then Android is available on mobile phones from various manufacturers, from Sony Ericsson, Motorola, HTC to Samsung
- *Can install a modified ROM*: if one is not satisfied with the standard view of their Android device, they can always change things around, as there are many custom ROM that can be used in mobile Android phones.

4.4.3 Programming Language

Mobile applications, depending on the platform of the mobile device can be written using different programming language. The developed system was written in Java Programming language

4.5 Application Features

This system was developed in order to aid tourists locate Points of Interests nearest to their location, displays details about each place so as to help the tourist make decision on places to visit, as well as map out the routes for them to get to the chosen destination.

Below are the features of the developed system:

- **Home Screen:** The system has a home screen which displays eight (8) buttons, namely; Hotels, Historical Places, Restaurants, Super Markets, Hospitals, Police Stations and Bus Stations, Map.
- **Buttons:** The buttons appear on the home screen at start up, once a button (Map button not inclusive) is clicked it checks the cloud for corresponding locations within 1km of the users’ radius and returns the locations within that parameter in form of a listview.
- **ListView:** Items within 1km of the users’ location are returned in a listview, and any item on the listview upon selection returns place details to the user.

- **Place Details:** The Place details displays details about a selected location, details like pictures, video, brief history/address and in some cases audio which gives audio version of the brief history of the place.
- **Map button:** The map button on the home screen displays a normal map pointing the users' current location. The map buttons that appear below the listview page that comes up upon the selection of a button displays a map with pins on the locations of the places within 1km of the users' radius which was displayed in form of a listview. The Map button on the place details page, displays the route from the users current location to the selected place.

4.6 User Interface

A user friendly application design is challenging but also very important. A user friendly Interface aids the application to gain popularity since users will not find the application difficult or strenuous to use. Bearing this in mind a developer needs to choose a design strategy to include in the mobile application which should depend on Attention, Relevance, Confidence and Satisfaction (Seraj & Wong, 2012).

4.6.1 Design Argument

According to Grasso and Roselli (2005), when developing an a mobile application, the Users and the Usability of the application should be considered.

User: A User of a mobile application is the person who uses the application or whom the application was meant for, in this case the User is the Tourist. The User is a very important factor to consider when developing an application. A proper evaluation of the user should be carried out before an application is developed, so as to meet users' attitudes and expectations.

Usability: Usability defines the quality of user screen design and interaction of an application. The Usability is an important factor to consider in developing an application since it reduces the time and cost of training, users' satisfaction, and improves quality of screen interaction (Seraj & Wong, 2012).

The Interface of a mobile application has to be designed without complexity and in a way it won't require high processing capacity, i.e. it should be designed in a simple way. To provide an acceptable usability level for a mobile application a set of principles has to be followed.

4.6.2 Principles of User Screen Design

Principles of User Screen Design are explained as follows (Motiwalla, 2007):

- There should be easy and consistent navigation throughout the pages of the application
- Application should be user friendly
- Similar actions need to be located at similar position

Generally, this chapter explains the mobile application that has been developed as well as its features. It also described the technologies, programming language and mobile platform used.

CHAPTER 5

SYSTEM IMPLEMENTATION

The developed mobile application is a GPS-Based Mobile Application for Tourist, it is meant to aid tourists locate the nearest POIs, view details about the POIs, and get directions to the POIs. This chapter describes the functionality of the developed system, with screenshots of the application to further describe how the system functions.

5.1 The Developed System

The developed application is installed on Samsung Galaxy S3 and the Figures that will be shown later are taken from it.

When the application starts, it launches SplashScreenActivity.java the purpose of this activity is to ask the device to get an updated location. The splash screen is shown in Figure 5.1, it shows for 3 seconds because we want to give some time for the Application to get the user's rough current location.

Figure 5.1: The start page (Splash Screen)

5.1.1 The Home Screen

The home screen comes up immediately after the splash screen. The home screen consists of eight (8) buttons as shown in Figure 11. Seven (7) of the buttons when selected returns a list of nearby locations (within 1km radius), each item on the list returned, can display place details of the location when selected and also there exist a Map button beneath each list returned, when selected it displays the locations on a map.

Figure 5.2: The home screen

The last button on the home screen is a Map button which displays the users' current location as shown in Figure 5.3.

Figure 5.3: Displaying users' current location

5.1.2 The Historical Site

When the user selects the Historical Site button, the application checks for Historical Sites within 1km radius of the user and returns a list of the places to the user as shown in Figure 5.4.

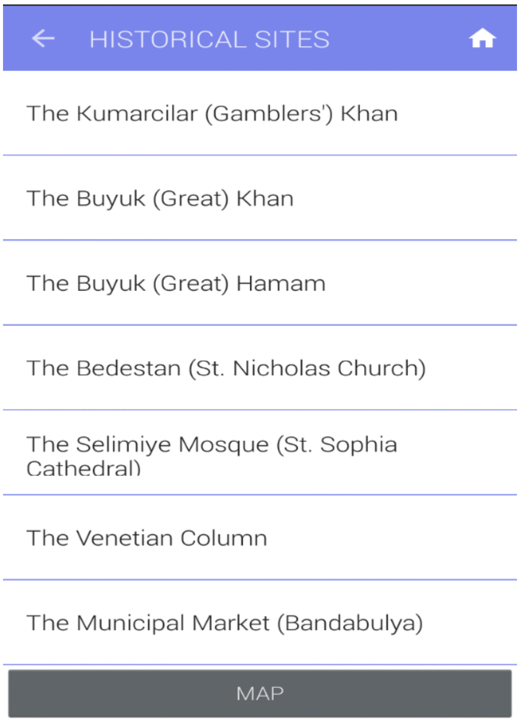

Figure 5.4: Historical sites

At the bottom of the list is a map button which when selected displays on a map all the locations that appear on the list, as shown in Figure 5.5.

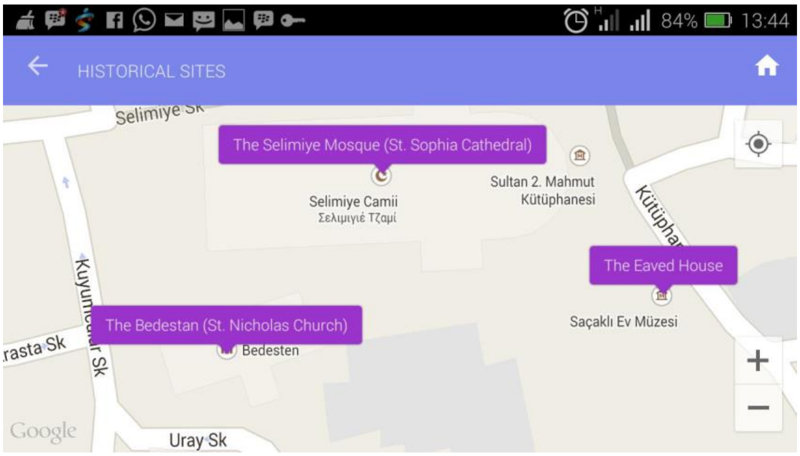

Figure 5.5: Displaying the nearest historical sites on a map

When an item is selected from the list of historical places returned by the application, it displays the place details of the selected item, which include; pictures, videos, brief history, and audio, as shown in Figure 5.6.

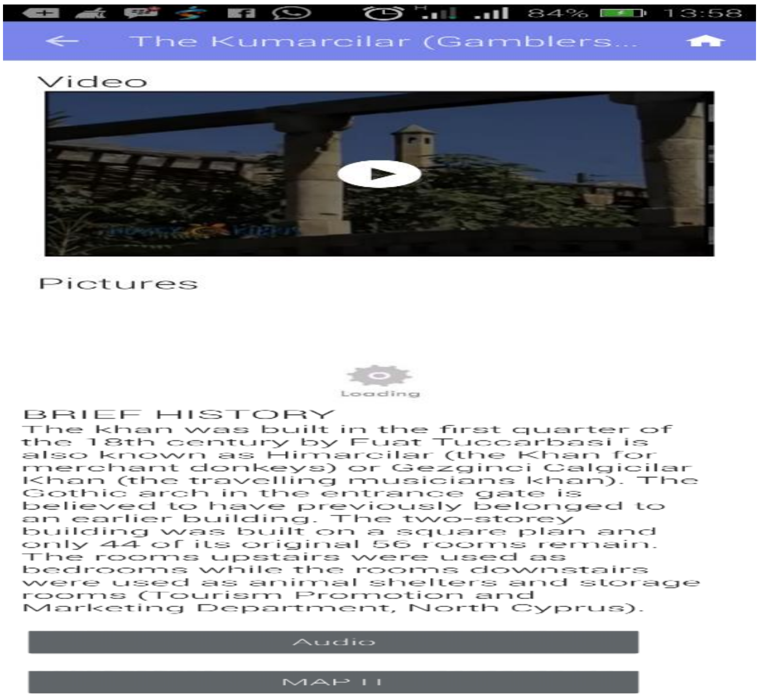

Figure 5.6: Sample of place details from historical sites

5.1.3 Hotels

When the user selects the Hotels button, the application checks for Hotels within 1km radius of the user and returns a list of the places to the user as shown in Figure 5.7.

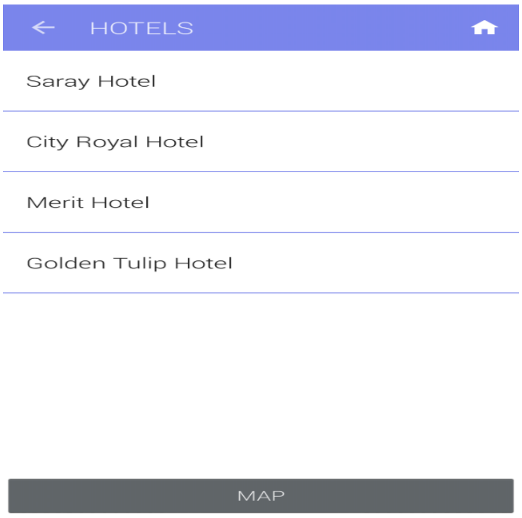

Figure 5.7: Hotels

At the bottom of the list is a map button which when selected displays on a map all the locations that appear on the list, as shown in Figure 5.8.

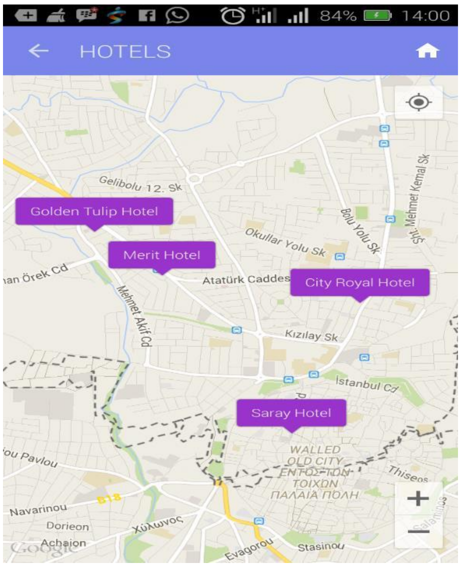

Figure 4: Map: Displaying the nearest hotels on a map

When an item is selected from the list of hotels returned by the application, it displays the place details of the selected item, which include; pictures, videos, address, as shown in Figure 5.9.

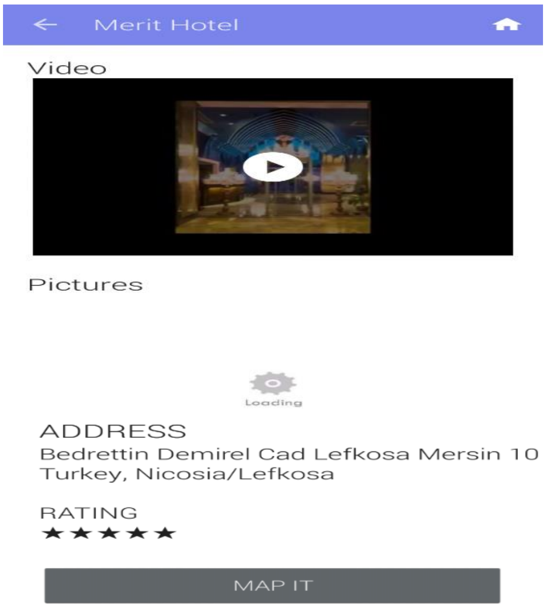

Figure 5.9: Sample of place details from hotels

5.1.4 Restaurants

When the user selects the Restaurants button, the application checks for Restaurants within 1km radius of the user and returns a list of the places to the user as shown in Figure 5.10.

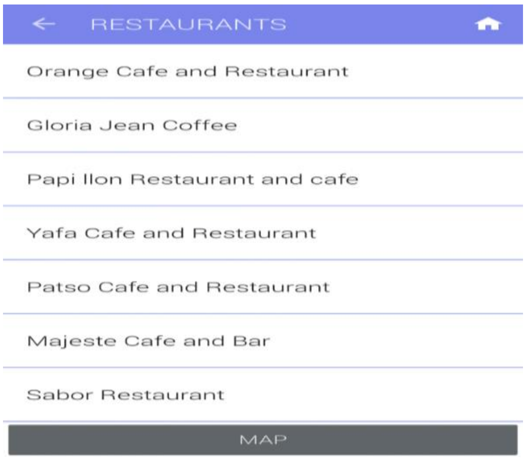

Figure 5.10: Restaurants

At the bottom of the list is a map button which when selected displays on a map all the locations that appear on the list, as shown in Figure 5.11.

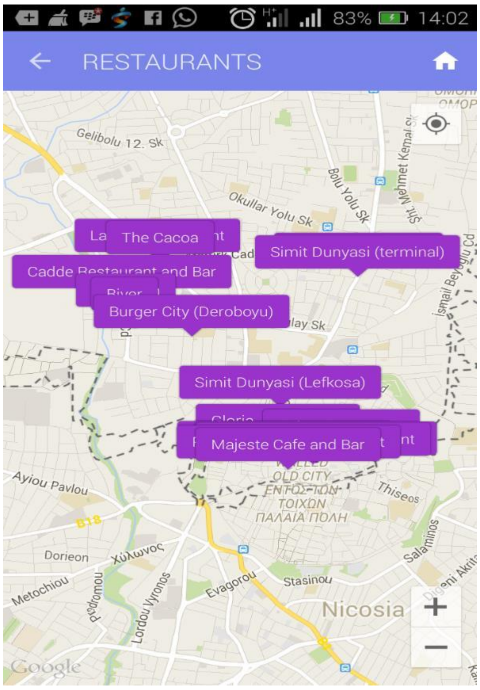

Figure 5.11: Displaying the nearest restaurants on a map

When an item is selected from the list of restaurants returned by the application, it displays the place details of the selected item, which include; pictures, videos, address, as shown in Figure 5.12.

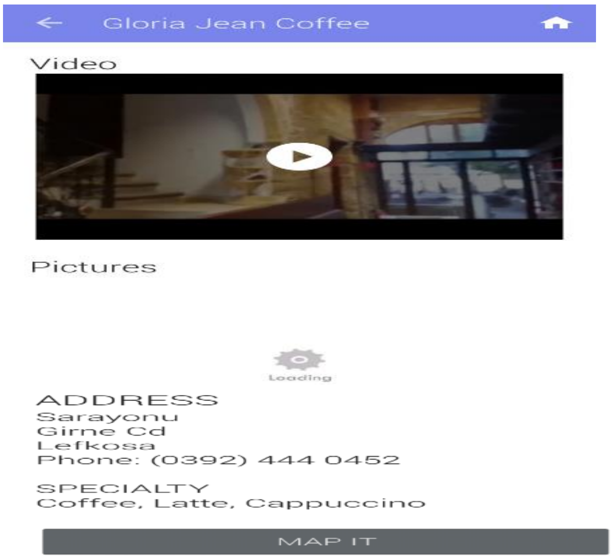

Figure 5.15: Sample of Place details from restaurants

5.1.5 Supermarkets

When the user selects the Supermarkets button, the application checks for Supermarkets within 1km radius of the user and returns a list of the places to the user as shown in Figure 5.13.

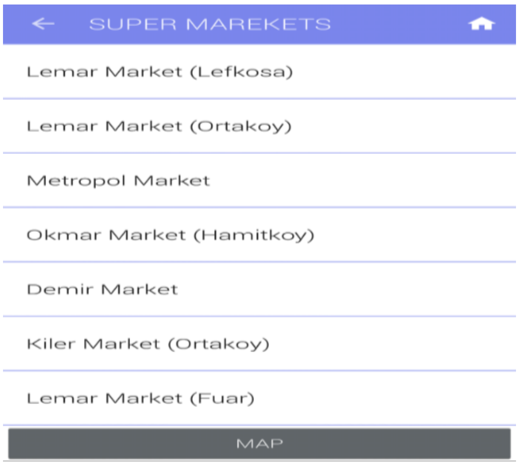

Figure 5.13: Supermarkets

At the bottom of the list is a map button which when selected displays on a map all the locations that appear on the list, as shown in Figure 5.14.

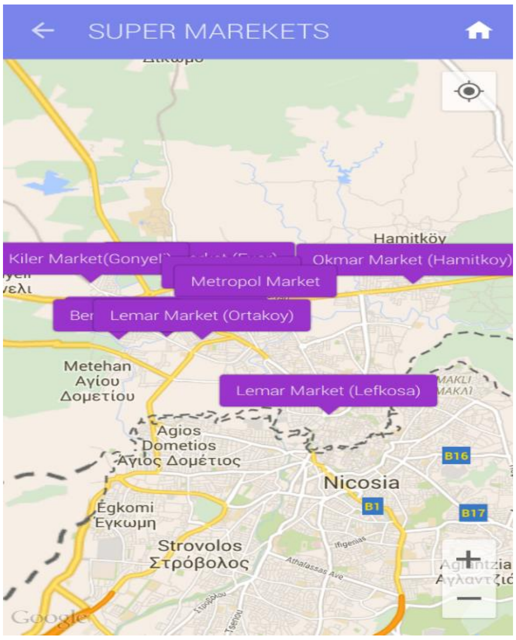

Figure 5.14: Displaying the nearest supermarkets on a map

When an item is selected from the list of supermarkets returned by the application, it displays the place details of the selected item, which include; pictures, videos, address, as shown in Figure 5.15.

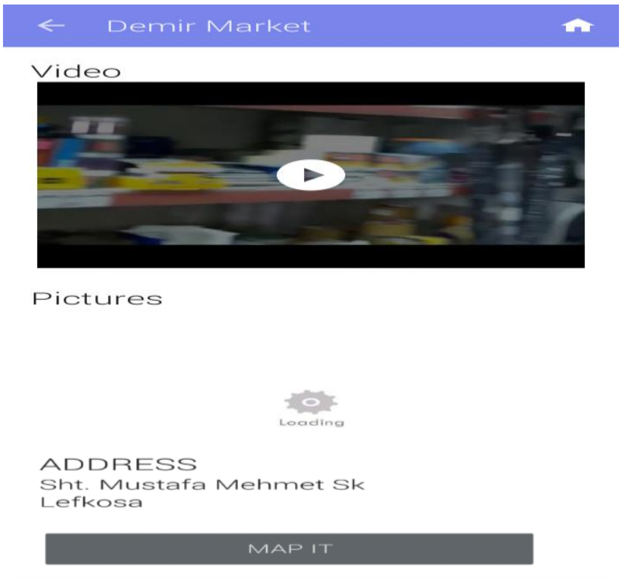

Figure 5.15: Sample of place details from supermarkets

5.1.6 Hospitals

When the user selects the Hospitals button, the application checks for Hospitals within 1km radius of the user and returns a list of the places to the user as shown in Figure 5.16.

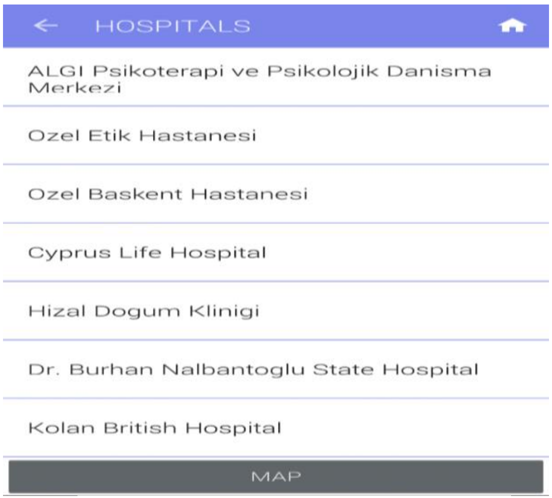

Figure 5.16: Hospitals

At the bottom of the list is a map button which when selected displays on a map all the locations that appear on the list, as shown in Figure 5.17.

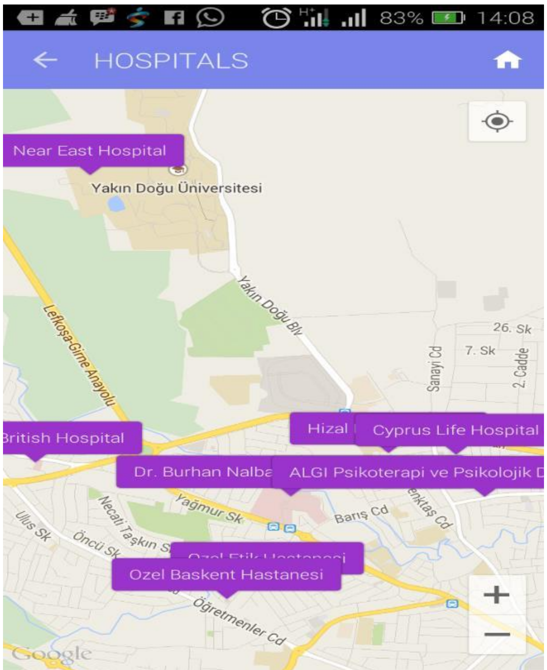

Figure 5.17: Displaying the nearest hospitals on a map

When an item is selected from the list of hospitals returned by the application, it displays the place details of the selected item, which include; pictures, videos, address, as shown in Figure 5.18.

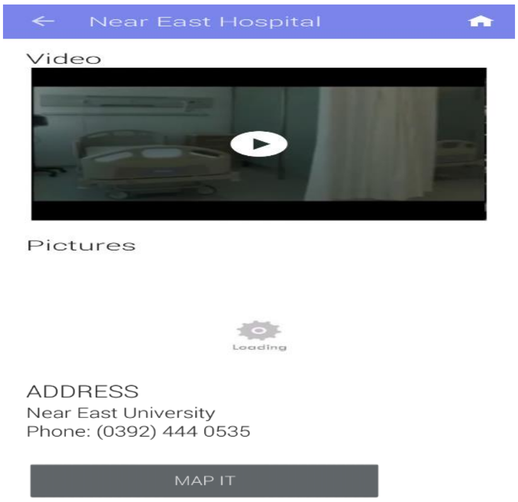

Figure 5.18: Sample of place details from hospitals

5.1.7 Police Stations

When the user selects the Police Stations button, the application checks for Police Stations within 1km radius of the user and returns a list of the places to the user as shown in Figure 5.19.

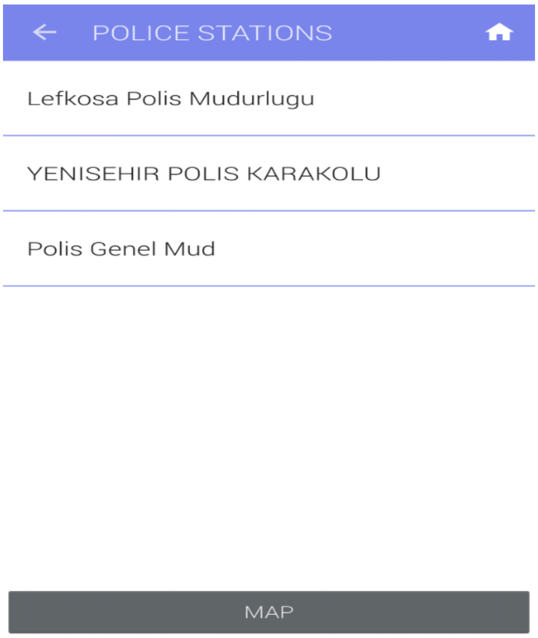

Figure 5.19: Police stations

At the bottom of the list is a map button which when selected displays on a map all the locations that appear on the list, as shown in Figure 5.20.

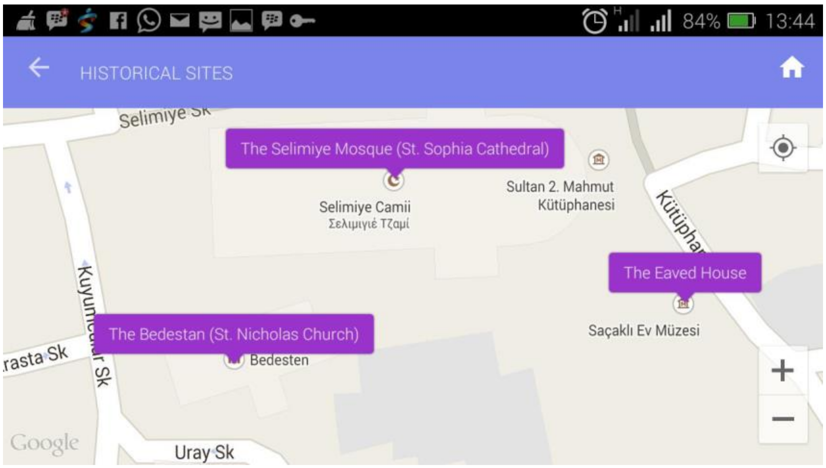

Figure 5.20: Displaying the nearest police stations on a map

When an item is selected from the list of Police Stations returned by the application, it displays the place details of the selected item, which include; pictures, videos, address, as shown in Figure 5.21.

z

Figure 5.21: Sample of place details from police stations

5.1.8 Bus Stations

When the user selects the Bus Stations button, the application checks for Bus Stations within 1km radius of the user and returns a list of the places to the user as shown in Figure 5.22.

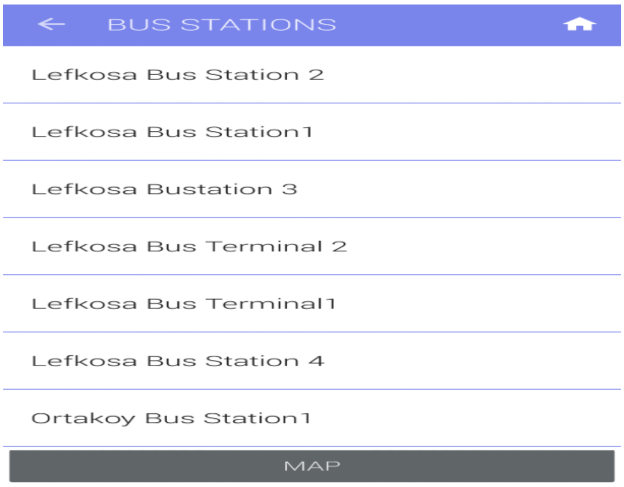

Figure 5.22: Bus stations

At the bottom of the list is a map button which when selected displays on a map all the locations that appear on the list, as shown in Figure 5.23.

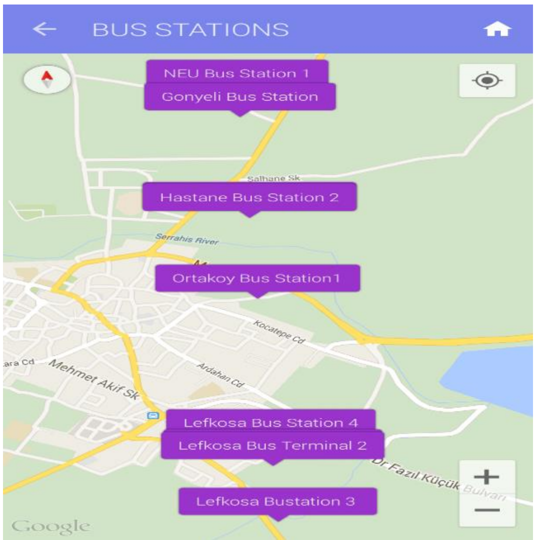

Figure 5.23: Nearest bus stations on a map

When an item is selected from the list of Bus Stations returned by the application, it displays the place details of the selected item, which include; pictures, videos, address, as shown in Figure 5.24.

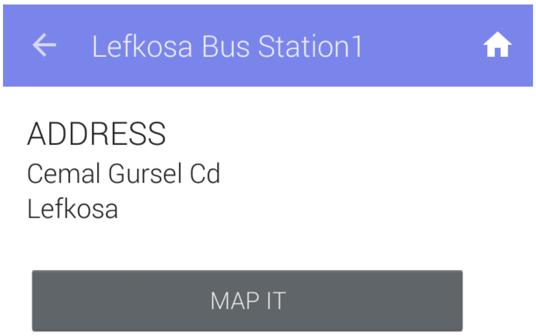

Figure 5.24: Sample of place details from bus stations

In this chapter, the snapshots of all the activities along with the features provided to the user, were displayed. This starts from the launching of the app on the mobile device, which in turn ushers us to the splash screen, and later the home screen.

The home screen consist of eight (8), Seven (7) of which when selected returns a list of nearby locations (within 1km radius). Other activities involved are mapping of a nearby location of choice, from one’s current location.

Brief details of POIs are made available. These include a short video of a location, pictures, contact number and addresses.

The application was tested with various phone models and sizes, and results were obtained as expected.

CHAPTER 6

CONCLUSION & RECOMMENDATIONS

6.1 Conclusion

In this thesis, a GPS-Based Mobile Application for Tourists called “Lefkosa Tour Guide”, was designed and implemented. This mobile application enable users get valuable information on POIs and serve as a guide for tourists/visitors to find the desired places in Lefkosa with more ease anytime and anywhere. In particular, this application can provide users location-based information of Lefkosa (few locations albeit for now) which can be accessed through a map. Mobile cloud computing is one of mobile technology trends in the future since it combines the advantages of both mobile computing and cloud computing, thereby providing optimal services for mobile users. In the recent past different types of Android app developers have emerged in the industry. In fact, many good reasons can be connected. Android is an open source platform and the developer can gain an edge over competitors. The developer community is constantly working in different applications and feed in some of the latest technical and sophisticated advancements. This ensures that the Android platform is less vulnerable to failure, as well as free from bug.

6.2 Recommendations

The developed system is an android based application, therefore there is a need to develop the same application for other platforms, such as IOS, windows, etc. There is also the need for a survey so tourists’ opinions can be taken and the application can be further improved. Also there is a need to take feedback from the application, therefore provision should be made for a review.

REFERENCES

Ahas, R., & Mark, U. (2005). Location based services-New challenges for planning and public administration? *Futures, 37*(6), 547-561.

Ahas, R., Aasa, A., Roose, A., Mark, U. & Silm, S. (2008). Evaluating passive mobile positioning data for tourism surveys: An Estonian case study. *Tourism Management, 29*, 469- 486.

Ahas, R., Aasa, A., Silm, S. & Tiru, M. (2007). Mobile positioning data in tourism studies and monitoring: Case study in Tartu, Estonia. In M. Sigala, L. Mich, & J. Murphy (Eds.), *Springer computer science: Information and communication technologies in tourism* (pp.119–128). New York, NY, USA: Springer.

Anacleto, R., Figoueiredo, L., Almedia, A. & Novais, P. (2014). Mobile application to provide personalized sightseeing tours. *Journal of Network and Computer Applications, 41*, 56-64.

Ballagas, R., Kratz, S., Yu, E., Borchers, J., Walz, S.P., Fuhr. C., Tann, M. & Hovestadt, L. (2007). REXplorer: a mobile, pervasive spell-casting game for tourists. *In Proceedings of the 2007 Conference on Human Factors in Computing Systems (CHI2007)* (pp. 1929-1934). New York, NY, USA: ACM.

Baltrunas, L., Ludwig, B., Peer, S. & Ricci, F. (2012). *Context relevance assessment and exploitation in mobile recommender systems. Pers Ubiquit Comput.* doi: 10.1007/s00779-011-0417-x.

Boohte, W. & Maat, K. (2009). Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: a large-scale application in The Netherlands. *Transportation Research Part C, 17*(3), 285-297.

Brodt, T. L. & Verburg, R.M. (2007). Managing mobile work-insights from European practice. *New Technology, Work and Employment, 22*, 52-65.

Buhalis, D. & Licata, M. C. (2002). The future eTourism intermediaries. *Tourism Management, 23*(3), 207–220.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. & Brandic, I. (2012). Cloud computing and emerging IT platforms: Vision, hype, and reality Requirements for Developing a Mobile Learning Prototype. *In the Proceedings of*

*International Conference on Computer & Information Science (2012)*2, 1014-1019, Kuala Lumpur, Malaysia. USA: IEEE.

Cavus, N. & Ibrahim, D. (2009). M-Learning: An Experiment in Using SMS to Support Learning New English Language Words. *British Journal of Educational Technology, 40*, 78-91.

Christensen J. H. (2009). Using RESTful web-services and cloud computing to create next generation mobile applications. *In the Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, pp. 627-634.

Dey A. (2001). Understanding and using context. *Personal and Ubiquitous Computing, 5*, 4-7.

Emmanouilidis, C., Koutsiamanis, R. A. & Tasidou, A. (2013). Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications, 36*, 103-125.

Edwin Schouten, (2013). IBM Smartcloud Essentials, Packt Publishing.

Emmanouilidis, C., Koutsiamanis, R. A. & Tasidou, A. (2013). Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications, 36*, 103-125.

Fall, P. I. (2012). Modern vegetation, pollen and climate relationships on the Mediterranean island of Cyprus. *Review of Palaeobotany and Palynology, 185*, 79-92.

Foster, I., Zhao, Y., Raicu, I. & Lu, S. (2009). Cloud computing and grid computing 360-degree compared. *In the Proceedings of Workshop on Grid Computing Environments (GCE)*, pp. 1.

Frihida, A., Marceau, D. J., & The'riault, M. (2004). Development of a temporal extension to query travel behaviour time paths using an object-oriented GIS. *GeoInformatica, 8*(3), 211–235.

Gavalas, D. & Kenteris, M. (2011). A pervasive web-based recommendation system for mobile tourist guides. *Personal and Ubiquitous Computing, 15*(7), 759-770.

Gavalas, D., Kenteris, M., Konstantopoulos, C. & Pantziou, G., (2012). A web application for recommending personalized mobile tourist routes. Institute of Educational Technology (*IET) Software*, *6*(4), 313-322.

Gavalas, D., Konstantopoulos, C., Mastakas, K. & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, *39*, 319–333.

Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M. & Pazzani, M. (2010). An energy efficient mobile recommender system. *In the Proceedings of the International Conference on Knowledge Discovery and Data Mining* pp. 899-908. New York, NY, USA: ACM.

Goodchild, M.F., (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal 69* (4), 211–221.

Grasso, A. & Roselli, T. (2005). From E-Learning Contents to M-Learning contents. *In the Proceedings of IADIS International Conference Mobile Learning 2005*, Qawra, Malta, June 28-30, pp. 33-38.

Hall, C. M., & Page, S. (2006). The geography of tourism and recreation. Environment place and space. Routledge.

Higham, J. & Hinch, T. (2002). Tourism, sport and seasons: The challenges and potential of overcoming seasonality in the sport and tourism sectors. *Tourism Management*, *23*(2), 175-185.

Hoang T. D., Chonho L., Dusit Ni. & Ping W. (2013). Wireless Communications and Mobile Computing, *13*(18), 1587–1611.

Horozov, T., Narasimhan, N. & Vasudevan, V. (2006).Using location for personalized POI recommendations in mobile environments. *In the Proceedings of the2006 international symposium on applications and the internet* (pp.124-129). Washington, DC, USA: IEEE Computer Society.

Kabassi, K. (2010). Personalizing recommendations for tourists. *Telematics and Informatics, 27,* 51–66.

Kenteris, M., Gavalas, D. & Mpitziopoulos, A. (2012). *A Mobile Tourism Recommender System. Computers and Communications*. doi: 10.1109/ISCC.2010.5546758.

Kisilevich, S., Keim, D. & Rokach, L. (2010). Geospatial thinking. In Painho, M., Santos, M.Y.& Pundt, H. (Eds.), *A novel approach to mining travel sequences using collections of geotagged photos* (pp. 163-182). Berlin, Heidelberg: Springer.

Le Faucheur, A., Abraham, P., Jaquinandi, V., Bouyé, P., Saumet, J.L. & Noury-Desvaux, B.(2008). Measurement of walking distance and speed in patients with peripheral arterial disease. *Circulation, 19*, 897-904.

Lee, H. N., Lim, S. H. & Kim, J. H. (2009). UMONS: Ubiquitous monitoring system in smart space. *IEEE Transaction on Consumer*, *55*(3), 1056-1064 .

Lew, A., and McKercher, B. (2006). Modeling tourist movements. A local destination analysis. *Annals of Tourism Research*, *33*(2), 403–423.

Liu, L., Moulic, R. & Shea, D. (2011) "Cloud Service Portal for Mobile Device Management". *In the Proceedings of IEEE 7th International Conference on e-Business Engineering (ICEBE),* pp.474, January 2011.

Mathkour, H.I. (2011). A GPS-based Mobile Dynamic Service Locator System. *Applied Computing and Informatics, 9*, 95-106.

Miskelly, F., (2005). Electronic tracking of patients with dementia and wandering using mobile phone technology. *Age and Ageing, 34* (5), 497–499.

Moiseeva, A. & Timmermans, H. (2010). Imputing relevant information from multi-day GPS tracers for retail planning and management using data fusion and context-sensitive learning. *Journal of Retailing and Consumer Services, 17,* 189-199.

Ngaia E. W. T. & Gunasekaranb A. (2007). A review for mobile commerce research and applications. *Decision Support Systems, 43*, 3-15.

Noguera, J.M., Barranco, M. J., Segura, R. J. & Martínez L. (2012). A mobile 3D-GIS hybrid recommender system for tourism. *Information Sciences, 215*, 37-52.

Ozay, N. (2005). A comparative study of climatically responsive house design at various periods of Northern Cyprus architecture. *Building and Environment, 40,* 841-852.

Panahi, M. S., Woods, P. & Thwaites, H. (2013). Designing and Developing a Location-Based Mobile Tourism Application by Using Cloud-Based Platform.

Technology, Informatics, Management, Engineering, and Environment. doi: 10.1109/TIME-E.2013.6611982.

Papinski, D. & Scott, D.M. (2011). A GIS-based toolkit for route choice analysis. *Journal of Transport Geography, 19*, 434-442.

Prasanna V. Javaprakash (2013). Practical Approach to Mobile Application testing: Testers perspective on IOS and Android. Canada: LeanPub.

Ratti, C., Frenchman, D., & Pulselli, R. M., et al. (2006). Mobile landscapes: Using location data from cell phones for urban analysis. *Environment and Planning B, 33*(5), 727–748.

Ricci, F. (2002). Travel recommender systems. *IEEE Intelligent Systems*, *17* (6), 55-57.

Ricci, F., Rokach, L., Shapira, B. & Kantor, P. (2010). Recommender Systems Handbook. New York: Springer.

Richardson, D.R., Volkow, N.D., Kwan, M. P., Kaplan, R.M., Goodchild, M.F. & Croyle, R.T. (2013). Spatial turn in health research. *Science*, *339* (6), 1390-1392.

Richardson, D.R., Volkow, N.D., Kwan, M.-P., Kaplan, R.M., Goodchild, M.F., Croyle, R.T., (2013). Spatial turn in health research. *Science 339* (6), 1390–1392.

Rodriguez-Sanchez, M.C., Martinez-Romo, J., Borromeo, S., and Hernandez-Tamames, J.A. (2013). GAT: Platform for automatic context-aware mobile services for m-tourism. *Expert Systems with Applications 40* (2013) 4154–4163.

Savage, N.S., Baranski, M., Chavez, N.E & Hollerer, T. (2011). I'm feeling LoCo: a location based context aware recommendation system. Retrieved May 28, 2014 from http://www.cs.ucsb.edu/~holl/pubs/Savage-2011-LBS.pdf

Salmre I. (2005). Writing Mobile Code: Essential Software Engineering for Building Mobile Applications. Boston, Massachusetts: Addison Wesley Professional.

Savage, N.S., Baranski, M., Chavez, N.E & Hollerer, T. (2011). I'm feeling LoCo: a location based context aware recommendation system. Retrieved May 28, 2014 from http://www.cs.ucsb.edu/~holl/pubs/Savage-2011-LBS.pdf

Seffah A. & Javahery H. (2004). Multiple User Interfaces: Cross-platform Applications and Context-Aware Interfaces. New York, NY, USA: John Wiley and Sons Inc.

Seraj, M. and Wong, C.Y. (2012). A study of User Interface Design principles and requirements for developing a Mobile learning prototype

Shoval, N. & Isaacson, M. (2010). Tourist Mobility and Advanced Tracking Technologies. London and New York: Routledge.

Shoval, N., Kwan, M., Reinau, K. H., and Harder H. (2014): The shoemaker's son always goes barefoot: Implementations of GPS and other tracking technologies for geographic research. *Geoforum 51* (2014) 1–5.

Spaccapetra S., Al-Jadir L. & Yu S. (2005). Somebody, sometime, somewhere, something. *In the Proceedings of the 2005 workshop on Ubiquitous Data Management* pp. 6-16. Washington, DC, USA: IEEE Computer Society.

Spinney, J. E. (2003). Mobile positioning and LBS applications. *Geography, 88*, 256-265.

Sugimoto, K. (2011). Analysis of Scenic Perception and Its Spatial Tendency Using Digital Cameras, GPS loggers, and GIS. *Procedia Social and Behavioral Sciences, 21*, 43–52.

Umlauft, M., Pospischil, G., Niklfeld, G. & Michlmayr, E. (2010). Lol@, a mobile tourist guide for umts. *Information technology & tourism*, *5,* 151-164.

Vecchiola, C., Chu, X. & Buyya, R. (2009). Aneka: A Software Platform for .NET-Based Cloud Computing. *High Speed and Large Scale Scientific Computing*, *18*, 267 – 295.

United Nations. (2010). Highlights in Space. Retrieved May 28, 2014 from http://www.iafastro.org/wp-content/uploads/2014/04/2010-Highlights-in-Space.pdf

Wang, D., and Cheng, T. (2001). A spatio-temporal data model for activity based transport. *International Journal of Geographical Information Science 5*(6), 561-585.

Wilson, D., Bertolotto, M. & Weakliam, J. (2010). Personalizing map content to improve task completion efficiency. *International Journal of Geographical Information Science, 24*(5), 741-760.

Wolf, I. D., Stricker, H. K. & Hagenloh, G. (2013). Interpretive media that attract park visitors and enhance their experiences: A comparison of modern and traditional tools using GPS tracking and GIS technology. *Tourism Management Perspectives, 7,* 59–72.

Yang, W. & Hwang, S. (2013). iTravel: A recommender system in mobile peer-to-peer environment. *The Journal of Systems and Software, 86,* 12-20.

Y. Huang, H. Su, W. Sun, J. M. Zhang, C. J. Guo, M. J. Xu, B. Z. Jiang, S. X. Yang & Zhu,J. (2010). Framework for building a low-cost, scalable, and secured platform for Web-delivered business services, *IBM Journal of Research and Development,54*(6), 535-548.

Zenk, S.N., Schulz, A.J., Matthews, S.A., Odoms-Young, A., Wilbur, J., Wegrzyn, L., Gibbs, K.,Braunschweig, C. & Stokes, C. (2011). Activity space environment and dietary and physical activity behaviors: a pilot study. *Health and Place, 17* (5), 1150–1161.

APPENDICES

APPENDIX A

USER MANUAL

System Overview

Lefkosa Tourguide is an android based mobile application that is aimed at providing users' with POIs nearest to them. Due to the limitations of memory in mobile phones, the system uses cloud database for data storage.

Application Requirements

The minimum requirements for running Lefkosa Tourguide are:

1. An android based mobile phone
2. The application is compatible with Android 4.1.2
3. The mobile phone needs to have a functioning and effective internet connection
4. The GPS on the mobile phone has to be enabled

Installing the Application

A .apk file will be provided to the user to install on the mobile phone. Once the application is stored on the device and selected the phone gives the user installation options. The application is easy and ready to use once it has been installed.

APPENDIX B

SAMPLE CODES OF MAPVIEW

```
public class MapViewFragment extends Fragment implements LocationListener {

        private int mCategoryID = -1; // Category. Used for doing database queries

        // UI Elements
        private View mRootView;
        private MapView mMapView;
        private GoogleMap mMap;

        // Map related
        private LocationManager mLocationManager;
        private static final long MIN_TIME = 400;
        private static final float MIN_DISTANCE = 1000;
        private double mGivenLat = -1;
        private double mGivenLng = -1;

        // In order to know which marker we have clicked, we need a HashMap to store
        // all the marker objects, with the corresponding mID array index, so we can
        // identify the correct mID to call
        private HashMap<Marker, Integer> mMarkerMap;

        // Database
        private String[] mResults = null;
        private int[] mID = null;
        private double[] mResultsLat = null;
        private double[] mResultsLng = null;

        // Empty constructor. Required by Google.
        public MapViewFragment() {
        }

        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

                /**
                 * If the caller is from Main Menu, mCategoryID, mGivenLat, mGivenLng
                 * should all be empty. Otherwise all three values should present.
                 **/
                Bundle args = getArguments();
```

```
if (args != null) {
        mCategoryID = args.getInt(Constants.PARAM_CATEGORY, -1);
        mGivenLat = args.getDouble(Constants.PARAM_LAT, -1);
        mGivenLng = args.getDouble(Constants.PARAM_LNG, -1);
}

// Enable tool-bar back button
((ActionBarActivity) getActivity()).getSupportActionBar().setDisplayHomeAsUpEnabled(true);
((ActionBarActivity) getActivity()).getSupportActionBar().setHomeButtonEnabled(true);

// Set tool-bar title
if (mCategoryID > 0) {
        ((ActionBarActivity) getActivity()).getSupportActionBar().setTitle(getTitleRes(mCategoryID));
} else {
        ((ActionBarActivity) getActivity()).getSupportActionBar().setTitle(R.string.lefkosa);
}

mRootView = inflater.inflate(R.layout.screen_mapview, container, false);

// Gets the MapView from the XML layout and creates it
mMapView = (MapView) mRootView.findViewById(R.id.mapview_mapview);
mMapView.onCreate(savedInstanceState);

// Gets to GoogleMap from the MapView and does initialization stuff
mMap = mMapView.getMap();

// Initialize Parse API
ParseUtil.initParse(getActivity());

// For main menu
if (mGivenLat == -1 || mGivenLng == -1) {
        mLocationManager = (LocationManager) getActivity().getSystemService(Context.LOCATION_SERVICE);

        try {
                // Depends on how accurate you need, you may also use
                // LocationManager.GPS_PROVIDER and
                // LocationManager.PASSIVE_PROVIDER for faster location fix.
                mLocationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, MIN_TIME, MIN_DISTANCE, this);
        } catch (IllegalArgumentException ex) {
```

```
                // The device has no such provider. Give up request.
            }
        }

        // Needs to call MapsInitializer before doing any CameraUpdateFactory
        // calls
        MapsInitializer.initialize(this.getActivity());

        // To show the Home button
        setHasOptionsMenu(true);

        return mRootView;
    }

    @Override
    public void onResume() {
        super.onResume();

        mMapView.onResume();
        initMap();

        if (mGivenLat != -1 && mGivenLng != -1) {
            // we do database lookup and draw markers on the map
            searchDB(mGivenLat, mGivenLng);
        }
    }

    @Override
    public void onDestroy() {
        if (mMapView != null) {
            mMapView.onDestroy();
        }
        super.onDestroy();
    }

    @Override
    public void onPause() {
        if (mMapView != null) {
            mMapView.onPause();
```

```
        }
        super.onPause();
}

@Override
public void onLowMemory() {
        if (mMapView != null) {
                mMapView.onLowMemory();
        }
        super.onLowMemory();
}

/**
 * Action bar home button - adding
 */
@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        // Inflate the menu; this adds items to the action bar if it is present.
        inflater.inflate(R.menu.menu_home, menu);
}

/**
 * Action bar home button - handling
 */
@Override
public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.action_home) {
                if (getActivity().getSupportFragmentManager().getBackStackEntryCount() > 0) {
                        getActivity().getSupportFragmentManager().popBackStack("MainMenuFragment", 0);
                        return true;
                }
        }
        return super.onOptionsItemSelected(item);
}

/**
 * Map view configurations
```

```
 */
private void initMap() {
        // You may want to pass a different provider in as the first arg here
        // depending on the location accuracy that you desire
        // see LocationManager.getBestProvider()
        // Criteria locationCriteria = new Criteria();
        // locationCriteria.setAccuracy(Criteria.ACCURACY_HIGH);

        UiSettings settings = mMap.getUiSettings();

        settings.setAllGesturesEnabled(true);
        settings.setCompassEnabled(true);
        settings.setIndoorLevelPickerEnabled(true);
        settings.setZoomControlsEnabled(true);
        settings.setMyLocationButtonEnabled(true);
        settings.setScrollGesturesEnabled(true);
        settings.setTiltGesturesEnabled(true);
        settings.setRotateGesturesEnabled(true);
        settings.setZoomGesturesEnabled(true);

        mMap.setMyLocationEnabled(true);
        mMap.setBuildingsEnabled(true);
        mMap.setIndoorEnabled(true);
        mMap.setTrafficEnabled(false);

        // Called from a category list screen, we need to move to the correct
        // place
        // in order to show the list correctly.
        if (mGivenLat != -1 && mGivenLng != -1) {
                LatLng latLng = new LatLng(mGivenLat, mGivenLng);
                CameraUpdate cameraUpdate = CameraUpdateFactory.newLatLngZoom(latLng, 14);
                mMap.animateCamera(cameraUpdate);
        }

        mMap.setOnMarkerClickListener(mOnMarkerClickListener);
}

// When a marker is clicked, we go to the PlaceDetailFragment and show the
// place details
```

```
private OnMarkerClickListener mOnMarkerClickListener = new OnMarkerClickListener() {

        @Override
        public boolean onMarkerClick(Marker marker) {
                // the hashmap will return the mID
                onMarkerClicked(mMarkerMap.get(marker));
                return false;
        }
};

private final void onMarkerClicked(int arrIndex) {
        PlaceDetailFragment fragment = new PlaceDetailFragment();

        Bundle args = new Bundle();

        // We can get the database primary key (places._id) from the adapter
        // by supplying the position.
        args.putInt(Constants.PARAM_PLACEID, arrIndex);
        args.putInt(Constants.PARAM_CATEGORY, mCategoryID);

        fragment.setArguments(args);

        ((ActionBarActivity) getActivity()).getSupportFragmentManager().beginTransaction().replace(R.id.fragment_main, fragment, "PlaceDetailFragment
        ((ActionBarActivity) getActivity()).getSupportFragmentManager().executePendingTransactions();
}

/**
 * android-map-utils Icon Generator helper methods
 */
private void drawMarkers() {
        mMap.clear();
        mMarkerMap = new HashMap<Marker, Integer>();

        IconGenerator iconFactory = new IconGenerator(getActivity());
        int resultsCount = mResults.length;

        iconFactory.setStyle(IconGenerator.STYLE_PURPLE);
        for (int iCount = 0; iCount < resultsCount; iCount++) {
                addIcon(iconFactory, mResults[iCount], new LatLng(mResultsLat[iCount], mResultsLng[iCount]), mID[iCount]);
```

```
        }
}

private void addIcon(IconGenerator iconFactory, String text, LatLng position, int idIndex) {
        MarkerOptions markerOptions = new MarkerOptions().icon(BitmapDescriptorFactory.fromBitmap(iconFactory.makeIcon(text))).position(position).anchor(iconFa
        mMarkerMap.put(mMap.addMarker(markerOptions), idIndex);
}

/**
 * Look up the Category string resource ID (for setText, setTitle, etc. use)
 */
private int getTitleRes(int category) {
        switch (category) {
        case Constants.CATEGORY_HISTORICAL_SITES:
                return R.string.historical_sites;

        case Constants.CATEGORY_HOSPITALS:
                return R.string.hospitals;

        case Constants.CATEGORY_HOTELS:
                return R.string.hotels;

        case Constants.CATEGORY_POLICE_STATIONS:
                return R.string.police_stations;

        case Constants.CATEGORY_RESTAURANTS:
                return R.string.restaurants;

        case Constants.CATEGORY_SUPERMARKET:
                return R.string.supermarket;

        }

        // failed to look up. this is a bug and should not happen
        // Caller will receive id = 0.
        return 0;
}

/****
```

```
 * The below methods are for "implements LocationListener"
 *
 * Because we want to move the MapView to user's current location, we need
 * to intercept the location change and move the map camera accordingly.
 */
@Override
public void onLocationChanged(Location location) {
        LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
        CameraUpdate cameraUpdate = CameraUpdateFactory.newLatLngZoom(latLng, 14);
        mMap.animateCamera(cameraUpdate);

        // As we have moved the map to the user's current location, we stop
        // listening to location changes to avoid the map keep on moving as the
        // user moves.
        mLocationManager.removeUpdates(this);
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
}

@Override
public void onProviderEnabled(String provider) {
}

@Override
public void onProviderDisabled(String provider) {
}

/****
 * The below codes are taken from ListViewAdapter, with some modifications
 * to keep also the LatLng.
 */
// Parse.com DB
public void searchDB(double lat, double lng) {

        // New Parse.com implementation
ParseGeoPoint point = new ParseGeoPoint();
point.setLatitude(lat);
```

```java
        point.setLongitude(lng);
            ParseQuery<ParseObject> query = ParseQuery.getQuery("places");
            if (mCategoryID > 0) {
                query.whereEqualTo("category_id", mCategoryID);
                query.whereWithinKilometers("latlng", point, 6500);
            }
            query.findInBackground(new FindCallback<ParseObject>() {
                public void done(List<ParseObject> object, ParseException e) {
                    if (object == null || object.size() == 0) {
                        // No places available
                        mResults = new String[0];
                        mID = new int[0];
                        mResultsLat = new double[0];
                        mResultsLng = new double[0];
                        // Nothing to do because no Markers to display
                    } else {
                        int effectiveRecordCount = 0;
                        mResults = new String[object.size()];
                        mID = new int[object.size()];
                        mResultsLat = new double[object.size()];
                        mResultsLng = new double[object.size()];
                        // Parse.com do all the distance filtering
                        // we only take all the return values
                        for (ParseObject thisObj : object) {
                            mID[effectiveRecordCount] = thisObj.getInt("Parse_id");
                            mResults[effectiveRecordCount] = thisObj.getString("name");
                            mResultsLat[effectiveRecordCount] = thisObj.getParseGeoPoint("latlng").getLatitude();
                            mResultsLng[effectiveRecordCount] = thisObj.getParseGeoPoint("latlng").getLongitude();
                            effectiveRecordCount++;
                        }
                        // Have search result, we can draw markers
                        drawMarkers();
                    }
                }

            });
        }

}
```

APPENDIX C

SAMPLE CODES OF PLACE DETAILS

```
public class PlaceDetailFragment extends Fragment {

        private int mCategoryID; // Category. Used for determining header text
                                                        // (Brief history/address)
        private int mPlaceID; // Category. Used for doing database queries

        // UI Elements
        private View mRootView;
        private TextView mTextViewHistoryTitle;
        private TextView mTextViewHistory;
        private TextView mTextViewVideo;
        private View mSegmentVideo;
        private NetworkImageView mVideoPreview;
        private Button mButtonAudio;
        private TextView mTextViewPhotos;
        private LinearLayout mSegmentPhotos;
        private Button mButtonMAPIT;

        // Media Player
        private MediaPlayer mMediaPlayer = null;

        // Volley - we use this library to help getting images from the network
        private ImageLoader mImageLoader;
        private final String YOUTUBE_PREVIEW_URL = "http://img.youtube.com/vi/%s/0.jpg";

        // Empty constructor. Required by Google.
        public PlaceDetailFragment() {
        }

        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {

                /**
                 * All fragment calls should have supplied the place ID, but to avoid
                 * force close we set default place ID = 1 in case the caller forgot to
                 * specify one
                 **/
                Bundle args = getArguments();
```

```
        mCategoryID = args.getInt(Constants.PARAM_CATEGORY,
                Constants.CATEGORY_HISTORICAL_SITES);
        mPlaceID = args.getInt(Constants.PARAM_PLACEID, 1);

        // Enable tool-bar back button
        ((ActionBarActivity) getActivity()).getSupportActionBar()
                .setDisplayHomeAsUpEnabled(true);
        ((ActionBarActivity) getActivity()).getSupportActionBar()
                .setHomeButtonEnabled(true);

        mRootView = inflater.inflate(R.layout.screen_detail, container, false);

        mTextViewVideo = (TextView) mRootView.findViewById(R.id.detail_video);
        mSegmentVideo = mRootView.findViewById(R.id.detail_videoframe);
        mVideoPreview = (NetworkImageView) mRootView
                .findViewById(R.id.detail_videopreview);

        mTextViewPhotos = (TextView) mRootView.findViewById(R.id.detail_photos);
        mSegmentPhotos = (LinearLayout) mRootView
                .findViewById(R.id.detail_photoframe);

        mTextViewHistoryTitle = (TextView) mRootView
                .findViewById(R.id.detail_history_title);
        mTextViewHistory = (TextView) mRootView
                .findViewById(R.id.detail_history);

        mButtonAudio = (Button) mRootView.findViewById(R.id.detail_audio);
        mButtonMAPIT = (Button) mRootView.findViewById(R.id.detail_map);

        // To show the Home button
        setHasOptionsMenu(true);

        ParseUtil.initParse(getActivity());

        return mRootView;
    }

    @Override
    public void onResume() {
```

```java
        super.onResume();

        // init Volley libray
        mImageLoader = VolleySingleton.getInstance(
                    getActivity().getApplicationContext()).getImageLoader();

        // Query the parse.com cloud database using the given place ID for the place information
        searchDBPlaceVideo(mPlaceID);
        searchDBPlacePhotos(mPlaceID);
        searchDBPlaceDesc(mPlaceID);
        searchDBPlaceAudio(mPlaceID);
        searchDBPlaceMAPIT(mPlaceID);
    }

    @Override
    public void onPause() {
        if (mMediaPlayer != null && mMediaPlayer.isPlaying()) {
            mMediaPlayer.stop();
        } else if (mMediaPlayer != null) {
            mMediaPlayer.release();
            mMediaPlayer = null;
        }
        super.onPause();
    }

    /**
     * Action bar home button - adding
     */
    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        // Inflate the menu; this adds items to the action bar if it is present.
        inflater.inflate(R.menu.menu_home, menu);
    }

    /**
     * Action bar home button - handling
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
```

```
        int id = item.getItemId();

        if (id == R.id.action_home) {
                if (getActivity().getSupportFragmentManager()
                                .getBackStackEntryCount() > 0) {
                        getActivity().getSupportFragmentManager().popBackStack(
                                        "MainMenuFragment", 0);
                        return true;
                }
        }
        return super.onOptionsItemSelected(item);
}

/**
 * Database
 */
private void searchDBPlaceVideo(int placeID) {

        // Hide Video First
        mTextViewVideo.setVisibility(View.GONE);
        mSegmentVideo.setVisibility(View.GONE);

        // New Parse.com implementation
        ParseQuery<ParseObject> query = ParseQuery.getQuery("videos");
        query.whereEqualTo("place_id", placeID);
        query.getFirstInBackground(new GetCallback<ParseObject>() {
                public void done(ParseObject object, ParseException e) {
                        if (object == null) {
                                // No video available for this place
                                mTextViewVideo.setVisibility(View.GONE);
                                mSegmentVideo.setVisibility(View.GONE);

                        } else {
                                final String videoURL = object.getString("url");

                                String youtubeThumbnailURL = getYouTubeThubmnail(videoURL);
                                if (youtubeThumbnailURL != null) {
                                        mVideoPreview.setImageUrl(youtubeThumbnailURL,
                                                        mImageLoader);
```

```
                                        mSegmentVideo.setOnClickListener(new OnClickListener() {

                                                @Override
                                                public void onClick(View v) {
                                                        // Play YouTube using built-in Youtube player
                                                        // startActivity(new Intent(Intent.ACTION_VIEW,
                                                        // Uri.parse(videoURL)));
                                                        Intent intent = new Intent(getActivity(),
                                                                        FullscreenPlayerActivity.class);
                                                        intent.putExtra(Constants.PARAM_UTID,
                                                                        getYouTubeID(videoURL));
                                                        startActivity(intent);
                                                }
                                        });
                                        mTextViewVideo.setVisibility(View.VISIBLE);
                                        mSegmentVideo.setVisibility(View.VISIBLE);

                                } else {
                                        mTextViewVideo.setVisibility(View.GONE);
                                        mSegmentVideo.setVisibility(View.GONE);
                                }
                        }
                }
        });
}

private void searchDBPlacePhotos(int placeID) {

        mTextViewPhotos.setVisibility(View.GONE);
        mSegmentPhotos.setVisibility(View.GONE);

        // New Parse.com implementation
        ParseQuery<ParseObject> query = ParseQuery.getQuery("photos");
        query.whereEqualTo("place_id", placeID);
        query.findInBackground(new FindCallback<ParseObject>() {
                public void done(List<ParseObject> object, ParseException e) {
                        if (object == null || object.size() == 0) {
                                // No photos available for this place
                                mTextViewPhotos.setVisibility(View.GONE);
```

```
                        } else {

                                LayoutInflater inflater = LayoutInflater
                                                .from(getActivity());
                                View photoitem;
                                NetworkImageView networkimageview;
                                String photoURL;

                                // As the number of photos per place is different,
                                // we add photo to the view one by one programmatically

                                // Clear the LinearLayout before adding new photos
                                mSegmentPhotos.removeAllViews();

                                for (ParseObject thisObj : object) {
                                        photoURL = thisObj.getString("url");

                                        photoitem = inflater.inflate(
                                                        R.layout.item_networkimageview, null);

                                        networkimageview = (NetworkImageView) photoitem
                                                        .findViewById(R.id.networkimageview);
                                        networkimageview.setImageUrl(photoURL, mImageLoader);

                                        mSegmentPhotos.addView(networkimageview);

                                }
                                ;

                                mTextViewPhotos.setVisibility(View.VISIBLE);
                                mSegmentPhotos.setVisibility(View.VISIBLE);
                        }
                }

        });
}

private void searchDBPlaceDesc(int placeID) {
```

```
        // Historical Sites and other categories have different header
        if (mCategoryID == Constants.CATEGORY_HISTORICAL_SITES) {
                mTextViewHistoryTitle.setText(R.string.brief_history);
        } else {
                mTextViewHistoryTitle.setText(R.string.address);
        }

        mTextViewHistory.setVisibility(View.GONE);

        // New Parse.com implementation
        ParseQuery<ParseObject> query = ParseQuery.getQuery("places");
        query.whereEqualTo("Parse__id", placeID);
        query.getFirstInBackground(new GetCallback<ParseObject>() {
                public void done(ParseObject object, ParseException e) {
                        if (object == null) {
                                // No description available for this place
                                mTextViewHistory.setVisibility(View.GONE);

                        } else {

                                // Set the place name as the actionBar title
                                ((ActionBarActivity) getActivity()).getSupportActionBar()
                                                .setTitle(object.getString("name"));

                                // Set brief history text - supporting basic html formatting
                                // tags
                                mTextViewHistory
                                                .setText(Html.fromHtml(object.getString("description")));

                                mTextViewHistory.setVisibility(View.VISIBLE);
                        }
                }
        });
}

private void searchDBPlaceMAPIT(int placeID) {

        mButtonAudio.setVisibility(View.GONE);
```

```
        // New Parse.com implementation
        ParseQuery<ParseObject> query = ParseQuery.getQuery("mapit");
        query.whereEqualTo("place_id", placeID);
        query.getFirstInBackground(new GetCallback<ParseObject>() {
                public void done(ParseObject object, ParseException e) {
                        if (object == null) {
                                // No map link available for this place
                                mButtonAudio.setVisibility(View.GONE);

                        } else {
                                final String mapitURL = object.getString("url");

                                mButtonMAPIT.setOnClickListener(new OnClickListener() {

                                        @Override
                                        public void onClick(View v) {

                                                WebViewFragment fragment = new WebViewFragment();
                                                Bundle args = new Bundle();
                                                args.putString(Constants.PARAM_MAPURL, mapitURL);
                                                fragment.setArguments(args);

                                                ((ActionBarActivity) getActivity())
                                                                .getSupportFragmentManager()
                                                                .beginTransaction()
                                                                .replace(R.id.fragment_main, fragment,
                                                                                "WebViewFragment")
                                                                .addToBackStack("PlaceDetailFragment")
                                                                .commit();
                                                ((ActionBarActivity) getActivity())
                                                                .getSupportFragmentManager()
                                                                .executePendingTransactions();
                                        }
                                });
                                mButtonMAPIT.setVisibility(View.VISIBLE);
                        }
                }
        });
}
```

```
private void searchDBPlaceAudio(int placeID) {

        mButtonAudio.setVisibility(View.GONE);

        // New Parse.com implementation
        ParseQuery<ParseObject> query = ParseQuery.getQuery("audios");
        query.whereEqualTo("place_id", placeID);
        query.getFirstInBackground(new GetCallback<ParseObject>() {
                public void done(ParseObject object, ParseException e) {
                        if (object == null) {
                                // No audio available for this place
                                mButtonAudio.setVisibility(View.GONE);

                        } else {
                                final String audioURL = object.getString("url");
                                mButtonAudio.setOnClickListener(new OnClickListener() {
                                        @Override
                                        public void onClick(View v) {
                                                // Play audio using external player
                                                // Intent intent = new Intent();
                                                // intent.setAction(android.content.Intent.ACTION_VIEW);
                                                // intent.setDataAndType(Uri.parse(audioURL),
                                                // "audio/*");
                                                // startActivity(intent);
                                                audioPlayer(Uri.parse(audioURL));
                                        }
                                });
                                mButtonAudio.setVisibility(View.VISIBLE);
                        }
                }
        });
}

private String getYouTubeID(String url) {
        try {
                // The youtube link must be in the format:
                // http://youtu.be/ABCAj0KzXs
                // then we extract the video ID: ABCAj0KzXs
```

```
            String[] splitUrl = url.split("/");
            return splitUrl[splitUrl.length - 1];

        } catch (NullPointerException npe) {
            return null;
        } catch (PatternSyntaxException pse) {
            return null;
        }
    }

    private String getYouTubeThubmnail(String url) {
        try {
            // The youtube link must be in the format:
            // http://youtu.be/ABCAj0KzXs
            // then we extract the video ID: ABCAj0KzXs

            String[] splitUrl = url.split("/");
            return String.format(YOUTUBE_PREVIEW_URL,
                    splitUrl[splitUrl.length - 1]);

        } catch (NullPointerException npe) {
            return null;
        } catch (PatternSyntaxException pse) {
            return null;
        }
    }

    // Audio Playback
    private void audioPlayer(Uri uri) {
        // set up MediaPlayer
        try {
            if (mMediaPlayer != null && mMediaPlayer.isPlaying()) {
                mMediaPlayer.stop();
                mMediaPlayer.release();
                mMediaPlayer = null;
                mButtonAudio.setText(R.string.audio);
                mButtonAudio.setClickable(true);
            } else {
```

```
                mButtonAudio.setClickable(false);

                // Media Player
                mMediaPlayer = new MediaPlayer();
                mMediaPlayer
                        .setOnBufferingUpdateListener(new OnBufferingUpdateListener() {

                            @Override
                            public void onBufferingUpdate(MediaPlayer mp,
                                    int percent) {
                                if (percent < 100) {
                                    mButtonAudio.setClickable(false);
                                    mButtonAudio.setText(String.format(
                                            "Buffering... (%d %%)", percent));
                                } else {
                                    if (!mMediaPlayer.isPlaying()) {
                                        mButtonAudio.setText("Stop");
                                        mMediaPlayer
                                                .setOnBufferingUpdateListener(null);
                                        mMediaPlayer.start();
                                        mButtonAudio.setClickable(true);
                                    }
                                }
                            }
                        });
                mMediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
                mMediaPlayer.setDataSource(getActivity(), uri);
                mMediaPlayer.prepareAsync();
                mButtonAudio.setText("Buffering...");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
```