DESIGNING AND EVALUATING A CONTEXT-AWARE RECOMMENDER SYSTEM FOR MOBILE DEVICES

AWS NAZAR AHMED ALAZZAWI

SEPTEMBER 2016

DESIGNING AND EVALUATING A CONTEXT-AWARE RECOMMENDER SYSTEM FOR MOBILE DEVICES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY
AWS NAZAR AHMED ALAZZAWI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING

SEPTEMBER 2016

Title of the Thesis: **Designing and Evaluating a Context-aware Recommender System for Mobile Devices.**

Submitted by **Aws Nazar Ahmed ALAZZAWI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Halil Tanyer EYYUBOĞLU
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Müslim BOZYİĞİT
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist Prof. Dr. Murat SARAN
Supervisor

Examination Date: 19.09.2016

Examining Committee Members

Assist. Prof. Dr. Murat SARAN	(Çankaya Univ.)	
Assoc. Prof. Dr. H. Hakan MARAŞ	(Çankaya Univ.)	
Assist. Prof. Dr. Gökhan ŞENGÜL	(Atılım Univ.)	

STATEMENT OF NON-PLAGIARISM PAGE

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	:	Aws ALAZZAWI
Signature	:	
Date	:	19.09.2016

ABSTRACT

DESIGNING AND EVALUATING A CONTEXT-AWARE RECOMMENDER SYSTEM FOR MOBILE DEVICES

ALAZZAWI, Aws Nazar Ahmed
M.Sc., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Murat SARAN

September 2016, 43 pages

Recommender systems have been in the field of research since the 1990s. During recent years, the importance of recommender systems have been on the rise due to the increase in the digital data usage which made it necessary to retrieve information in a timely, efficient and relevant way. Context aware technologies are beginning to be more integral and necessary due to its capability to create more personalized and user specific recommendations. In this thesis, an Android application was developed to test a context aware recommender system that uses contextual information gathered from the mobile device's location sensor to suggest places of interest near the user in order to investigate the efficiency of the technology used in modern mobile devices. The results showed adding context to recommender systems such location can add a valuable information to the recommendation process thus making such systems more relevant. The performance tests showed that such system didn't need to consume a large amount of the device's resources. However, it can have an effect on power consumption when using GPS sensors to calculate the user's location.

Keywords: Recommender system, Context-awareness, web services

ÖZ

MOBİL CİHAZLAR İÇİN BAĞLAM BİLİNÇLİ ÖNERİ DİZGESİ TASARIMI VE DEĞERLENDİRMESİ

ALAZZAWI, Aws Nazar Ahmed
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi: Yrd. Doç. Murat SARAN

Eylül 2016, 43 sayfa

Öneri dizgesi, 90'lı yıllardan beri araştırma konusu olmuştur. Son zamanlarda, öneri dizgenin önemi, yeterli ve uygun bir yolla bilgi edinimini önemli kılan sayısal veri kullanımıyla beraber artışmıştır. Bağlam bilinçli teknolojiler, daha kişisel ve kullanıcıya özel öneriler oluşturabilme kapasitesi ile daha bütünleşik ve gerekli hale gelmeye başlamıştır. Bu tezde, kullanıcının yakınındaki ilgisini çekebilecek yerleri önermek ve modern mobil cihazlarda kullanılan teknolojinin yeterliliğini araştırmak için mobil cihazların konum sensöründen elde edilmiş bağlamsal bilgileri kullanan bağlam bilinçli öneri dizgesini test etmek için bir Android uygulaması geliştirilmiştir. Elde edilen sonuçlar, öneri dizgesine bağlam eklemenin (konum vb.) öneri aşaması için önemli olacağını ve bu tip sistemlerin daha uygulanabilir olacağını göstermiştir. Performans testleri, böyle bir sistemin çok miktarda cihaz kaynağı kullanmasına gerek olmadığını göstermiştir. Bununla birlikte, sistemin, kullanıcının konumunu belirlemek için GPS sensörünü kullanırken cihazın güç tüketimi üzerinde etkisi olabileceği ortaya çıkmıştır.

Anahtar Kelimeler: Tavsiye sistemleri , Bağlam temelli yaklaşım, Web servisler

ACKNOWLEDGEMENTS

I would like to thank Assist. Prof. Dr. Murat SARAN and express full gratitude for his supervision, guidance and great help during the development of this thesis. I would like to thank my family for their support and help through the period of this thesis. I would like to thank my friends Mohammed Abdulaziz and Barbora halkova for proof readying and helping with their suggestions.

TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM iii
ABSTRACT iv
ÖZ ……. v
ACKNOWLEDGEMENTS …….. vi
TABLE OF CONTENTS …….. vii
LIST OF FIGURES ……... x
LIST OF TABLES ……..... xi
CHAPTERS:

1. INTRODUCTION..... 1
1.1. Overview 1
1.2. Objective..... 1
1.3. Organization of the Thesis..... 2
2. BACKGROUND..... 3
2.1. Recommender Systems 3
2.1.1. Content-Based 4
2.1.2. Collaborative Filtering 5
2.1.3. Knowledge-Based Filtering..... 5
2.1.4. Hybrid..... 6
2.2. Context-awareness..... 6
2.2.1. Context in Recommender Systems 6
2.2.2. Acquiring Contextual Information 7
2.2.3. Location-Aware Recommendation Systems 7
3. LITERATURE REVIEW AND RELATED WORK..... 9
3.1. Related Systems..... 11
3.1.1. AroundMe 11
3.1.2. FourSquare 12
4. METHOD AND IMPLEMENTATION 14

4.1. Recommendation Method and Motivation 14
4.2. System Overview 14
4.3. Tools and Frameworks 15
4.3.1. Android Platform 15
4.3.2. .NET Framework 16
4.3.3. Windows Communication Foundation (WCF) 17
4.3.4. RESTful WCF Service 19
4.4. Data Representation 19
4.4.1. Extensible Markup Language (XML) 19
4.4.2. JavaScript Object Notation (JSON) 20
4.4.3. XML or JSON 20
4.5. System Requirements 21
4.5.1. Client Application 21
4.5.2. Server Application 22
4.6. Design and System Architecture 22
4.6.1. Decomposition 23
4.7. System Components 23
4.7.1. Client Application 23
4.7.2. Server Application 27
5. EXPERIMENT 33
5.1. Defining Goals 33
5.1.1. Recommendation Relevancy 33
5.1.2. System Performance 33
5.2. Experiments 34
5.2.1. Recommendation relevancy 34
5.2.2. System Performance 34
6. RESULTS AND EVALUATION 36
6.1. Recommendation Relevancy 36
6.2. Analyzing the Findings 37
6.2.1. System Performance 38
6.2.2. Memory Allocation 38
6.2.3. CPU Utilization 39
6.2.4. Network Requests 39

6.2.5. Accuracy..... 40
6.2.6. Battery Consumption..... 40
7. CONCLUSIONS AND FUTURE WORK..... 43
REFERENCES R1
APPENDICES ……...
A. CURRICULUM VITAE …….. A1

LIST OF FIGURES

FIGURES

Figure 1	General Architecture of Content-based Recommender Systems	4
Figure 2	AroundMe Application Interface …….…..	11
Figure 3	Foursquare Interface …...……	12
Figure 4	System Overview …….……	15
Figure 5	Android Software Stack……..	16
Figure 6	.NET Framework Component Stack	17
Figure 7	Simple XML Example ……...……	19
Figure 8	Simple JSON Example ……...	20
Figure 9	Login and Registration Interfaces ……...	24
Figure 10	Interests Screen…….……...	24
Figure 11	Main Screen…….	25
Figure 12	Visualization of The Database Tables and Relations …….	28
Figure 13	Typical Recommendation Response from the Server in Json Format……..	32
Figure 14	The Distribution of The Ratings Given by all the Users…….	37
Figure 15	Memory Allocation…….	38
Figure 16	CPU Utilization…...……	39
Figure 17	Network Requests……...……	39
Figure 18	Energy Usage over Time……..……...	41
Figure 19	Power Consumption over Time…..…….	41

LIST OF TABLES

TABLES

Table 1	Some Differences Between JSON and XML …...……..	21
Table 2	Differences Between GPS and Network Location Provider …..	26
Table 3	Ratings Given By Each User ……..	36

CHAPTER 1

INTRODUCTION

1.1 Overview

The rapid rise of digital data as well as the massive growth of the World Wide Web and its environment has gradually made information overload a problem for users [1]. Recommender systems have shown that they can be a valuable tool in solving this problem. Recommender systems have been in the research field since the mid-1990s when the various methods and approaches have been developed [2] [3]. However, traditional recommender systems do not use contextual information such as the location of the user and the time that the suggestion was requested. Context Aware recommender systems are the type that uses contextual information as part of its algorithm and in many cases this adds useful information that can create more user-specific and personalized suggestions [4]. In this thesis the focus will be on context-aware recommender system for mobile devices. A study will be made on current technologies and an analysis as to their efficiency will also be conducted. This will be realized through an application that can enhance user experience suggesting points of interests based on the user's locations and personal interests.

1.2 Objective

The aim of this thesis is to conduct a full evaluation of a context-aware recommender system for mobile devices and inspect the usability and effectiveness of the algorithm used during the experiment under a functional and relevant recommender system created for mobile devices.

1.3 Organization of the Thesis

This thesis contains 7 chapters. In Chapter 2 a background of the topic will be presented and will cover recommender system general notion and the different approaches. Context awareness and context aware recommender system will also be covered in this chapter. Chapter 3 includes related work in the field of recommender systems and context aware recommender systems. Chapter 4 give a full details about the system and methods presented in this thesis including the system architecture and platforms. Chapter 5 introduce the experiments that was made in details. Chapter 6 demonstrates the results and evaluation. And finally, Chapter 7 contains the conclusion and future work part.

CHAPTER 2

BACKGROUND

This chapter contains an overview of recommendation methods, context awareness and context-aware recommender systems. Methods and algorithms relevant to the research in this thesis will be discussed in order to provide an insight to the research problem and the motivation of the study.

2.1 Recommender Systems

Recommender systems are based on a concept that is common and persistent in our daily lives. When put in the situation of having to choose from a couple of seemingly comparable options we generally tend to seek the help of an expert or someone with previous experience in this situation. These additional opinions are supposed to bring relevant and insightful information that can facilitate the initial dilemma. Common examples of this concept are informal recommendations within social groups or reviews on online forums and in magazines. Recommender systems predict users' preferences on a group of items and suggest which items a user might be interested in [5]. This requires the recommender system to collect information about the user's previously made preferences such as an online market that records previously purchased items and suggests products similar to the ones that the user has already bought . The recent growth of data in the digital world has led researchers to develop more effective methods and techniques [3]. Based on information about the user and the analysis of general behavior of other users, it is now possible to make more accurate predictions on what an individual user might be interested in.

The following techniques are the most commonly used in recommender systems for different purposes and needs [5].

2.1.1 Content-based

Content-based recommender systems suggest items to users using information about both users and items [6]. Each item includes a description to categorize it in the system and similarly, users' preferences are recorded in the system in form of user profiles. This information can either be collected explicitly by asking the user to enter their favorite categories of products and activities; or it can be collected implicitly by recording their user pattern or previously liked items [8]. For example, a content-based recommender system for restaurants would suggest Italian food restaurants for a user who previously liked Pizza, Pasta and Calzone. Figure 1 shows the general architecture of content-based recommender systems.

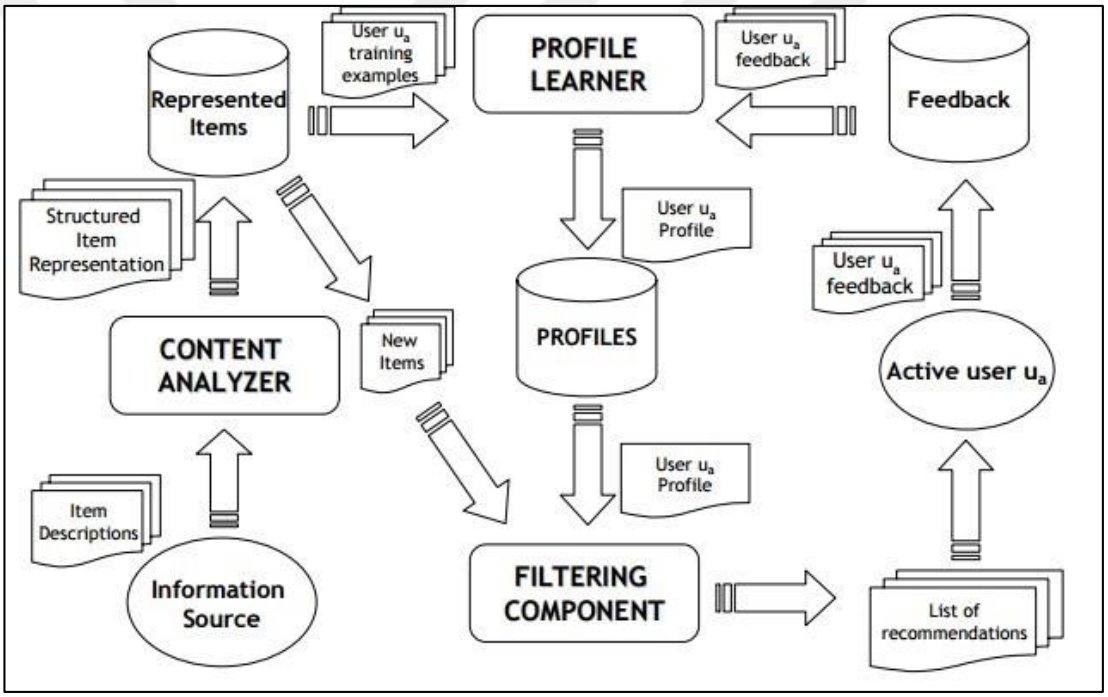

Figure 1 General Architecture of Content-based Recommender Systems

Content-based recommender system has an important advantage because it can be used without the need to wait for ratings from other users. Moreover, it also recommends new items to the user simply because of having similar properties to the items known to the user. However, the disadvantage of this approach is that each new user has to rate a sufficient amount of items for the system to understand the user's preferences, which is known as the *new user problem*. This approach also lacks the ability to

recognize items that have vague properties such as music and images that can be subject to interpretation. Furthermore, this approach requires information and feature data of items and that might be difficult to collect and store in a database [9].

2.1.2 Collaborative Filtering

Collaborative filtering recommender systems recommend items according to ratings they obtain from users with similar preferences. This is called user-based collaborative filtering [7]. When used in a music a recommender system will suggest items to a particular user using ratings and evaluations provided by other users with similar taste in music. In an item-based collaborative filtering the system compares the items the user has previously rated and compares them to all the other items in a database. Afterwards, it recommends the most similar ones with the highest average ratings. This makes collaborative filtering different from other approaches because it uses evaluation instead of analysis. In other words this means that recommendations are based on the quality of items instead of their properties since a high rating generally indicates a high-quality item [10]. However, collaborative filtering does not only face the *new user problem* but it will also face the *new item problem* since sufficient amount of ratings is needed for each item so that the system is able to recommend it to a user.

2.1.3 Knowledge-based Filtering

A knowledge-based system recommends items according to the user's needs by gathering information about the user either by asking the users about the relevant information or by recording the user's choices and preferences within certain criteria within the system. For instance, a recommendation of a smart phone will depend on whether the weight or screen size or battery life is more important to the user [11]. Knowledge coding of such kind can has different approaches such as rules in a rule-based system [12].

2.1.4 Hybrid

A hybrid recommender system uses a combination of two or more methods in order to compensate for the shortcoming of each method and provide more accurate and relevant results to users [11]. Such systems can predict recommendations separately and then combine them in the end. Furthermore, they can also use contextual information to provide more precise results.

2.2 Context-awareness

Before proceeding to contextual information and its potential for recommender systems it is essential to discuss the general meaning of context and contextual information. The concept of context has been studied in many study fields including linguistics, philosophy, cognitive science, information retrieval and computer science (mainly in artificial intelligence and ubiquitous computing) [4,14]. This has lead context to have many definitions depending on the field of study. However, the Oxford Advanced Learner's Dictionary defines context as "a situation in which something happens and that helps you to understand it" [49]. Additionally Webster's dictionary describes context as "a situation in which something happens: the group of conditions that exist where and when something happens" [50]. In computer science one of the most referenced definitions is the one by Abowd et al. where context is defined as any information that can be used to characterize the situation of an entity [13].

2.2.1 Context in Recommender Systems

Traditional recommender systems that were mentioned earlier mainly focus on users, items and their relations presented by ratings. Nevertheless, these systems assume that the recommendation is valid regardless of the situation in which it was requested. Despite the fact that in some recommender systems user's preferences and items properties might be sufficient to provide a relevant result, in other systems there are many factors that might affect user's preference. Factors such as (location, weather, time, etc.) can have a considerable impact on the way recommendation is made.

2.2.2 Acquiring Contextual Information

It is possible to acquire contextual information in the following ways [4]:

- *Explicitly*, or by directly asking users to provide information about themselves. For example, this might be done through a questionnaire, filled out by a user, providing insight of their preferences through a series of direct questions about themselves in order to get access to a certain software. Similar examples can be found on many websites and mobile applications in which users are asked to provide personal information such as age, gender, country of origin, and personal interests.
- *Implicitly*, by directly accessing sources of contextual information such as location sensors in a mobile phone. This way does not require user's interaction with the system and generally this way is more available than explicit information although it can have less quality information.
- *Inferring* by using data mining techniques and building statistical models to infer contextual information. For instance, information such as age and gender can be inferred in a movie service that is used by multiple users by monitoring user's choices.

2.2.3 Location-aware Recommendation Systems

In modern systems and especially in the mobile environment and personal devices location became an important parameter for many systems and applications including context-aware recommenders [16]. *Location-Aware Recommendation Systems (LARS)* uses spatial properties (location) to generate relevant suggestions and Recommendations since users typically prefer nearby items rather than items that are far away [15, 16]. LARS model function can be represented in terms of Items, User and Location as shown in Eq. (2.1)

$$\text{Problem: User} \times \text{Item} \times \text{Location} \rightarrow \text{Recommendations} \tag{2.1}$$

In [15] the authors proposed that there are 3 types of location-based ratings: *Spatial ratings for non-spatial items*, where it is represented as (user, user location, rating,

item), which means that while the user location is known, the item location is unknown or irrelevant, for example as user is rating a product from his home computer; *none-spatial ratings for spatial items*, represented as (user, rating, item, item location), the user location here in not known but the location of the item is known, for example a user is rating a tourist attraction from an unknown location; *spatial ratings for spatial items*, represented as (user, user location, rating, item, item location), this type of location based rating both user location and item location are considered and incorporated into the system, for example a user for certain city rating a museum.

CHAPTER 3

LITERATURE REVIEW AND RELATED WORK

This chapter reviews the different approaches done previously in recommender systems and context aware recommender systems and the methods used by different researchers. Moreover a number of related applications will be overviewed.

Prior work done in recommender system have generally raised and addressed the issues of scalability [17], accuracy [17], [18], the cold-start problem [20], as well as the explanation of recommendations [21]. Research in the area of mobile device recommender systems follow a tendency to focus on the visualization of the recommendation results [22] [23] [24] [25]. The monetary value of recommendations has been assessed through a look into the mobile application sales [26] [27] [28] [29] and a utilization of the context of application consumption as well as use. It has been suggested that recommender systems be made context aware through the addition of another dimension to the item-user matrix [28] which is to represent the context.

Numerous mobile recommender systems have focused on points of interest, tourism [22] [30] [31] and media [22] [32]. It has been suggested that use of explicit user ratings and input be limited as far as possible due to the concern that it would require further effort from the user [33]. Instead implicit ratings could be utilized as "predictions based on time spent reading (implicit rating) are nearly as accurate as predictions based on explicit numerical ratings" This type of rating system has found increasing support and is widely used in mobile device recommender systems.

It has been noted that level of user satisfaction does not only depend on the accuracy of the recommendations [34]. Diversification of the recommendation lists allows the recommender to further encompass the user's various tastes and metric has been proposed to achieve this [34]. One other way to improve user satisfaction is to explain recommendations [21]. This works by increasing the trust of the user in the system and hence increases the user's acceptance. This method has the advantage that it would serve to educate the user of the weaknesses and strengths of the system.

Location Aware recommender systems provides a solution that was not available in traditional recommender systems [15] suggests three types of location-based ratings: “spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items” the system set by [15] used user partitioning and travel penalty techniques that can be applied independently to support to support spatial ratings and spatial items. The outcome of the analysis that was done with real data sets showed that it provided efficient recommendations with higher quality results than traditional recommender systems. In another technique of location based recommender system [48] employed Bayesian Network (BN) and User's Preference Model and implanted a map-based interface in a mobile device. The system obtain context from a mobile device and uses user profile information. In [48] it is explained that the BN was built by an expert and a dataset was used to learn the parameter. The system collects context information such as location, time and weather and depending on those information the system displays the appropriate suggestions. The authors provided a solution to overcome resource and display limitation in mobile devices. However, further work is needed to test and confirm the usefulness and stability.

A multicontext-aware Recommendation introduced by [51] for Ubiquitous Commerce presented by a shopping application that relies on location context and audio context. The experiment employed consumer's preferences and behavior as a weighting factor. The application would be used in a shopping area, the user location context and audio context will be used to determine the user’s behavior. The authors suggested that by increasing the amount of contextual information can give a closer look on user’s current behavior and thus makes prediction of future behavior and that would provide a better recommendation process. The conclusion of this work that the method was able to recommending accurately.

3.1 Related Systems

In this section related application will be reviewed and explore the approaches and functions

3.1.1 AroundMe

AroundMe[1] is a mobile app that can be found on both android and iOS platforms, it allows users to find Points of interests according to their location.

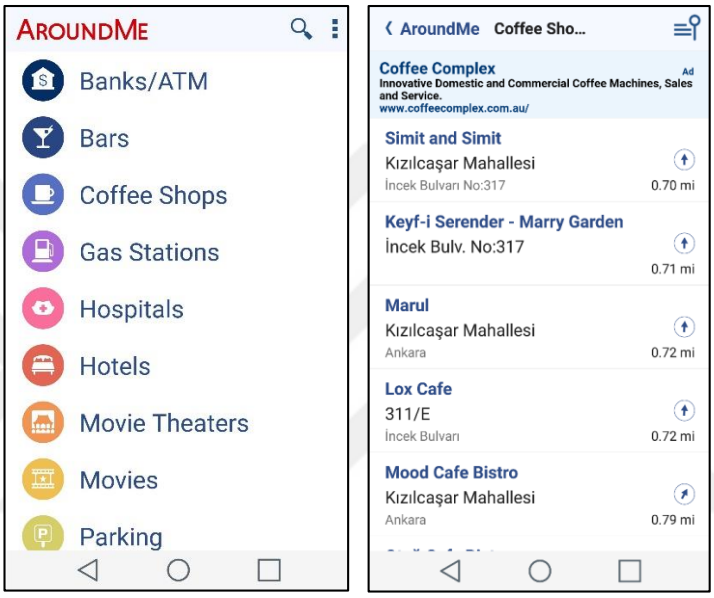

Figure 2 AroundMe Application Interface

The method used by AroundMe is straight forward as it depends mainly on the user's location and on direct user input of the category needed at the time the user makes request for a recommendation. After the user choose the thing's he/she is searching for the application will start filtering locations that fits the choice the user made and are within a close distance from the user.

[1] www.aroundmeapp.com

3.1.2 FourSquare

Foursquare[2] is described as a local search-and-discovery mobile app and a social network. It was launched around 2009 by Dennis Crowley and Naveen Selvaduria and has around 45 million users from around the world. Since it was launched and up until 2014 Foursquare allowed users to share their location with friends manually through "check in" in which the social aspects plays a major role in its functionality. However, this changed when the company released Foursquare 8.0 in august 2014.

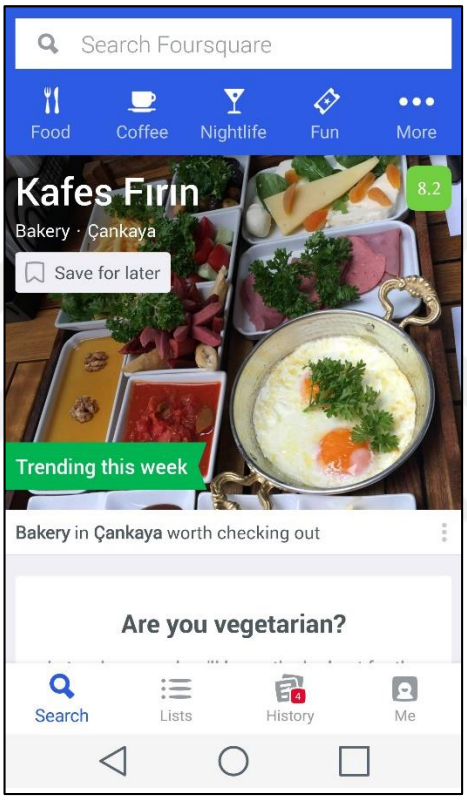

Figure 3 Foursquare Interface

Foursquare uses location to let users search find nearby places of interests such as shops, restaurants, parks etc. in its recommendation criteria, it uses personalized recommendations based on their ratings, tastes, friend's reviews, and time of the day. This means foursquare is a multicontext-aware system that employs several factors and contexts to generate recommendations.

[2] www.foursquare.com

In [47], user checking dynamics was analyzed in attempt to extract useful meaningful spatio-temporal pattern in a large-scale study of user behavior in Foursquare. More than 12 million user's *checkins* were collected over a period of more than 100 days. An analysis of the geotemporal dynamics of collective user activity was presented to reveal the daily behavior of users and their weekly patterns. It concluded that location based social networks offers new opportunities and potential of more accurate location/activity recommender systems.

CHAPTER 4

METHOD AND IMPLEMENTATION

In this chapter the methods and approaches used in this thesis will be described, including collection of data, implementation of the algorithm as well as the experimental setting. The goal of this chapter is to clearly present the means necessary to develop a context aware recommender system.

4.1 Recommendation Method and Motivation

The method used in this thesis uses content based filtering along with contextual information represented as location context. The recommendation can be shown in this formula:

$$\text{Problem: User} \times \text{Item} \times \text{Location} \rightarrow \text{Recommendations} \tag{4.1}$$

This method will be applied in a context-aware recommender system for android mobile devices. The goal is to evaluate and study content based context-aware recommender systems in a mobile device environment and test its capabilities and performance as a real life application.

4.2 System Overview

We developed a context-aware recommender system for places of interests. Figure 2 illustrates an over view of the system. Users connect to the server through the internet. After receiving contextual information which is represented in the form of the user's location and interests along with the desired distance between the user and places of interests the server will generate and provide the user with a nearby places of interests. Those recommendations include the name of the place, working hours, phone number and the exact location of points of interest. Figure 2 illustrate the structure of the system

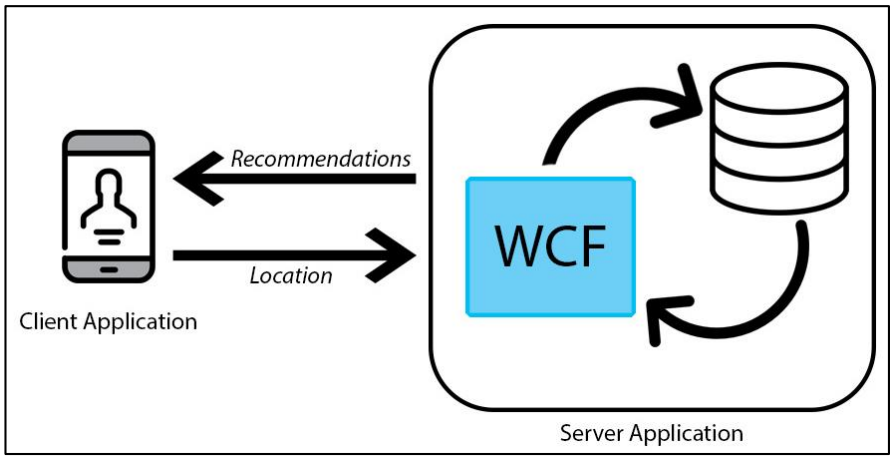

Figure 4 System Overview

4.3 Tools and Frameworks

This section describes the tools and frameworks used to developed a prototype of the system.

4.3.1 Android Platform

The system was implemented using Android platform for the client application. Android was developed by Google but then it became part of the Open Handset Alliance [35]. Android is described as an open source platform for mobile devices that was based on Linux kernel. Java is used to write the applications and the Software Development Kit (SDK) provides all the tools and APIs needed for development [36].

Figure 5 shows the android software stack and how application framework for standard Graphical User Interface (GUI) components can be used by the applications with many components such as Activity Manager and Location System for location.

There are four types of Android application components: Activities, Content Providers, Broadcast receivers and services. Activities represents the GUI, almost all activities interact with the user and each activity works independent from the other activities in an application although they work together to form a seamless user experience of an application. Content provider is a way to handle storage in an application and manages a shared set of data. A service runs in the background and does not have an interface. Location service will be used to prototype this application.

Broadcast receiver receives and responds to system announcements and while these announcements can be from other applications they usually originate from the system such as the announcement that the battery is low or that the cell network is down.

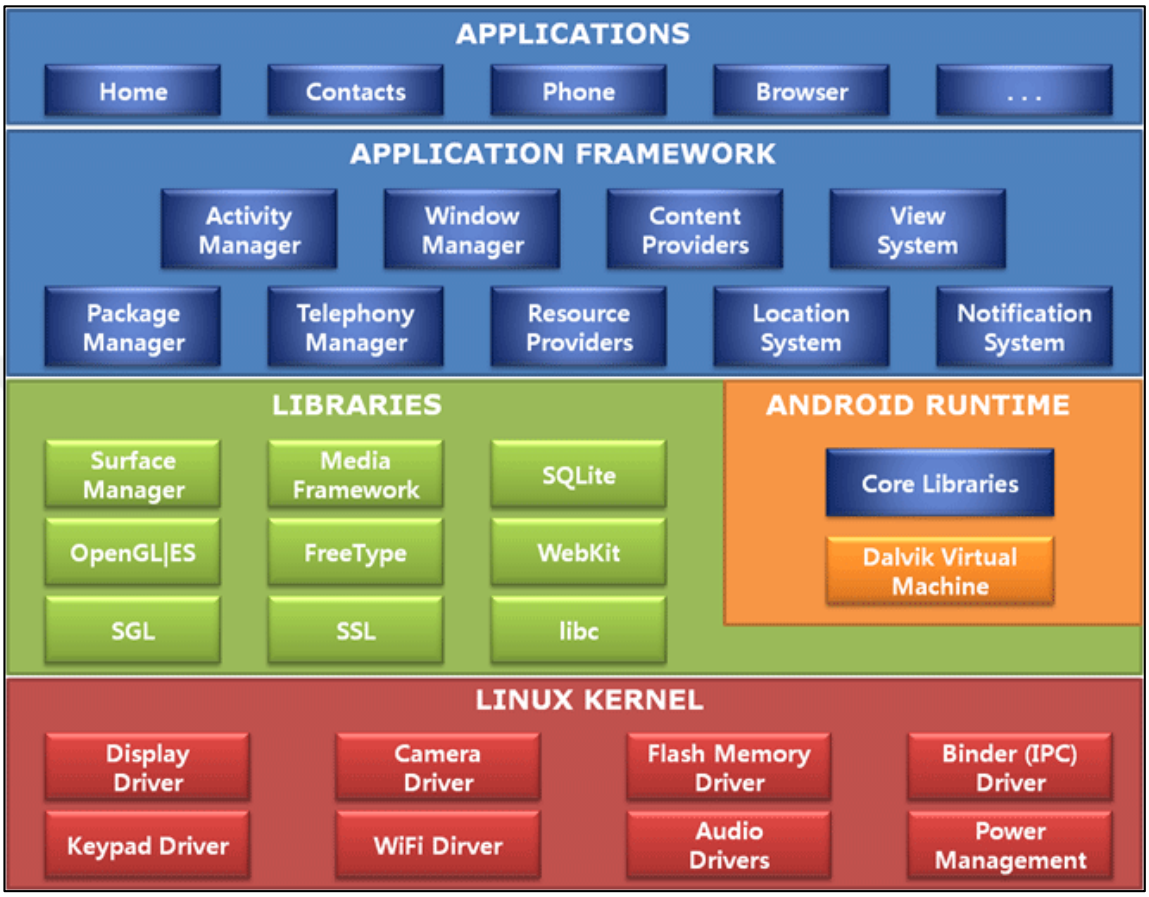

Figure 5 Android Software Stack

4.3.2 .NET Framework

The server side of the system was implemented using .NET Framework. Developed by Microsoft in the late 1990s .NET Framework is a programming infrastructure for developing and running applications services and to help create mobile, desktop, and web applications, and it contains thousands of shared code that makes developing these applications much easier [17]. It contains 3 parts; the Common Language Runtime (CLR) is a major component of the .NET and is described as a runtime environment that handles the execution of .NET. Framework Class Library (FCL) is a large class library that provides language interoperability across several programming languages

and provides services such as interaction with databases, consuming and producing XML and building SOAP based webservices and web based client applications [17]. ASP.NET is Web technology from Microsoft that takes an object-oriented programming approach and is used to create Web pages and Web services.

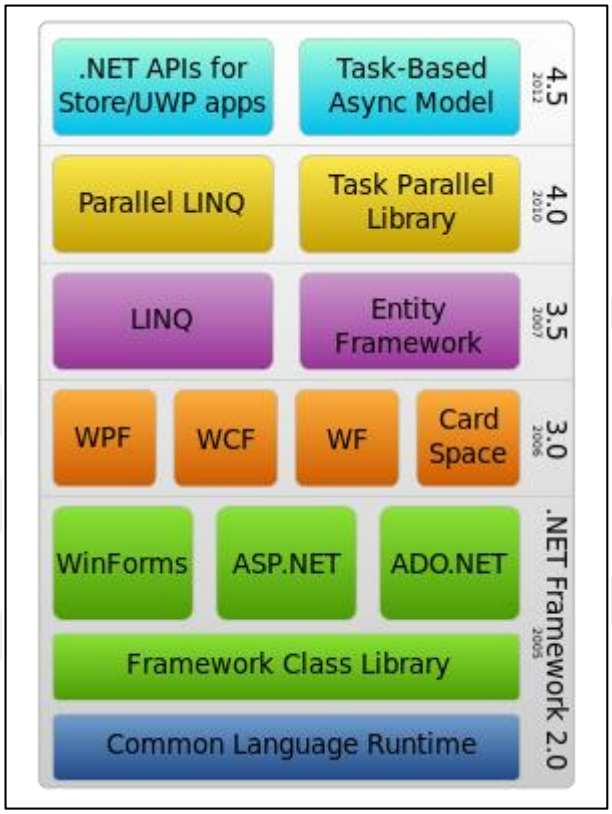

Figure 6 .NET Framework Component Stack

4.3.3 Windows Communication Foundation (WCF)

WCF service was used to create services that will communicate with the client application over the internet and with the database on the server. It will process all the operations between the client and the database as per the service criteria.

WCF is a service oriented application building framework. Service endpoints can communicate and send data as asynchronous messages from one to another. The data can be sent as a stream of binary data or characters and words as XML [38].

WCF Features

WCF has several features which makes it useful for the development of the application in this thesis. And they can be summarized as follows [38]:

- **Service Orientation:** It allows developers to create service oriented applications and allows clients to connect to any service using a platform as long as it meets the essential contracts, this is due to its loosely-coupled relationship advantage.
- **Multiple Message Patterns:** It offers several patterns to exchange messages between endpoints. Those patterns can be in the form of request where one endpoint send a request to the other endpoint. Or it can be in form of a one way message where the endpoint sends data to the second endpoint with no reply. It can also be more complex where communication between the two end points can be in a way of sending and receiving data from one to another.
- **Service Metadata:** It supports meta data to be published as http/https. This meta data can set clients to access the WCF service.
- **Data Contracts:** Since it is built using .NET Frame work, it allows using methods to provide contracts needed for development. It makes working with data easier by including a comprehensive system to create classes that represent a data entity and automatically generates the Meta data needed for the clients.
- **Security:** It provides ways to encrypt Messages between the endpoints using common standards such SSL.
- **Encodings:** It supports several encoding and transport protocols. One of these protocols is the well-known SOAP messages via Hyper Text Transfer Protocol (HTTP).
- **Durable Messages:** Which means message will never be lost as a result of disruption of the communication. This due the messages always saved to a database and whenever disruption happen, messages exchange can be resumed from the database whenever the connection is restored.
- **REST Support:** Representational State Transfer (REST) is a web 2.0 technology. WCF supports processing XML data including specific formats such as ATOM used commonly in RSS feeds, it also supports non-XML formats such as JavaScript Object Notation (JSON).

4.3.4 RESTful WCF Service

REST is an architectural style used for web development which has the advantages of being independent, interception, improved scalability and performance [39]. REST uses the common HTTP methods to delete, create, retrieve or insert as follows:

1. **GET**: Used to request a resource
2. **PUT**: Used to create or update a resource
3. **DELETE**: Used to delete a resource
4. **POST**: Used to send data to a resource and process it.

4.4 Data Representation

XML and JSON are common formats to interchange data through the web. WCF can process and produce both formats. Therefore it is important to choose the correct format to insure a fast and stable system. The two formats are described as follows;

4.4.1 Extensible Markup Language (XML)

XML is a markup language and a tool to store and transfer data independently from hardware and software. It encodes document using set of rules to make it readable by humans and machines. It's designed to make transporting data across the internet simple, useable. While it is widely used to represent arbitrary data structures [40] used in web services. Figure 7 show a simple example of XML

```
<phone>
   <company>Samsung</company>
   <name>Galaxy S3</name>
   <price>900</price>
</phone>
```

Figure 7 Simple XML Example

4.4.2 JavaScript Object Notation (JSON)

JSON was first introduced by Douglas Crockford in 2001 and it follows the same goals and is used as an alternative to the above mentioned XML format. It is described as a lightweight text-based open standard designed for human-readable data interchange and was derived from JavaScript scripting language and despite that, it is language independent and can be parsed by many other programming languages [41].

The primary use of JSON is to transmit data between web applications and servers and can also be used while writing JavaScript based applications. Figure 8 shows a simple example of JSON .

```
{
   "company": Samsung,
   "name": "Galaxy S3",
   "price": 900
}
```

Figure 8 Simple JSON Example

4.4.3 XML or JSON

To meet the need of a recommender system speed and a smooth process are priorities. JSON serialization format for the data is light-weight and smaller thus in general it is faster than XML [42]. That being said XML is still great format and has its own advantages in different aspects.

Table 1 provides some of the differences between XML and JSON.

Table 1 Some Differences Between JSON and XML

XML	**JSON**
Derived from SGML	Derived from JavaScript
Has a heavier Syntax than JSON	Has a lighter Syntax than XML
Has start and end tags	Does not include start and end tags
Does not supports datatype and arrays	Supports datatype and arrays
Supports Namespaces	Does not support Namespaces
Support for Comments	Does not support Comments
Better used for configuration	Better used for web services
Data can be changed to other formats	Data cannot be changed to other formats
Document oriented	Data oriented
Better data exchange format	Better Document exchange format

4.5 System Requirements

This section mentions the requirements needed to be achieved by the application in order to be meet the evaluation of the method used in this thesis. Those requirements have been divided into parts as follows

4.5.1 Client Application

The client application works as an interface and a link to connect the user to the Server and provide information from and to the user and allow the user to navigate to the desired place of interest. The information is sent to the server to be used to generate the recommendations. In addition, information about the recommendations will also be sent and stored into the server in ordered for it to be used while evaluating the system. Requirements for the client application are:

R1 – User Creation and Login

The client application shall provide an interface that allows new users to create a new profile and allows existing users to login to their account.

R2- Send and Update Interests

sending user's interests of different categories shall be made possible in the application.

R3 – Obtain Location

Client application shall be able to use location acquiring tools in available in the device and shall get as much accurate coordinates as possible.

R4 – Request Recommendations

Users shall be able to send a request of new recommendations to the server application.

R5 – Select Place of Interests

The client application must provide an interface that allows users to see and select places of interests and to be shown a map of the location.

4.5.2 Server Application

The server application will provide recommendations to the user using information sent by the client application. The requirements of the server application are:

R7 – Store User's Information

Server application shall be able to store information such as user's name, username etc. and also user's login information. In addition it shall also store user's interests when sent by the client application.

R8 – Recommend Using Contextual Content Based Filtering

The server application shall receive, process and then recommend using the information stored about the user's interests and the location information sent by the client application.

4.6 Design and System Architecture

In this section it will be shown how the system satisfies the aforementioned requirements, firstly by describing the system architecture and then each component in detail.

4.6.1 Decomposition

The users interact with the system through the client application interface. The interface allows the user to gain access to his or her account (R1), update interests (R2), and before they can request recommendations (R4) the client application will obtain the user's location (R3), and when the results are sent from the server the user can have information about the places of interest nearby (R5) , and then rate the recommendation (R6). The server application will receive user information that will be sent from the client application and stored in the server (R7). And to generate and send recommendation the server application will receive contextual information (location) from the client application and filter it according to the user's preferences (R8).

4.7 System Components

The recommender system will be designed as client and server and with the following components to conform to the requirements.

4.7.1 Client Application

MySpot is an Android based prototype application developed to show a list of nearby places of interests, and the users can choose the distance from those places by selecting 4 different options (200m, 500m, 700m, 1000m). These places vary from restaurants, malls, shops, cafes etc.

4.7.1.1 Interface

MySpot interface is user friendly interface that includes maps from Google maps [3] service to increase the usability. The application consists of 4 screens (Activities) and they can be summarized as follows:

Login: Login screen will allow the user to open a session with a username and a password. See Figure 9

[3] maps.google.com

Register: New users will be able to create a profile that includes some information about them and will include their username and password.

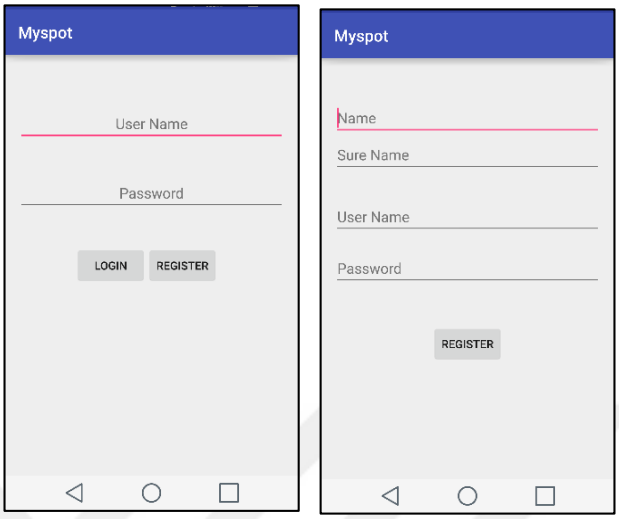

Figure 9 Login and Registration Interfaces

Update Interests: This screen will allow users to update their interests which is to be sent and stored in the server database and later used when generating recommendation. See Figure 10.

Figure 10 Interests Screen

Main Screen: The main screen will allow user to refresh the recommendation list by refreshing the results every time they are needed. It will also allow them to access the above Update interest screen. The users can also logout of their current session.

Figures 11 shows the different screens of the application's main screen in different stages from requesting a recommendation to giving the route to a point of interest.

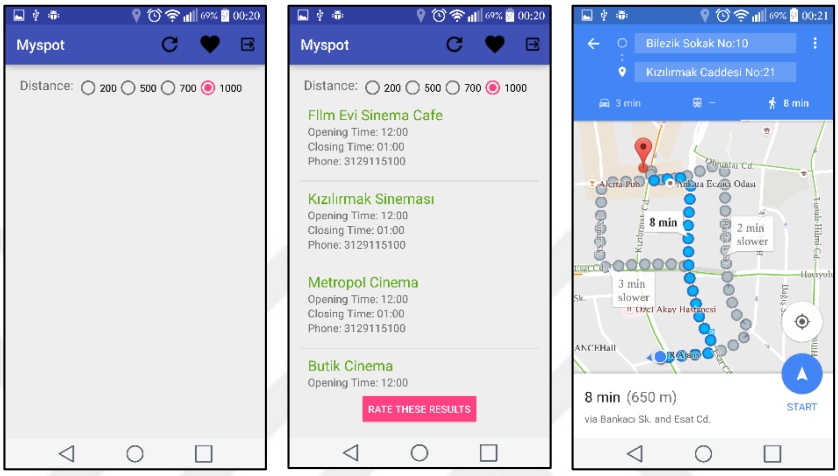

Figure 11 Main Screen

4.7.1.2 Obtaining Location

The other important function of the client application is getting an accurate data about the user's application. Since systems that rely on location became so common, modern mobile phones are equipped with the necessary technologies to obtain location information. However, getting this information can be tricky in some cases. Regardless of the source that provides these information it can contain some errors due to several reasons, for example when the user is moving the location must be updated more often to give an accurate position or when the user is located in a closed location such as a building or a tunnel. Moreover speed accuracy and battery-efficiency should all be taken in consideration when choosing between GPS, Network Location Provider and Wi-Fi networks as each source has its own advantages and disadvantages [43].

In our system we mainly use a combination of GPS sensor and Network Location Provider data.

GPS

GPS (Global Positioning System) sensors receive signals from the satellites around the earth and calculates the position. The idea was established in the 1960s by the United States and the system was launched in 1978 and became fully operational in 1995. The system consist of 24 satellites orbiting around the earth and broadcasting their position and exact time. Using this information the GPS receiver must receive the signal from 4 different satellites and using the position and time information sent by each of satellite the exact position of the receiver can be calculated with an accuracy of 2-3 meter range. Devices such as smart phones and tablets usually use those sensors for various functions and application [44].

Network Location Provider

Cellular network is another way to discover the users' location is used widely by location based applications in mobile devices. GSM (Global System for Mobile communication) is among the most commonly used network platform around the world. Typically each network provider has its own HLR (Home Location Register) database that records user information and when the user change location the new location will be recorded via VLR (Visitor Location Register) and sent to the HLR. Each VLR has a unique location area identity allowing HLR to find the user's location once the location area identity is determined [45]. Each of those methods come with its own advantages and disadvantages. Table 2 shows a comparison between the methods.

Table 2 Differences between GPS and Network Location Provider

GPS	**Network Location Provider**
Accurate to a range of few meters.	Depends on the cell towers and can be inaccurate.
Difficult to work indoors and in some cases it does not work at all.	Can work indoors and outdoors.
Consumes a lot of power and drains the battery.	Uses less battery power.
Relatively slow response.	Faster response.

In our system we obtain the location data through Android Fused location provider available in Google play service, it allows us to get the last known location of the user and request for the location to be updated and stored. It chooses the best location provider whether it is GPS or Network Location Provider to insure high accuracy and quick response. However, since location is a sensitive information and can be a privacy concern, a permission must be granted by the user upon installing the application.

4.7.2 Server Application

The server application communicates with the client application to obtain contextual data and user information then provide recommendations according to the method used in this thesis. The server is running Windows Server 2008 and the system is based on Microsoft IIS.

4.7.2.1 Database

The database is designed and handled with MSSQL Server 2012 R2. It contains information about the user including name, surname, user name and password, it will also store information about places of interests and including name, longitude, altitude, working hours and phone number.

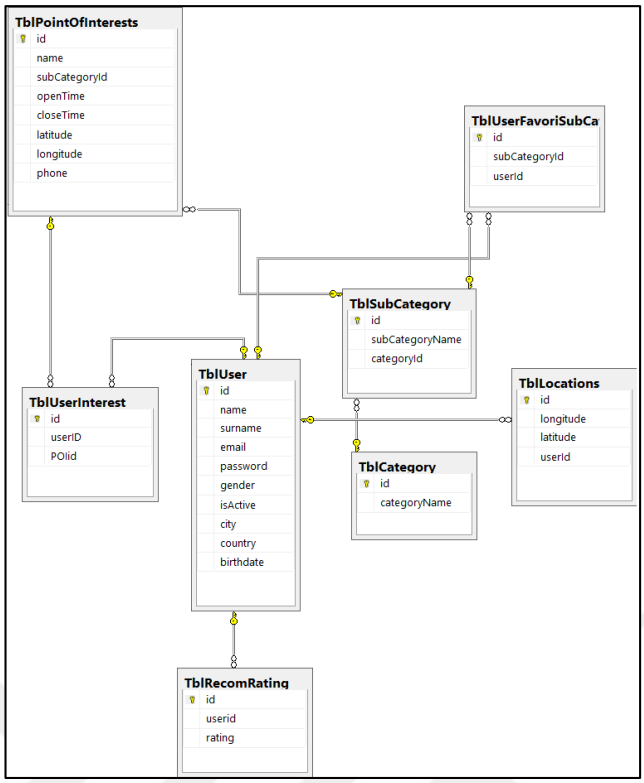

Figure 12 Visualization of the Database Tables and Relations

As shown in Figure 12 the database contains a number of tables which the server write and reads from. The table are named *TblUser*, *TblUserFavoriateSubCategory*, *TblCategory*, *TblSubCategory*, *TblPointsOfInterests* and *TblRecomRating.*

TblUser: This table contains the user's profile information including id, name, surname, email, and password. This table will be called each time a user request to login, update interests or request a recommendation. Each user has a unique id that will be used to identify user's preferences.

TblCategory: This table contains the main categories that organize the locations in the database and gathers the subcategories in a bigger class. The purpose of this table is to organize data input process.

TblSubCategory: This table contains a list of subcategories that will be used to classify interests for each users. It contains two columns; Id and SubCategory.

TblUserFavoriateSubCategory: This table will store each user's preferences and will write the information every time the user updates their interests. The server will use

this table to determine user's interests and use the information to suggest places within the interests of the user.

TblPointsOfInterests: This table was used to store all the locations and location information such longitude, altitude, phone number, open time and will also include information of which subcategory each location belong in. This table used to determine locations near the user within the recommendation criteria and will be used every time the user requests a recommendation.

TblRecomRating: While this table has no effect on the recommendation process, it will be used to evaluate the recommendation process. It will contain feedback from the users.

4.7.2.2 Recommender Engine

The recommendation engine links and communicates data between the database in the server and the client application. It processes the information coming from the client application mentioned in Section 4.5.2 and then sends back data from database after filtering it according the recommendation criteria.
The WCF introduced in the server application consists of several web services that can be invoked by the client application when communicating with the server by sending an http GET method request to the WCF with a URI (Uniform Resource Identifier) . Each of these services serves a different purpose in the system and can be described as the following:

`addNewUser:` When using the application, users need to create a user profile with a unique *username* and *password*, and the reason of having a unique username for each user is to make sure the recommender system will give the correct recommendation for each user depending on their preferences. From the client application an http request with the following template:

```
(addNewUser/{Name}/{Surname}/{Email}/{Pass})
```

This will be sent to the WCF to invoke the user registration webservice. It contains information of the user name represented with the variable *`Name`*, the user surname

represented with the variable *Surname,* then the email must be unique to distinguish users profiles and it is represented with the variable *Email* and lastly password represented with the variable Pass which adds protection to the user's account.

checkUser: After an account is created for a user they can login to the system through the client application that invokes this webservice in order to determine the user's account and allow access to it the user must enter an email and password. Similar to the previously mentioned webservice addNewUser, the client application will request the login by the following http request template:

(checkUser/{Email}/{pass})

where Email is the user's unique email and pass is the user's password. If the login is successful the webservice will return the user's id number that will be user later in the recommendation process, gathering information about the user and when getting feedback from the user. However, if the login fails the webservice will return "0" which indicates that the email or password is incorrect.

sendInterest: An important part of the method used to generate recommendations is to collect information about the user's interests which will be done explicitly by asking the users to fill their profile with their interests (e.g. shop, restaurants, gyms etc.). This webservice will be invoked by the following http request template from the client application.

(sendInterest/{Id}/{SubCatId})

Where Id is the user unique id and SubCatId is the sub category of the interests.

sendPOI: This webservice is the main service where the recommendation method will be applied. It gets the information from the user through the client application. This information must contain user's id, user's latitude, user's longitude, and the desired distance of the places nearby. The http request must follow the following template:

sendPOI/{Id}/{Lat}/{Long}/{Dis}

where Id is the user's `Id`, Lat is the user's latitude at the moment the request was made, similarly `Long` is the user's longitude and `Dis` is the distance from the places of interests and can have values of 200m,500m,700m or 1000m.

The result of a successful recommendation request will contain information about the nearby places of interests such as the name of the place, the subcategory, working hours, latitude and longitude (location) and the phone number. This information will be in JSON format that later will be consumed and parsed by the Android client application to make them useable. Figure 13 show a typical recommendation response from the server in JSON format.


```
[
    {
        "Id": 34,
        "Name": "Place0",
        "SubCategoryId": "15",
        "OpenTime": "always open",
        "CloseTime": "always open",
        "Latitude": "39,908807",
        "Longitude": "32,878389",
        "Phone": "888754122 "
    },
    {
        "Id": 30,
        "Name": "Place1",
        "SubCategoryId": "13",
        "OpenTime": "7:00",
        "CloseTime": "10:00",
        "Latitude": "39,909301",
        "Longitude": "32,876123",
        "Phone": "2110011 "
    },
    {
        "Id": 31,
        "Name": "Place2",
        "SubCategoryId": "13",
        "OpenTime": "10:00",
        "CloseTime": "18:00",
        "Latitude": "39,907832",
        "Longitude": "32,877267",
        "Phone": "66611120 "
    }
]
```


Figure 13 Typical Recommendation Response from the Server in JSON Format

`recomRating:` This webservice will be used in the experiment stage of this thesis. It will collect users' feedback after each recommendation in the form of rating scheme

where the relevancy of the recommendation results will be evaluated. The http request template is given as the following:

```
recomRating/{Id}/{Rating}
```

Where Id is the user's id and rating is the rating given by the user.

CHAPTER 5

EXPERIMENT

This chapter provides a description of the experiments and settings. In this study, in order to conduct an evaluation of the recommender system, analytical data is to be obtained from two sources, namely, user feedback and device analysis.

5.1 Defining Goals

The goal of the experiments is to gather data and analyze th recommendation relevancy and system performance. These aspects are explained as the following:

5.1.1 Recommendation Relevancy

As the approach is based on using content base filtering and contextual data to obtain recommendations, these recommendations accuracy in terms of relevancy to the user will be examined and analyzed.

5.1.2 System Performance

System performance will be tested from different physical aspects as follows: **Location Accuracy**: As was mentioned in the previous chapter Android system provides strategies to obtain location and is used for location based application, these strategies will be tested in terms of accuracy of the location obtained. **System Speed**: As these type of applications require a fast and responsive system to provide a good user experience.

Battery Life: Battery life is a huge concern for every application developer since the application can use several system resources and drain the battery causing a negative user experience.

5.2 Experiments

In order to satisfy the goals and evaluate the system experiments were conducted for different aspects of the system as follows:

5.2.1 Recommendation Relevancy

In order to determine the relevance of the recommendations generated by the system a real world data was gathered from real people who participated in the test. Ten Android phones users installed Myspot on their devices and used the system for a duration of 1 week in various areas in Ankara. The database included more than 300 different locations with different categories scattered around the city, thus giving the chance for the user to test the actual system and provide feedback in the form of a rating of the recommendation relevancy. Then, these data are gathered and analyzed. However, some of the ratings were impacted by using the application outside the area that was set in the database which caused the system to return empty recommendation as a result of the lack of data in that area. The data collected from the participants are stored in a database in the `recomRating` table. Each recommendation and will be associated with the user id.

5.2.2 System Performance

For system performance testing, an LG-D855TR running Android version 5.0 was used for testing under *Android Device Monitor;* a standalone tool that helps developers to profile, debug and analyze their applications. The tool is provided with various tools to test Memory and can be described as the following [46]:
Memory Monitor: This tool provides information about the amount of memory the app is using and helps identify memory leaks and evaluate the memory usage.

CPU Monitor: It displays the CPU usage in real time and allows it to monitor the CPU utilization.

GPU Monitor: It allows the developers to find out how much time it takes to render the frames of UI (User Interface) Window. It helps to reduce memory usage by using the information to optimize the code that displays the graphics.

Network Monitor: This tool will provide information to analyze network requests. Network use can have an effect on the battery life.

All these tools are available when the using a real device. However, some of these tools such as Network Monitor are disabled when using Android emulator. Moreover, battery performance and usage are not included. Therefore, *PowerTutor*[4] an Android application developed by University of Michigan for mobile devices is used to measure power consumed by the system components and different applications.

[4] http://ziyang.eecs.umich.edu/projects/powertutor/

CHAPTER 6

RESULTS AND EVALUATION

In this chapter the results of the experiments will be evaluated for recommendation relevancy and system performance.

6.1 Recommendation Relevancy

After finishing the testing period for 10 different users, 124 ratings were obtained from the participants.

The average rating for all users was 3.38. However, during the testing there was cases of an empty recommendations due to testing outside the testing area and this led to a ratings of “1” as the user did not see any locations, and after excluding the ratings related to this issue the average rating appeared to be 3.69 based on 109 ratings. Table 3 shows the rating result averages for each of the 10 users:

Table 3 Ratings Given By Each User

User Id	**Number of Ratings**	**Average Rating.**
63	18	3.72
64	4	3.66
65	11	3.81
67	7	4.1
68	12	3.66
69	10	3.7
70	9	3.88
71	17	3.68
72	12	2.92
73	9	3.77
Total	109	3.7

Figure 14 shows the distribution of the ratings.

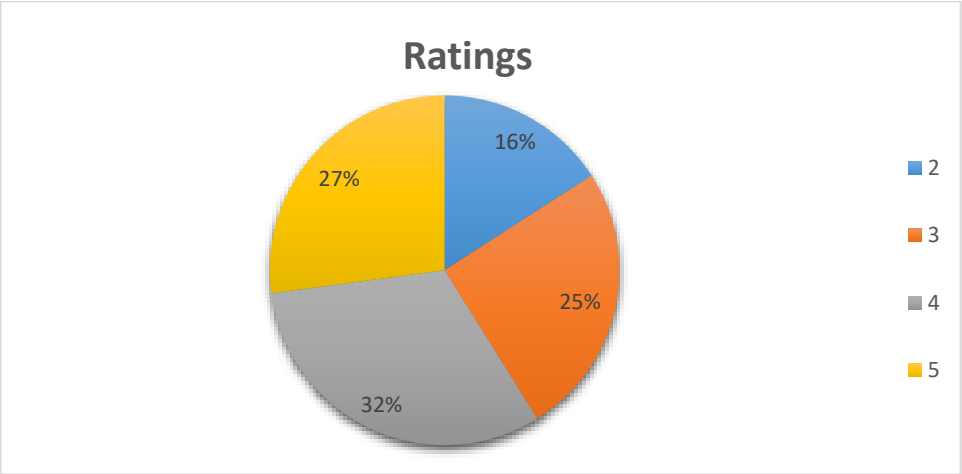

Figure 14 The Distribution Of The Ratings Given By All The Users (5 Is Highest)

This chart shows that 59% of the ratings of the results given by the recommender system were between 4 and 5 which can be considered a positive feedback. While 25% had feedback that was neither positive nor negative and finally 16% gave a feedback that can be considered negative.

6.2 Analyzing the Findings

These findings can be interpreted as 52% say the rating was useful and relevant while the rating that was giving 3, 2 or 1 showed that the users either found the ratings were not relevant or that there were no results at all. The absence of recommendation results provided by the system in some cases was due to the user being outside of the supported zone, since the database included places around certain areas in Ankara, thus the system would return no results within the recommendation criteria.

Other results can be rated negatively due to hardware limitations. When using the system indoors the GPS can be slow or would not return results and then the system would rely on the network location service or the last saved location and those 2 sources can be inaccurate which would cause the system to give inaccurate results.

6.2.1 System Performance

This section will discuss the data obtained from the Android device monitor tool for each test while the application was connected and returning results from the server. The speed of returning recommendation can depend on the device general performance that can vary from one to another, and on the server performance. Since this test was made in relatively small scale of 10 to 12 people at most, latency or slow performance was not noticeable nor was it measurable. However, these types of systems are known to eventually hitting hardware limitation barrier when reaching certain amount of users and requests. This can cause the system to slow down or crash. And from the device perspective, the following data was collected:

6.2.2 Memory Allocation

The memory utilized by the application is represented in Figure 15 where the horizontal axis represents time in seconds and the vertical axis represents the amount of memory allocated in Mega bytes. It shows that the application allocated 31.00MB,and while making requests and communication with the server application the amount increased to around 32.00MB. This amount is not significant for a modern smart phone or mobile device and thus it does not curb the device performance not effected the speed of the application itself.

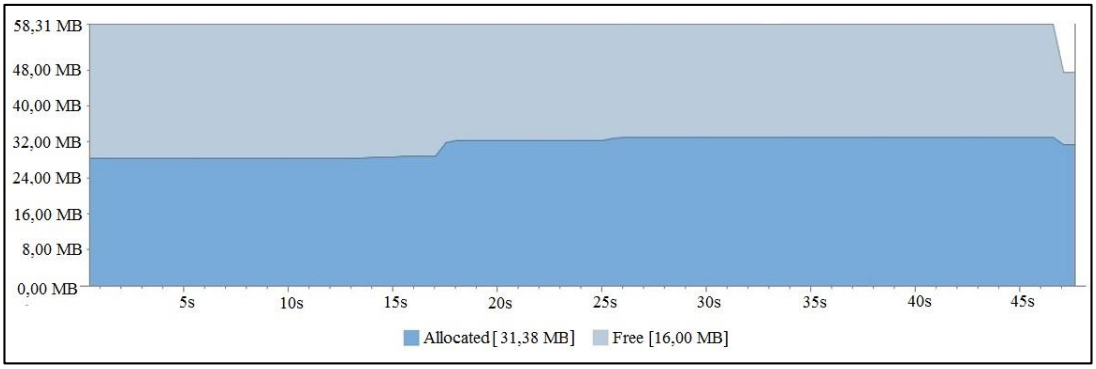

Figure 15 Memory Allocation

6.2.3 CPU Utilization

The CPU utilization was in a minimum amount, with small spikes that did not exceed 30% in the peak of the application work while communicating with the server application. It worth to mention that the application does not require heavy processing power since most of the work and calculation is made in the server application.

Figure 16 shows a timeline if the CPU utilization along the operations that was done. The vertical axis represents the percentage of CPU utilized by the application and the horizontal axis represents time in seconds.

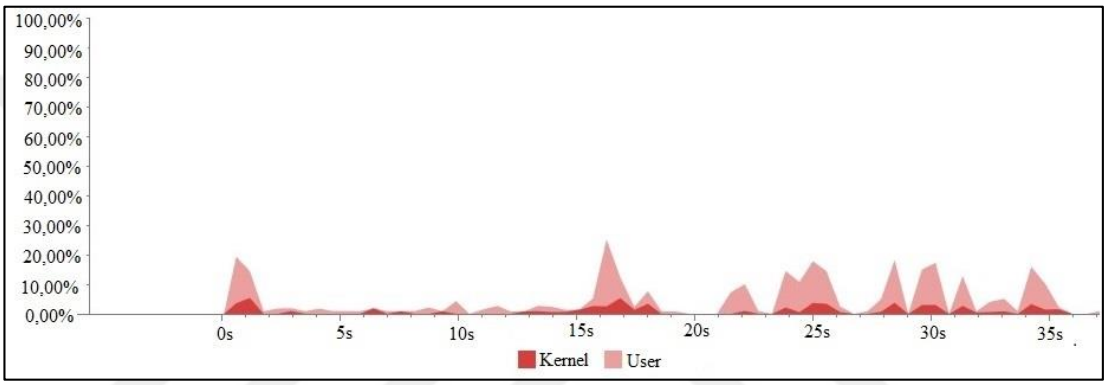

Figure 16 *CPU Utilization*

6.2.4 Network Requests

The graph in Figure 17 where the horizontal axis represents the amount of data sent and recieved and the the vertical axis represents time in seconds. The figure shows the time line when requests such sending interests to be stored into the database or sending a request of recommendations. Since HTTP requests is a small amount of information, the uploaded data was relatively tiny, similarly the downloaded data through JSON format which is known for being light weight format, was small. It is important to mention that the application and the server exchanged textual data only with no visual or auditory information that might increase the size of the data.

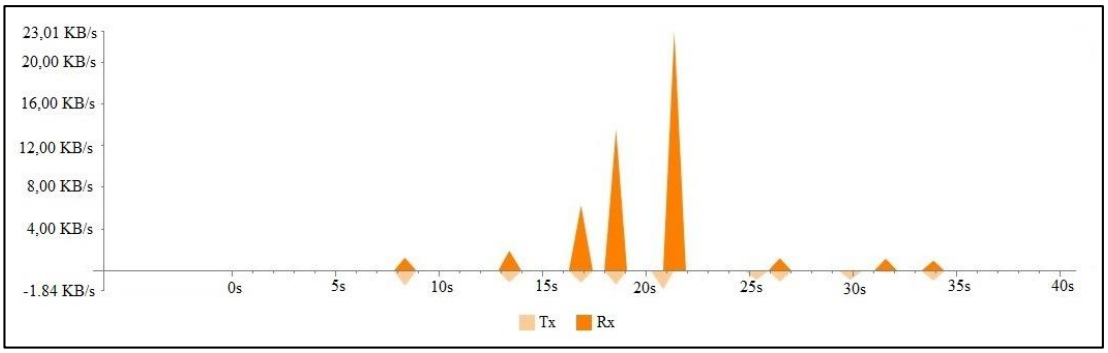

Figure 17 *Network Requests*

6.2.5 Accuracy

The accuracy of the user's location depends on the source used to obtain the location data, either GPS provider or location provide. During the tests the system both providers were tested separately and together. Network provider showed that while it gives quick response it sometimes can be inaccurate and wasn't giving the actual correct location of the user. While GPS provider was excellent in providing highly accurate location, it had two problem; it was almost limited to open areas. Since GPS require communication between the device and satellites, some indoor areas had bad reception and thus slowing the process of obtaining location significantly.

When both ways were combined, obtaining location became faster and easier most of the time. It is worth to be mentioned that most modern mobile device uses this way to obtain location.

6.2.6 Battery Consumption

The Android application did not have a background service that works while the application is inactive. Therefore, the only time the application was using resources actively is when the user request a communication with the server such as logging in or requesting a recommendation Figure 18 shows Myspot was using merely 0.05% over the period of 15 minutes usage.

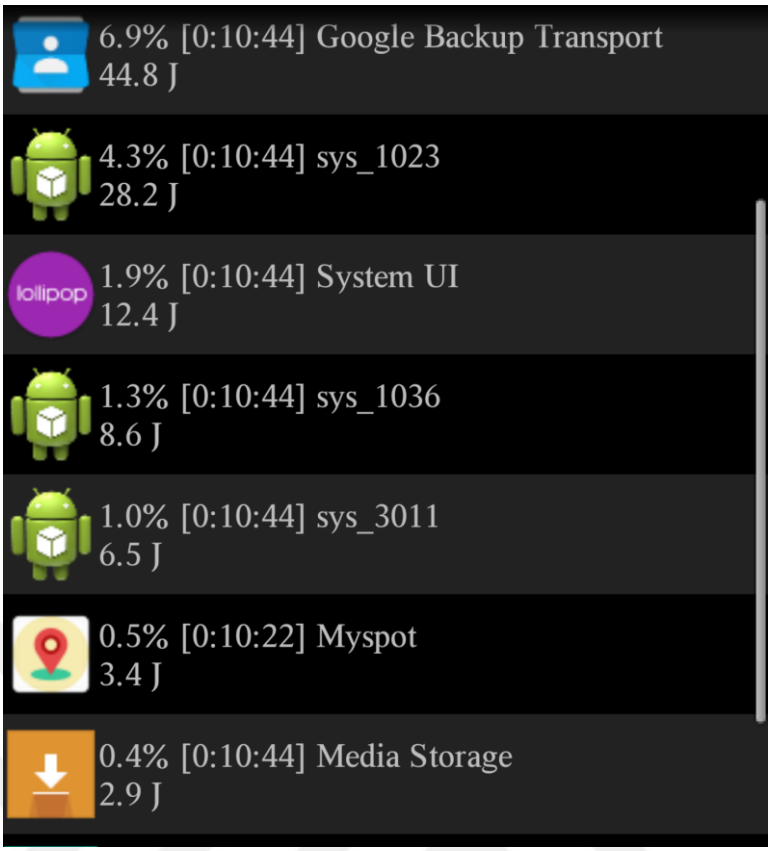

Figure 18 Energy Usage over Time

However, updating location which works in the background can increase power consumption due to the use of the GPS which is known to be power consuming. During the test enabling the GPS showed that it increased the energy consumption by 400mW. Figure 19 shows a chart of power consumption over the time the application was being used.

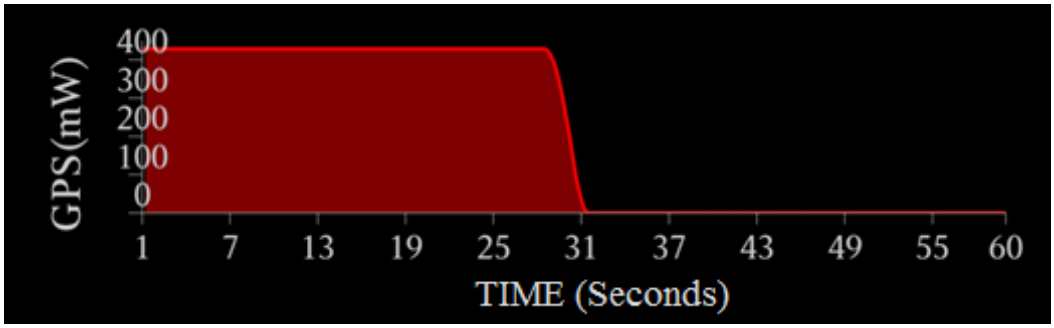

Figure 19 Power Consumption over Time

This amount of power consumed by GPS is high when compared to the other system components. PowerTutor showed that the GPS was responsible for about 30% of power consumption when the mobile phone is actively being used and it also works in the background which can make it reach 50% of the total consumption when the LCD screen is off and the processor is not active.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis, an evaluation and test was conducted on a context-aware recommender system that uses content-based approach with incorporation of context in form of location information. The findings showed that users found the recommendations related to what they were expecting from the system this means a content based recommender systems can be improved dramatically when including contextual information such as location information and can turn more relevant and meaningful results than none context-aware systems. Moreover, the earlier mentioned multicontext-aware recommender system [51] reflects a conclusion that context information present a strong candidate for recommender systems in various applications, on various domains. For example, Foursquare as a close related system to the system presented in this thesis demonstrate this by incorporating more context such as social context in its algorithm which showed better results during the testing.

As for the application and system performance tests, it was determined the Android application was lightweight and responsive when applying the basic tasks of the method used and has little impact on its general performance and energy consumption. On the other hand, in the server part such systems can reach hardware limitation due to the inflation of the amount of the clients using the system, which can lead to a slow performance or a total crash.

In the future of context-aware recommender system more research can be conducted and more complex machine learning techniques can be incorporated to gain knowledge about the users more accurately and implicitly. More context factors can have a great impact to the accuracy and relevancy of recommendations and predictions.

REFERENCES

1. **Mustansar, A., Ghazanfar, A., and Prugel-Bennett, A. (2010),** "*A Scalable Accurate Hybrid Recommender System*", in Third International Conference on Knowledge Discovery and Data Mining, pp. 94-98.

2. **Anand, S.S., Mobasher, B.7 (2005).** *"Intelligent techniques for web personalization," Intelligent* Techniques for Web Personalization, pp. 1–36.

3. **Mahmood, T., Ricci, F. (2009).** "*Improving recommender systems with adaptive conversational strategies*". C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82.

4. **Adomavicius, G. And Tuzhilin, A. (2008).** "*Context-aware recommender systems"*. In Proceedings of the 2008 ACM Conference on Recommender Systems, (RecSys 2008), pp 335–336, New York, USA.

5. **Resnick P. and Varian H. R. (1997).** "*Recommender systems*", volume 40. ACM Press, http://doi.acm.org/10.1145/245108.245121.

6. **P. Melville, R. Mooney, and R. Nagarajan. (2001).** "*Content-boosted collaborative filtering*", vol- ume 40. In Proceedings of the 2001, citeseer.ist.psu.edu/melville01contentboosted.html.

7. **Badrul, M., Karypis G., A. Konstan, J. And Reidl, J. (2001).** "*Item-based collab- orative filtering recommendation algorithms*" citeseer.ist.psu.edu/sarwar01itembased.html.

8. **. Herlocker, L. (2000).** "*Understanding and Improving Automated Collaborative Filtering Systems*", PhD Thesis, University of Minnesota.

9. **Massa, P. & Avesani, P. (2004)**. "*Trust-aware collaborative filtering for recommender systems*." Proc. of Federated Int. Conference on the Move to Meaningful Internet: CoopIS, DOA, ODBASE, citeseer.ist.psu.edu/article/massa04trustaware.html.

10. **Desrosiers, C. & Karypis, G. (2011)**." *A comprehensive survey of neighborhood-based recommendation methods*." In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, Recommender Systems Handbook, pp. 107–144. Springer US.

11. **Burke, R. (2002)**. "*Hybrid Recommender Systems: Survey and Experiments.*" User Modeling and User-Adapted Interaction, 12(4):331–370.

12. **Burke, R. (2000)**. "*Knowledge-based recommender systems. Encyclopedia of Library and Information Systems*", pp.69, 175-186.

13. **Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles. (1999).** "*Towards a better understanding of context and context-awareness.*" In Handheld and ubiquitous computing pp. 304–307.

14. **Palmisano, C., Tuzhilin, A., and Gorgoglione, M. (2008).** "*Using context to improve predictive modeling of customers in personalization applications*." IEEE Transactions on Knowledge and Data Engineering, 20(11):1535– 1549.

15. **Levandoski, J. J., Sarwat M., Eldawy A., Mokbel M. F. (2012).** "*LARS: A Location-Aware Recommender System.*", Proceedings of the International Conference on Data Engineering, ICDE.

16. **del Carmen, M., Ilarri S., Trillo-Lado, R., Hermoso, R.. (2015)**. "*Location-Aware Recommendation Systems: WhereWe Are and Where We Recommend to Go.*" RecSys LocalRec.

17. **Mustansar, A., Ghazanfar, A. and Prugel-Bennett, A.** "*A Scalable Accurate Hybrid Recommender System*", Third International Conference on Knowledge Discovery and Data Mining, 2010, pp. 94-98.

18. **Bell, R., Koren, Y. And Volinsky C. . (2007).** "*Modeling relationships at multiple scales to improve accuracy of large recommender systems*", Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 95-104.

19. **Shean M. McNee, John Riedl, and Joseph A. Konstan, (2006),** "*Being accurate is not enough: how accuracy metrics have hurt recommender systems*", Conference on Human Factors in Computing Systems, pp. 1097-1101.

20. **Ahn, H. J. (2008).** "*A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem*", Information Sciences, 178(1), pp. 37-51.

21. **Jonathan, L. Herlocker, Joseph A. Konstan, and Riedl, J. (2000).** "*Explaining collaborative filtering recommendations*". Proceedings of the 2000 ACM conference on Computer supported cooperative work pp. 241-250.

22. **Ricci, F. (2011).** "*Mobile Recommender Systems*". International Journal of Information Technology and Tourism. 12(3):205-231.

23. **Church, K., Smith, B., and Keane, M. (2006).** "*Evaluating Interfaces for intelligent Mobile Search*". International Cross-disciplinary workshop on Accessibility, pp.69-78 ACM, New York.

24. **Nguyen, Q. N., Ricci, F., and Cavada, D. (2004).** "*Critique- based Recommendation for Mobile Users: GUI Design and Evaluation*", 3rd International Workshop on 'HCI in Mobile Guides'.

25. **Hegelich, K. & Jannach, D. (2009).** "*Effectiveness of different recommender algorithms in the Mobile Internet: A case study*", RecSys '09 Proceedings of the third ACM conference on Recommender systems pp. 205-208.

26. **Woerndl, W., Schueller, C. and Wojtech, R., (2007)**, "*A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications*",in Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop pp.871-878.

27. **Woerndl, W. & Groh, G. (2007).** "*Utilizing Physical and Social context to improve Recommender Systems*", Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Workshops pp. 123-128.

28. **Hosseini-Pozveh, M., Nematbaksh, M. and Movahhedinia, N. (2009).** "*A multidimensional approach for context-aware recommendation in mobile commerce*", CoRR, abs/0908.0982.

29. **Adomavicious, G., Sankaranarayanan, R., Sen, S., and Tuxhilin. (2005),**"*Incorporating contextual information in recommender systems using a multidimensional approach*", ACM Trans. Inf. Syst., 23(1):103-145.

30. **Cena, F., Console, L., Gena, C., Goy, A., Levi, G., Modeo, S. and Torre, (2006).** "*Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide*", AI Commun, 19(4):369-384.

31. **Vincent, W. Zheng, B. C., Zheng, Y., Xie, X., and Yang, Q. (2010).** "*Collaborative Filtering Meets Mobile Recommendation: A User- centered Approach*",in proceedings of AAAI conference on Artificial Intelligence.

32. **Shardanand, U., and Maes, P. (1995).** "*Social Information Filtering: Algorithms for Automating 'Word of Mouth'*", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems pp.210-217

33. **Nichols, D. (1997).** "*Implicit rating and filtering*", in Proceedings of the fifth DELOS Workshop on Filtering and Collaborative Filtering pp. 31-36.

34. **Ziegler, C., M.McNee, S., A. Konstan, J., and Lausen, G. (2005).** "*Improving Recommendation Lists Through Topic Diversification*", in Proceedings of the 14th international conference on World Wide Web pp. 22-32.

35. **Open handset alliance.** *"Industry Leaders Announce Open Platform for Mobile Devices".*
Accessed 1-05-2016.
http://www.openhandsetalliance.com/press_110507.html.

36. **Google Inc.** *"Introduction to Android".* Accessed 19-05-2016
http://developer.android.com/guide/index.html.

37. **Microsoft.** *".NET Documentation".* Accessed 19-07-2016.
https://docs.microsoft.com/en-us/dotnet

38. **Microsoft**. *"What Is Windows Communication Foundation?"*. Accessed 01-06-2016.
https://msdn.microsoft.com/en-us/library/ms731082.

39. **Evans, K. (2008).** "*Creating RESTful Services Using WCF"*, blogs.msdn.microsoft.com/kaevans/2008/04/03/creating-restful-services-using-wcf.

40. **Fennell, P. (2013).** "*Extremes of XML*". XML London 2013 Conference pp. 80-86.

41. **json.org. "***Introducing JSON***"**. Accessed 19-08-2016. http://www.json.org.

42. **json.org.** *"JSON: The Fat-Free Alternative to XML"*. Accessed 19-08-2016 http://www.json.org/xml.html.

43. **Google Inc.** *"Location Strategies"*. Accessed 21-08-2016. *https://developer.android.com/guide/topics/location/strategies.html*.

44. **Official U.S. Government information about the Global Positioning System.** *"What is GPS?"*. Accessed 30-8-2016, *http://www.gps.gov/systems/gps.*

45. **John J. Borkowski. Network inclusion probabilities and Horvitz-Thompson (1999).** "*estimation for adaptive simple Latin square sampling*.", Environmental and Ecological Statistics 6 pp. 291-311.

46. **Google Inc.** *"Android Monitor Overview"*. Accessed 22-08-2016. https://developer.android.com/studio/profile/android-monitor.html.

47. **Noulas, A., Scellato, S., Cecilia, Pontil, M. (2011).** "*An Empirical Study of Geographic User Activity Patterns in Foursquare*.", ICWSM 11, 70-573.

48. **Park, M., Hong, j. And Cho, S. (2007).** ”*Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices.*”, In Proceedings of the 4th international conference on Ubiquitous Intelligence and Computing pp.1130-1139.

49. **Oxford Advanced Learner's Dictionary (1995),** Oxford University Press, USA.

50. **Merriam-Webster's Learner's Dictionary (2006).** Merriam-Webster Inc.

51. **Kwon, J. & Kim, S.** “*Multicontext-Aware Recommendation for Ubiquitous Commerce*”, Distributed Computing and Internet Technology
Volume 4317 of the series Lecture Notes in Computer Science pp 291-304.

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ALAZZAWI, Aws Nazar Ahmed

Date and Place of Birth: 29 June 1986, Baghdad

Phone: +90 507 683 83 76

Email: ausnazar@hotmail.com

EDUCATION

Degree	Institution	Year of Graduation
B.Sc.	Middle East Technical University, Computer Engineering (Exchange student)	2011
B.Sc.	Saba University, Computer Engineering	2008
High School	Abdulnasir High School	2004

WORK EXPERIENCE

Year	Place	Enrollment
2011-2014	Self Employed	Web Developer
2008-2010	Polypus Int. (Yemen)	IT

LANGUAGES

Arabic – Native, English – Advanced, Turkish – Intermediate