Enhancement In Navigation System By Unique Tagging

A dissertation report submitted in partial fulfillment of the requirements for the award of degree

of

Master of Technology

in

Computer Science & Engineering

with specialization in

Artificial Intelligence and Artificial Neural Network

By

Gaurav Kumar Rajput

R102212002

Under the Esteemed Guidance of

Mr. Pakanj Badoni **Assistant Professor** **CIT-CoES, UPES**	**Mr. Vinay Rao** **Senior Technical Supervisor** **I2India Pvt limited**

Centre for Information Technology

University of Petroleum & Energy Studies

Bidholi, Via Prem Nagar, Dehradun, UK

April – 2014

CERTIFICATE

This is to certify that the project work entitled **"Enhancement In Navigation System By Unique Tagging"** done by **Gaurav Kumar Rajput, R102212002,** for partial fulfillment of the requirements for the award of the Degree of Masters of Technology in Computer Science & Engineering With Specialization in Artificial Intelligence and Artificial Neural Network, to Centre For Information Technology, University of Petroleum & Energy Studies is a bonafide report of the work carried by them under our guidance and supervision.

To my best of knowledge, the literature embodied in this project work has not been submitted to any other University/Institute for the award of any degree or diploma.

Pankaj Badoni
1/5/2014

Mr. Pankaj Badoni
Assistant Professor
Centre for Information Technology
UPES, Dehradun

1/5/14

Dr. Manish Prateek
Associate Dean
Centre for Information Technology
UPES, Dehradun

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in this thesis entitled **"Enhancement In Navigation System By Unique Tagging"** in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Computer Science & Engineering With Specialization in Artificial Intelligence and Artificial Neural Network and submitted in the Department of CIT, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my own work carried out during a period from January, 2014 to April, 2014 under the supervision of Mr. Pankaj Badoni., Assistant Professor, CIT, CoES.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

G.K. Rajput

(GAURAV KUMAR RAJPUT)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

PANKAJ BADONI

Pankaj Badoni 1/5/2014

Date: 29/04/2014

Supervisor name and signature

The Dissertation. Viva-Voce Examination of **Mr. Gaurav Kumar Rajput,** Roll No R102212002 has been held on 23/04/2014.

1/5/14

Official/Department Stamp with signature head.

ACKNOWLEDGEMENT

I wish to express our deep gratitude to my supervisors **Mr. Pankaj Badoni and Mr. Vinay Rao** for all advice, encouragement and constant support they has given me throughout my project work. This work would not have been possible without their support and valuable suggestions.

I am grateful to I2India Private Limited for giving me the platform and supporting technical staff that helped me to successfully accomplish the project.

I am also grateful to **Dr. Manish Prateek, Associate Dean** and **Dr. Kamal Bansal, Director** CoES, UPES for giving me the necessary facilities to carry out my project work successfully.

I would like to thank all our **friends** for their help and constructive criticism during my project work. Finally I have no words to express our sincere gratitude to our **parents** who have shown me this world and for every support they have given me.

Name: **Gaurav Kumar Rajput**

Roll No: **R102212002**

ABSTRACT

A novel concept of custom map-tags and cellular-phone service platform is proposed for countries like India, to enable efficient, automated, route-finding directions for mobile phone users with or without access to internet and efficiently manages last mile directions even in locations that are poorly mapped or with poor signage. The basic concept is to provide a convenient and faster one step/click solution to help locate places even in challenging situations such as but not limited to , when mobile/internet connectivity is not available , where locations are poorly mapped and marked (i.e. where road names are missing or buildings are not numbered properly), where new/temporary construction has changed the directions/map from what is available on the Internet, where the person looking for direction is too short on time or in an inconvenient location/situation to undergo map finding on the mobile phone etc. This solution for the above mentioned problem is enabled through three optional methods ,firstly Short Messaging Service (SMS) on a mobile phone secondly through a custom application designed for a conventional smart phone or tablet and finally through a conventional personal-computer connected to the internet.

The application primarily focuses on getting point to point direction for under mapped location for countries like India. Since most of the places or residential addresses in India are not mapped in the present existing maps, this application provides better way to reach the under mapped location using Google navigation services or by using the internet services.

Keywords:- Fast Map tag, Google navigation, Android

CONTENTS

Topics **Page No .**

List of Figures

Figures **Page No.**

1. Chapter 2

2. Chapter 3

3. Chapter 4

4. Chapter 5

5. Chapter 6

6. **Chapter 8**

List of Tables

Tables **Page No.**

1. Chapter 7

List of Graphs

Graphs **Page No.**

1. Chapter 7

Chapter 1
Introduction

1. INTODUCTION

With changing times, the mobile technology has changed a lot and in the last few years we have seen the arrival of various new kinds of gadgets in the form of smart phone, camera-phone, Android and tablet phones. In fact, the handset industry has turned from simple budget handsets to ultra-modern high end mobile phones. Today's device is almost everything - it is fashionable, innovative, appealing, high-performing, durable, stylish and multi-tasking. Latest gadgets can be used for various purposes like browsing mobile, internet, navigation, playing games, emailing, blogging, messaging and accessing all popular social networking sites like YouTube, Google search, Gmail and more.

Similarly for navigating to different places various smart phone are using different navigation apps that are using different types of maps like Nokia maps, Google maps, Apple maps, open street maps etc.

Above said maps are used all over the world for getting direction to different places. In India also the sane maps are used for getting directions. But the problem with India is that, it is a developing country and not all places of the country are marked in Google or any other maps. Unlike other developed country where all places are mapped in the existing maps.

So in India if one searches for commercial or famous places in the map it gives the correct location and correct direction to reach the destination. But if a person enters a full address in the map and searches the person will get the result as "place not found" or it will give the direction to any other place, it will not give exact location of the address. There are also some cases we get the difficulties on existing mobile application maps:-

- Typically the user has to first launch the mapping application from the mobile phone, which in itself takes some time to load.
- The next step is to enter detailed location address, which is also cumbersome and prone to spelling errors especially on a mobile phone keyboard. In non-English native countries/locations, names can often have multiple and difficult spellings, making the task even more challenging.

- The third step is to select from various options or the closest option that the mapping application can find, which itself can take a bit of waiting time.
- The fourth step is to set starting location either through GPS (current location) or by laboriously entering the starting address with similar issues as in the previous step.
- The fifth step is to wait for the directions to load. This is often the longest step as it requires loading of the map with annotated directions, before the text directions can be reviewed.
- The final and sixth step is to select between map view and step-by-step direction view.

The entire process can take several minutes, may often not work at all and often does not provide the correct destination. In locations such as several Indian cities and villages, the roads/landmarks are not signed/labelled either in the maps and/or in the physical world, making direction finding a real hassle. The Finder has to resort to making calls to the Findee or asking bystanders or both. This whole process wastes time, costs extra money, wastes mobile battery power, and is greatly inefficient. The web-mapping systems in vogue today were developed with a developed country like environment in mind, where broadband mobile internet and labelled maps are taken for granted today. The systems do not fully address Indian requirements.

For all above cases one need some sort of abbreviation type thing for the addresses which are easy to remember and easy to type on mobile phone while travelling also. The solution of this situation is that "TAG" system, also called FAST MAP tag system. By using the map tag the whole address can be replaced by a simple tag which will be convenient to the user and easy to remember also.

So, in this report in continuation to the front end part the goal is to basically deal in database management part (back end) which enables the users to perform situation driven task for example making new unique entry for a new "TAG" metadata, fetching relevant information about specific "TAG", updating an existing "TAG" information. And also to find out the most optimized and simpler way of performing all above tasks as we are dealing with the mobile devices .

Chapter 2
Literature Review

- Research and Study
- Challenges in Mobile Application Development
- Marker research

2. LITERATURE REVIEW

Android is a relatively new platform. It is produced by Google, Inc., and its first release was presented in 2007[1]. Android is installed on many different mobile devices and its users can download Android apps and other content through Google Play service, which replaced the old Android Market [2]. This thesis discusses technologies incorporated in Android application development and how they apply to the research problem. Google claims that “Android powers millions of phones, tablets and other devices.” Phones and tablets are mobile devices that can have Android applications installed on them. These applications are written in Java programming language [3] and they are called mobile device applications or apps. Development techniques for apps are structured sets of Java code focused on implementing particular task that provides content for a mobile device application. Although Java programming language includes a broad variety of topics, this thesis focuses on development techniques required for successful implementation of Android Mobile FAST MAP TAG. However, tools and techniques that have long been refined for creating successful desktop computing environments do not translate well to the mobile environment. The focus of our proposal is on developing the next generation of mobile computing applications, which will incorporate a near-constant internet connection, novel interactions (e.g., multi-touch), sensors, and machine learning (e.g., activity inference) to provide rich, interactive experiences

2.1. Research and Study

Over the past year, Google’s Android and Apple’s iOS have been strong competitors in the Mobile app market. According to comScore, iOS mobile devices captured 30% of the market in February 2014. That's up only slightly from November 2013, despite the introduction of the iPhone on Verizon's network. On the other hand, iOS' biggest competitor (in the eyes of many), Google's Android, has grown 7% since November 2013, and now commands 33% of the smart phone subscriber market in the United State .

Smart phones have surpassed feature phones in number of units shipped, the move being driven by cheap Android devices which took 75% of the smart phone market in 2013, while iOS claimed 18%. And others is still struggling with only 3% market share[4].

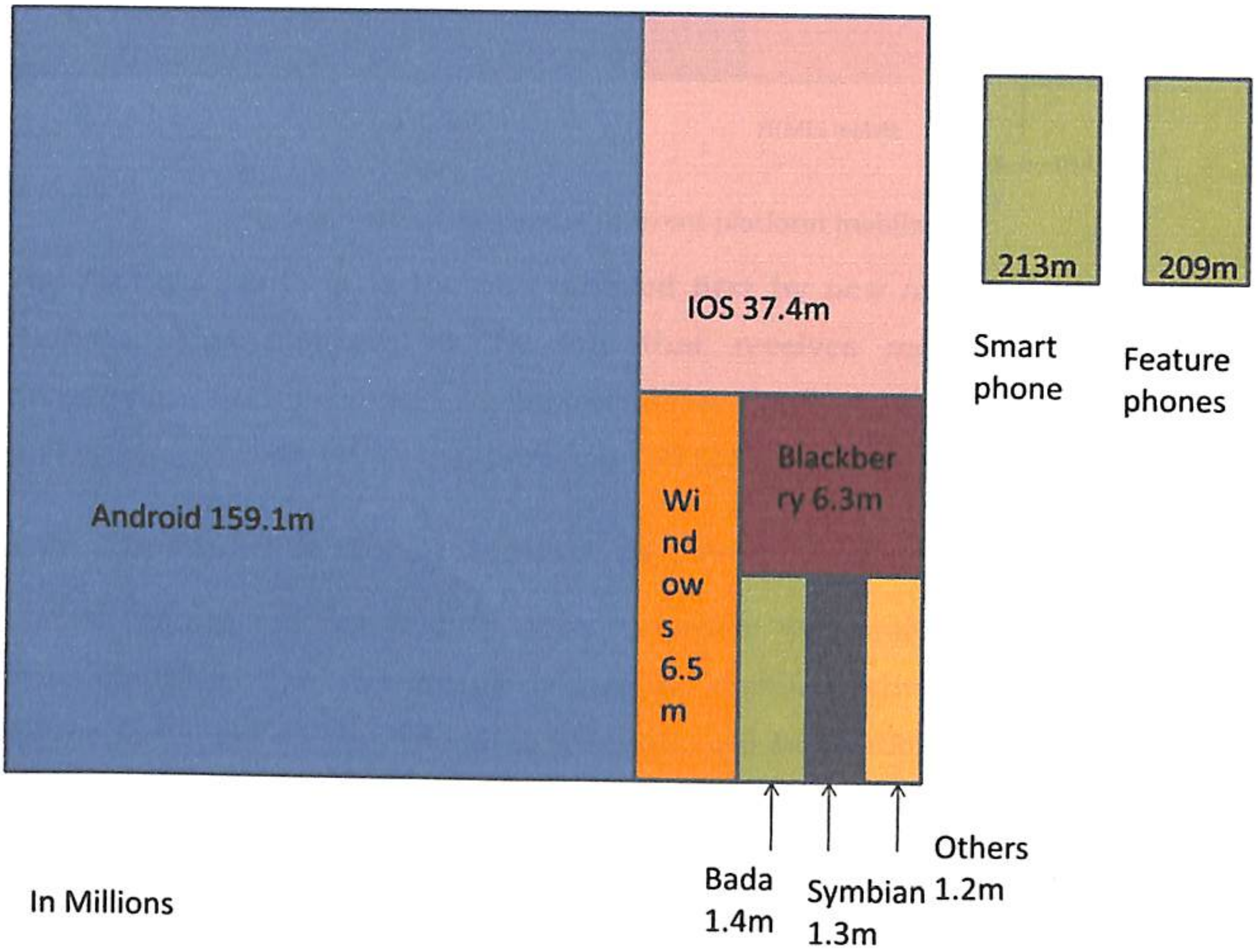


Fig. 2.1 Graph of different mobile users

So the maximum developers are using android platform to build their application on the platform. According to our survey, developers use 2.9 platforms on average and they choose between them based on revenue, reach, delivery Speed, costs, app discovery and development environment. The preferred (primary) platforms are Android (34.4%), iOS (32.7) and HTML(17.3). The graph has been shown below:-

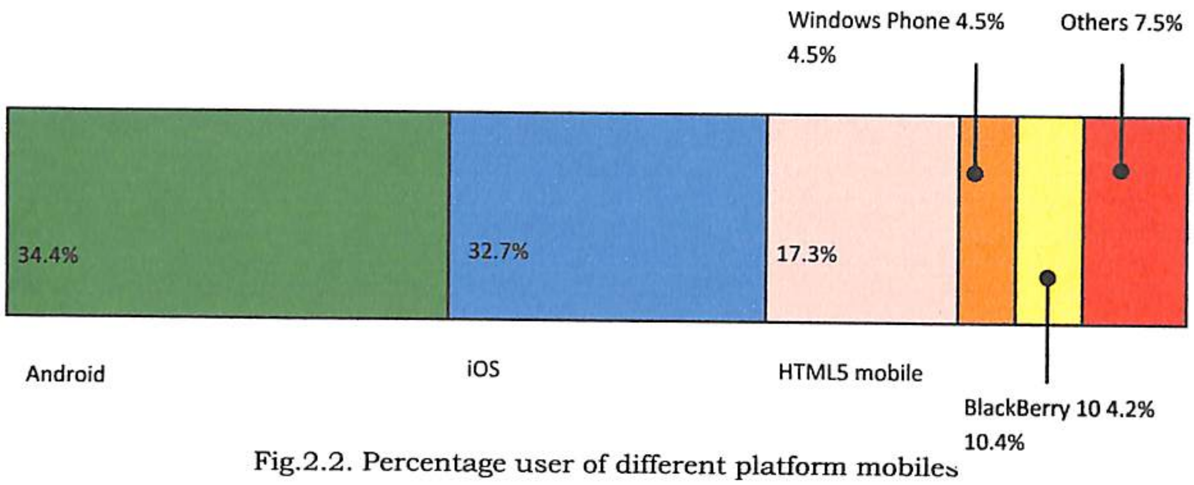


Fig.2.2. Percentage user of different platform mobiles

The Android platform is the one targeted first by new applications or new features. This platform is the one that receives most attention and investment. Android's lead is explained by the fact that new developers prefer this platform (40%) compared to iOS (21%).

2.2. Challenges in Mobile Application Development

Mobile phones are not just "phones" anymore with only "voice and SMS" functionalities. The continuing spread of mobile technology will have a dramatic impact on the lives of individuals and institutions. Convergence of internet and telecommunication technologies is increasing rapidly. However, in the progress of mobile application development, due to complex structure of mobile ecosystem, there is a fragmentation in terms of different mobile "Operating Systems", "Screen Resolutions", "Device Models and Capabilities, " and "User Experience". Fragmentation is the word that defines the biggest barrier. Fragmentation increases the cost and the time to develop mobile applications. Figure 2.3 shows the fragmentation/cost of application development [5].

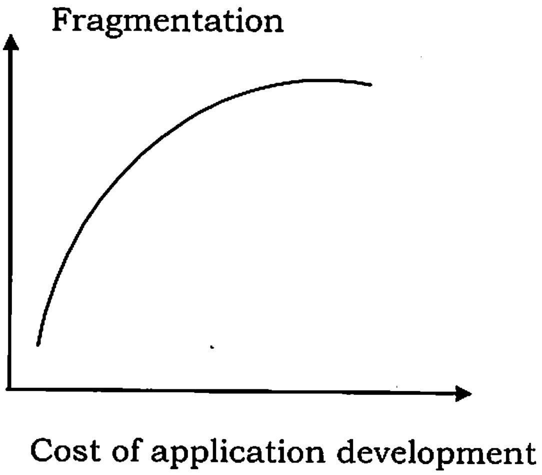


Fig.2.3. Fragmentation and cost of application development

Different device models support different functionalities, such as touch screen, gravity censors, camera flash, etc. Hardware performances also vary between devices. In addition to that, some applications need to support external device functionalities [5]. Screen resolution is crucially important in application development. Ongoing trend is to have bigger screen resolutions for expanded multimedia support, data presentation, and browsing. However, device manufacturers tend to support multi range of resolutions to address the needs of different user segments [5].

User experience, which is simply the way the users behave while using the keyboard, screen, entry- exit functions, number of clicks, etc., is different across device models. Usage scenarios and actions are different between device models and these differences should be realized and taken into consideration in order to deploy successful mobile applications [5].

2.3. Market Research (Existing maps and Analysis)

Due to mapping and direction issues, lot of new systems have been developed to address the challenges. Most of these systems/applications try to avoid using Web based mapping service like Google Maps®, replacing them with custom-created local content. Some of the mobile applications are as follows:-

- **Wonobo:-**

  Provides 360 degree street view of all places in India. It's for people travelling in India having no clue about the destination. Mobile version is yet to be available but it might involve several steps in locating the destination before having the final 360 degree view [6]. It will require broadband mobile internet and a significant amount of download time. It will probably still not address the issue of locating last-mile directions in a poorly annotated world.

- **Location Finder:-**

  Favours Finder, provided they have saved locations beforehand. Location finder helps people save their location with a subject, notice and an expiry date. You will have many such entries which in future with one click, help you to find the same location on Google maps. This still requires Finder to navigate through Google maps to find locations.

- **Quick place finder:-**

  Favours Finders, tapering around Google maps . Quick Place Finder purpose is to simplify place searches via useful and some popular keywords around current location. They have features like popular quick search categories, storing favourite places into local phone database, User managing search radius for filtering the results, user sharing place details via email to her friends etc.

- **Find place:-**

  Helps find anything about places. They find places nearby or by location. It can find restaurants, malls, Entertainment services etc along with the radius distance and Google map attached to it. But it does not provide exact navigating directions to the last mile and search involves several steps. Might require turning on GPS for navigation.

The above said application uses the maps depending upon their flexibility and usefulness. Like Wonobu application uses open street map for getting the destination direction.The solutions listed above, still involve use of a broadband internet connection, to navigate directions, involve multiple steps before obtaining the last mile directions, and in many cases may not even be able to find select locations. The proposed solution enables a unique method for removing all aforementioned challenges faced by current systems, while enabling faster turnaround time. This proposed solution will be very useful for individual users (Finders and Findees) as well as for businesses that involve delivery services e.g., pizza/food delivery, couriers, postal services, bank couriers etc. It is also very helpful for business Findees, especially restaurants and shopping establishments, who might register themselves with Fast-MapTag and get their unique tag to help Finders locate them without having to attend lots of phone calls.

After exploring the steps involved in creating a new user defined TAG using Google Map, the process halts in authorizing the TAG generator and existence of tagged entity which may takes time in units of days so, in this direction the work is to somehow make users available with an intermediate layer which allows users to get immediate availability of a newly created TAG in general to help in reaching to it.

Chapter 3
Insight Frameworks Used

- Android platform architecture
- Anatomy in Android Application
- Google maps API and Geocoding API
- Database management Tool

3. Framework Used for Development

3.1. Android platform Architecture

Android is an open-source software platform developed by Google, for mobile app development on devices powered by the Android OS. It is a complete software stack that provides all the middleware needed to run end-user applications on mobile devices such as: device drivers, OS, core libraries, an optimized virtual machine, Java Native Interface (JNI), and a complete Java development environment. This section provides a detailed introduction to the Android framework and describes the platform architecture, execution model, and key concepts pertinent to the design of the Place Me app, which are more generally applicable to other apps as well. As shown in the figure below, the Android software stack is a tiered architecture that consists of 5 principle layers [7].

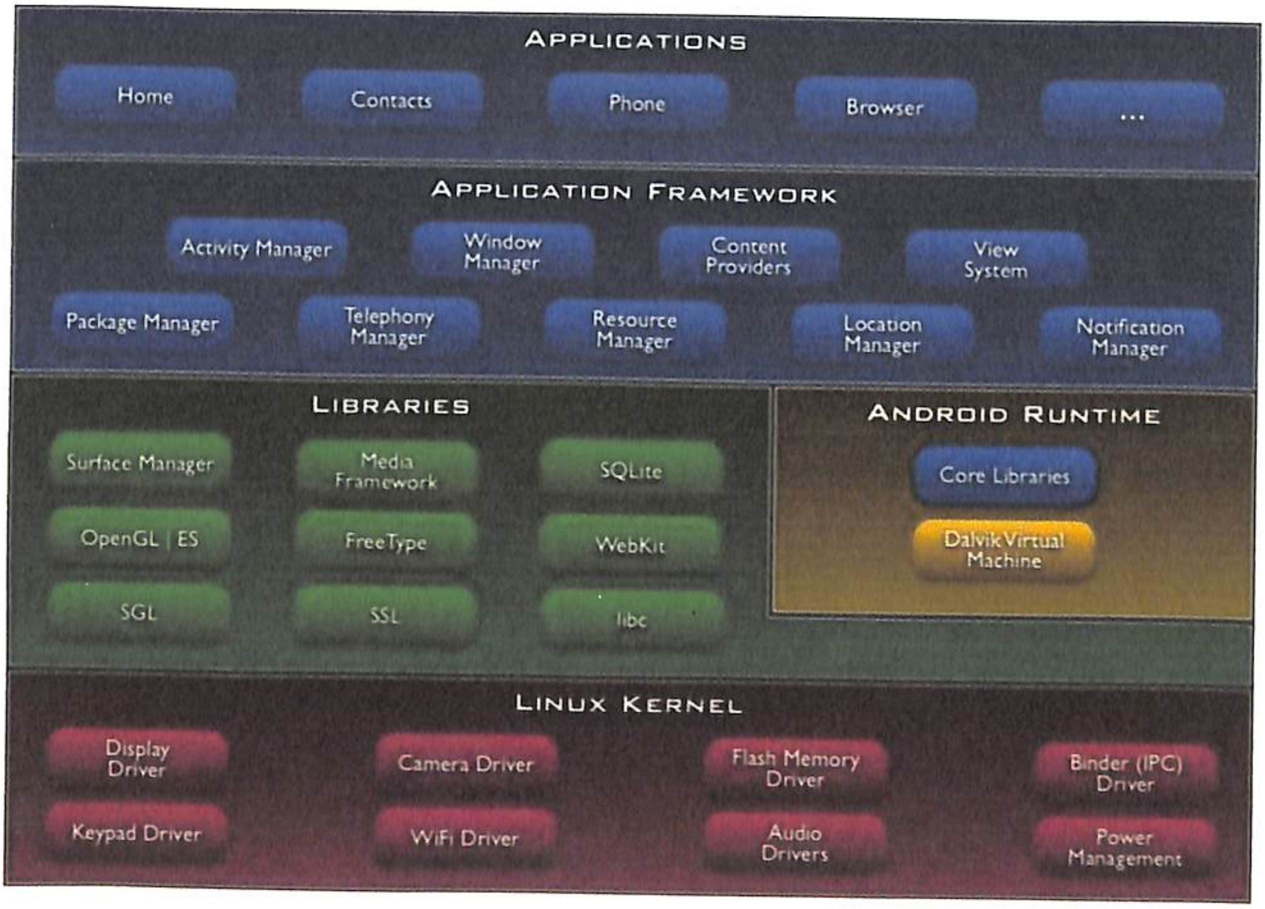


Fig. 3.1 Android Architecture [Courtesy Source: http://developer.android.com/guide/basics/what-is-android.html]

3.1.1. Linux Kernel

Android is built upon Linux 2.6 kernel which serves as the hardware abstraction layer. Linux is used since it provides proven and robust, low-level system infrastructure components such as memory and process management, security, network stack and hardware driver model. Original Equipment Manufacturers (OEMs) can thus bring-up Linux on their system and have the drivers running before loading the other components of the stack [7].

3.1.2. Libraries

On top of the kernel layer are the native libraries written in C and C++, which provide most of the real power of the Android platform. The surface manager is responsible for composing, coordinating and rendering surfaces on the screen from windows owned by different applications, running in different processes in tandem, and ensuring the pixels show up correctly on the screen. OpenGL/ES and SGL are the core 3-D and 2-D graphics libraries respectively. The 3-D graphics can be accelerated in hardware if a 3-D chip is present. Most of the applications commonly use 2-D graphics; however, the platform allows combining 2-D and 3-D graphics as well [8]. The media framework provides all of the audio and video codecs responsible for delivering a rich media experience.

3.1.3. Android Runtime

On the same level as the libraries discussed above is the Android runtime, which is designed for running Java programs in resource constrained, embedded environments with limited computational power, battery life, and memory. One of the main components of the Android Runtime is the Dalvik virtual machine [9]. The Dalvik VM is an optimized byte-code interpreter for efficient byte code execution on small-scale processors used in mobile devices. The Java class and JAR files are translated into ".dex" 25 files at build time, for execution on the Dalvik VM [10].

3.1.4. Application framework and applications

This layer consists of a set of tools and APIs written in the Java programming language which are used by the application developers. Different framework manager is responsible for different activities. And the top-most layer in the stack is the Applications layer. All applications are written in Java and used the same set of APIs provided by the Application Framework. This included applications that are shipped with the phone such as Home, Dialer, Contacts, Browser, etc. as well as those developed by the programmers [8].

3.2. Anatomy of Android App

Briefly discussion the basic components of an Android app and defines the key concepts and vocabulary needed to understand the design of Fast map tag app [11].

3.2.1. View

Views are the fundamental building blocks for creating user-interfaces. A View typically consists of the content visible to the user on the screen such as a button, text field, etc [12]. It is the point of user interaction and handles UI events such as a button press. Views are grouped into a hierarchical structure to form different layout schemes such as a lists, tables, etc which organize the Views into specific pattern for rendering [8]. Layouts and Views are typically specified in XML files.

3.2.2. Activity

An Activity is essentially a piece of user-interface that consists of a set of related tasks a user can do in one screen. For example, a mail app can be divided into 3 basic activities: (1) Mail list Activity that shows all received mails (2) Mail view Activity that shows a single mail message (3) Compose Activity that allows creating and sending outgoing messages [8].

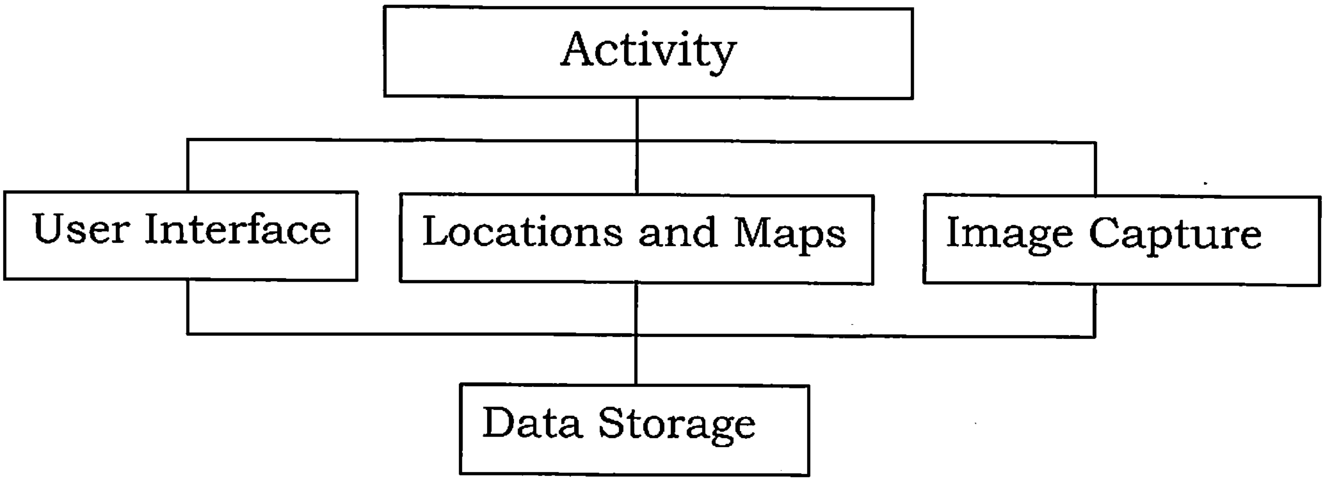


Fig. 3.2 Activity flow

3.2.3. Intents

Intents are the fundamental message passing constructs in Android which allow communication of data and action between and among different system components such as: Applications, Activities, Services, etc. For example, when a new email is received, Intents are fired from the mail listener service to update the mail list screen to show the newly received messages. Apps can also register to receive specific kinds of Intents (generated internally or externally) in order to wake up and execute code when the appropriate Intent is received [13].

3.2.4. Services

Services are background processes launched from Activities that typically perform long-running tasks and have no user interface. For example, a music player can be started as a service and keep playing music while the user may be checking emails [8]. Other Activities or Applications can also bind to the service for performing specific tasks such as pausing, rewinding or fast-forwarding the music.

3.2.5. Data Storage

The data storage options which the Android supports: Shared Preferences, Internal Storage, External Storage, SQLite Databases and Network Connection. Trip tracker uses internal storage, External Storage and SQLite databases for storing persistent application data [14].

By default, the application once installed is stored in the internal storage of the Android system. Android also provides full support to SQLite databases. All databases that are created in the application are accessible by name to any class in the application but none outside [15]. The Android SDK includes a sqlite3 database tool which is required to browse the table contents, run SQL commands and perform other SQL functions. Executing an SQL query returns a Cursor which stores the result set pointing to all the rows found by the query. In this application, relational database is used to create two tables for storing tag names, tag details: location name, location description, image URI and geo points of the location [16].

3.3. Google Maps API and Geocoding API

3.3.1. Google Maps API

Android application can access the location services supported by the device through the classes in the android.location package and the Google Maps external library. The main component of the location framework is the LocationManager [17] system service, which provides APIs to determine location and bearing of the underlying device. Google provides a Maps external library that includes the com.google.android.maps package. The com.google.android.maps package used in this application offered built-in downloading, rendering, and caching of Maps tiles, as well as a variety of display options and controls. The key class in the Maps package is com.google.android.maps.MapView, a subclass of ViewGroup.

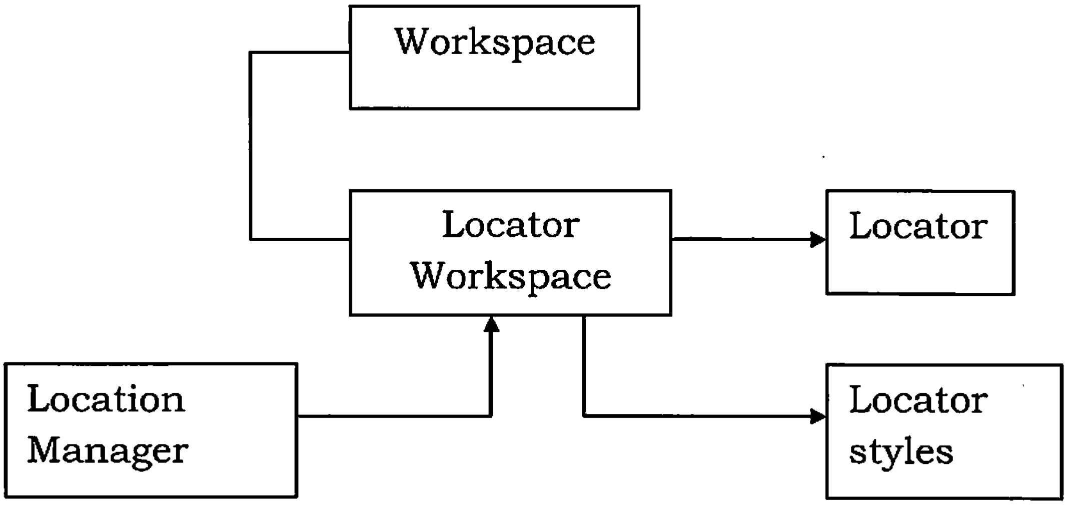


Fig. 3.3 Map API

A MapView displays a map with data obtained from the Google Maps service. When the MapView has focus, it will capture key presses and touch gestures to pan and zoom the map automatically; including handling network requests for additional maps tiles. It also provides all of the UI elements necessary for users to control the map. A LocationListener [18] is the interface implemented to receive location updates. To use Google Maps in the application, a Maps API key had to be obtained to register with the service and Android system had to be notified that the application wishes to implement the add-on Google APIs which are external to the base APIs. This was done by using the uses library element in the Android manifest file, informing Android that the application used classes from the com.google.android.maps package.

3.3.2. Geocoding API

Geocoding is the process of creating geometric representations for locations (such as point features) from descriptions of locations (such as addresses). It is the process of assigning locations to addresses to that they can be placed as points on a map, similar to putting pins on a paper map, and analyzed with other spatial data. The process assigns geographic coordinates to the original data, hence the name geocoding. It is also called address-matching. In a typical geocoding process, the data list might include an address like 508 W. 5th St. and a street centerline GIS data layer would have a street segment corresponding to the 500 block of West 5th Street. The result of geocoding would be a point placed somewhere along the even-address side of that street segment [19].

Two sets of data are needed for the geocoding process - the **address data** that you want to place on a map, e.g., a list of addresses, and the GIS data layer that you will use as the geographic **reference layer**, e.g., a city's street centerlines layer or a parcel address point layer. Both data sets need to be prepared prior to geocoding.

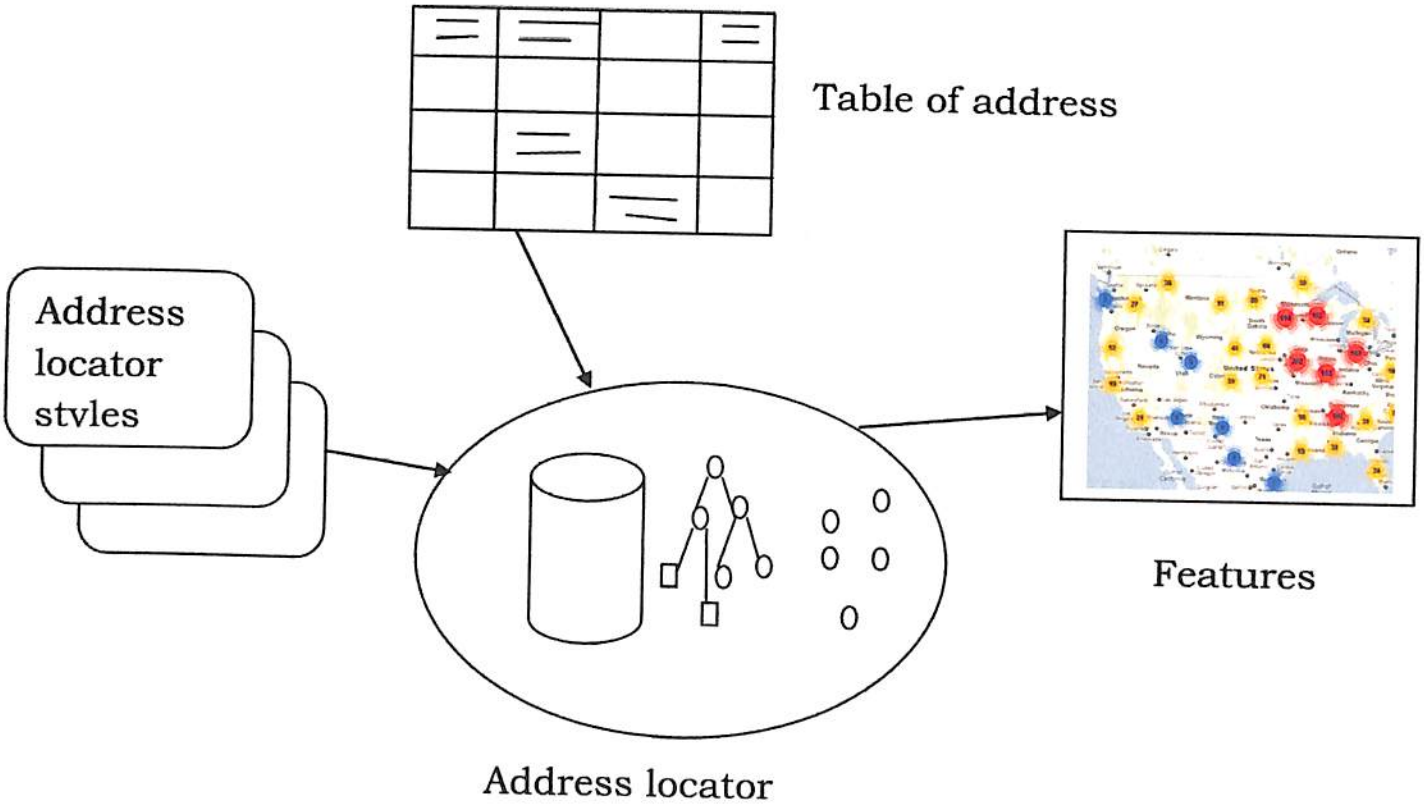


Fig. 3.4 Geocoding Process

3.4 Database Management Tool

Database Management tool is a software system that facilitates the process of defining, constructing, manipulating, and sharing database among various users. This software is mainly used for creating and maintaining the database.

As considering the architecture of the application both frontend as well as backend are equally weighted for providing a well organized and compiled system. So, database management tool plays a vital role in coupling the application modules together and to work smoothly.

For this application, the database management tool used is MySQL. MySQL is an efficient tool for database management, specifically for creating and manipulating from the database. Keeping the application functionality and different use cases into account the frontend is moulded according to the application and working scenario utilizing/consuming data.

Major characteristics of database approach.

- Most importantly self-describing nature of database system.
- Provides insulation between programs and data.
- Supports multiple views of the data.
- Gives you the facility of sharing of data and multiuser data transaction.

Advantages

- To control redundancy.
- Restricting unauthorized access
- Persistent storage.
- Provides backup and recovery.
- To represent complex relationships among data.

- ✓ **Data Definition Language** (DDL), used by the DBA and by database designers to define schemas. The DBMS will have a DDL compiler whose function is to process DDL statements in order to identify descriptions of the schema constructs and to store the schema description in the DBMS catalog.

- ✓ The DBMS provides a set of operations or a language called the **Data Manipulation Language** (DML) for these purposes.

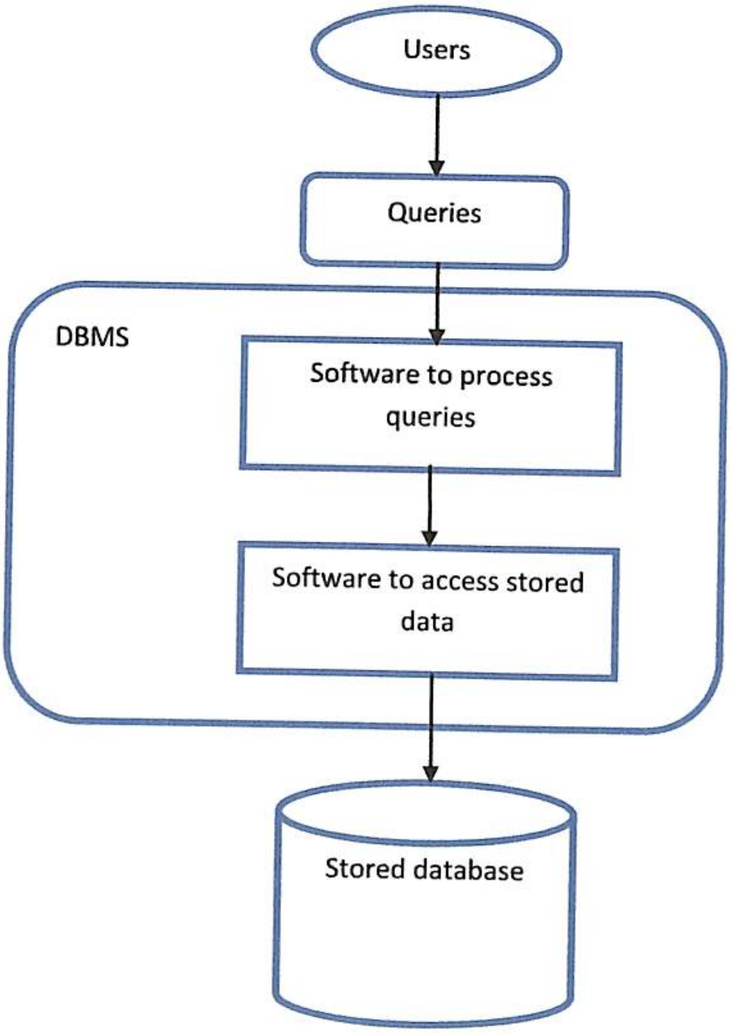


Fig. 3.5 A simplified database system environment.

Chapter 4
System Modules

- Fast Map Tag system
- Fast Map Tag modules

4. APPLICATION SYSTEM AND MODULES

4.1. The Fast Map Tag System

The system can be divided into the following modules and sub-systems namely:

1. **Fast-MapTag Website** a custom-created website with the following modules
    - Fast-MapTag Creation and Editing Module
    - Fast-MapTag Based Address Search Module

2. **Fast-MapTag Mobile Application**
    - Fast-MapTag Creation and Editing Module
    - Fast-MapTag Based Address Search Module
    - Find-Me-Now Module – for instant 1-click location messaging without knowledge of location or having to provide additional details

3. **Fast-MapTag Web-Server** with following components
    - Map Management Module – to manage queries to internal or external mapping and direction creation service such as Google Maps®.
    - Cloud Database Module – to store all Fast-MapTag relation information

4. **SMS-based Direction Query Service** with following components
    - SMS Server – to manage queries to internal or external mapping and direction creation service such as Google Maps®.
    - Business logic unit
    - Network Interface with Fast-MapTag Sever

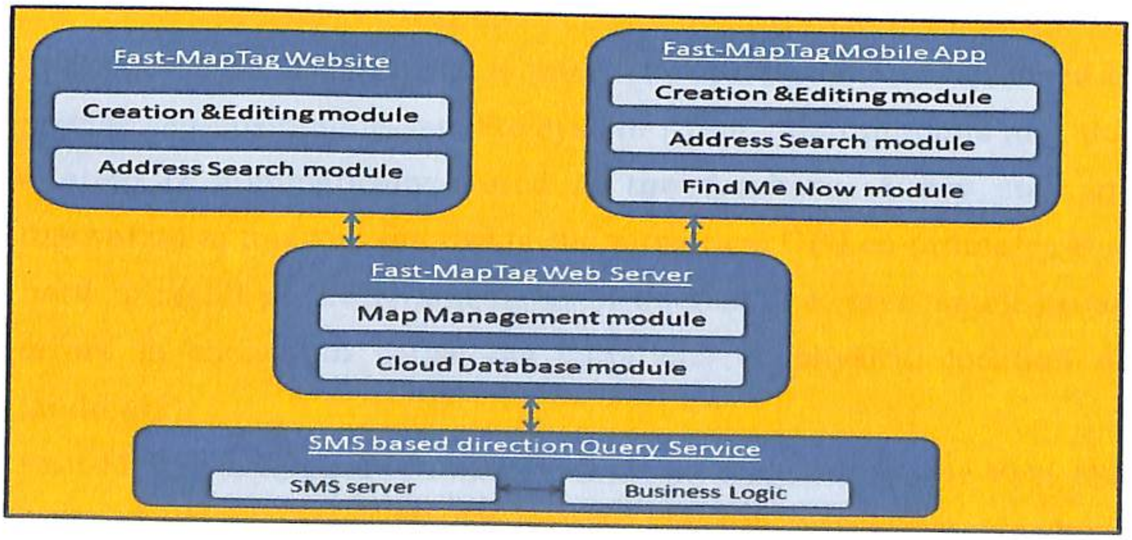


Fig.4.1. Fast map tag system

4.2. Fast Map Tag Module

4.2.1. Fast-MapTag Website

a. Fast-MapTag Creation and Editing Module

Findee can create their unique location tags using the Fast-MapTag application via internet. The user who needs to create a Fast-MapTag for their address location using Fast-MapTag application via internet first navigates to the Fast-MapTag website using a conventional web browser. The creation process is described as below:

1. The Fast-MapTag creation module comprises of
   a. User Direction input module,
   b. Interface with Fast-MapTag database,
   c. Interface with web based mapping service providers,
   d. Business logic and
   e. Output module

   Using this Fast-MapTag Creation Module, the Findee creates a Fast-MapTag for their location by entering the full address in the input module and locating their site using one or more underlying web-based mapping services (e.g. GoogleMaps®). The Findee is also given the option of suggesting a natural language description for their Fast-MapTag e.g. "Anil Gupta Koramangala Home". These inputs get translated to a custom maptag format using business logic with the help of web based mapping service

2. The Fast-MapTag so created is internally and automatically linked to the precise geo-positioning-satellite system (GPS) co-ordinations and this co-relation is automatically stored in the database. A key step in this innovation is to avoid the use of the numerical GPS co-ordinates that are hard to recall precisely and no fault-tolerant (i.e. even small numerical errors in them can cause big variations in physical location of the landmark).
3. Fast-MapTags application helps people to create a unique Fast-MapTag using their conventional English language or in their vernacular language

which can be easily communicated to others in a much faster way than providing their complete address every time.

4. The business logic includes the process of creating corresponding navigation direction by connecting internally to one/more web based mapping services like Google maps and storing it in database, doing redundancy check on maptags, validating the user input address, pushing the valid maptags to Google maps etc.
5. Creation module recommends Findee to provide meta-data including but not limited to navigation landmarks, driving assistance directions and photographs, individually identified with multiple incoming directions.
6. Database management module will store the user's Fast-MapTag, its corresponding physical address, any relevant landmark(s) provided by Findee, any direction-finding metadata provided by Findee and any optional pictures uploaded by Findee.
7. The output module display Fast-MapTag in a simple conventional English or vernacular language which is easily to read and remember.
8. Creation Module is login and password protected by the Findee to allow only the said Findee to change/manage the maptags and associated meta-data.

b. Fast-MapTag Based Address Search Module

The user who needs direction to navigate from one location to another can get navigation directions through an internet connected communication device, if they have the desired destination maptag. The user first navigates to a custom-created Fast-MapTag service website using a conventional web browser. The Finding module which is associated with finding the directions is described as below:

1. The Finding module comprises of user

a. Fast-MapTag input module,

b. Interface with Fast-MapTag database search,

c. Business logic module

d. Interface with Mapping Service module

e. Output module

2. The Finder inputs Fast-MapTag in a simple conventional English or vernacular language which is a definite hard-coded format. The Fast-MapTag is first tested via Business logic for possible entry errors such as spelling errors and thereupon the data is queried from the database to obtain the corresponding physical address. The address is sent to Mapping Service module to obtain navigation directions and associated details including destination map. These details are displayed using the Output module.
3. The output module will display the user with clear navigation path for the destination in a short format which is easily understood by the user.

4.2.2. Fast-MapTag Mobile Application

a. Fast-MapTag Creation and Editing Module

The user who needs to use Fast-MapTags application via mobile device will first install Fast-MapTagApp in their communication device like mobile phone or tablet, which supports installing mobile based application in the device. The creating and editing module is same as that of creating tags in fast map tag website.

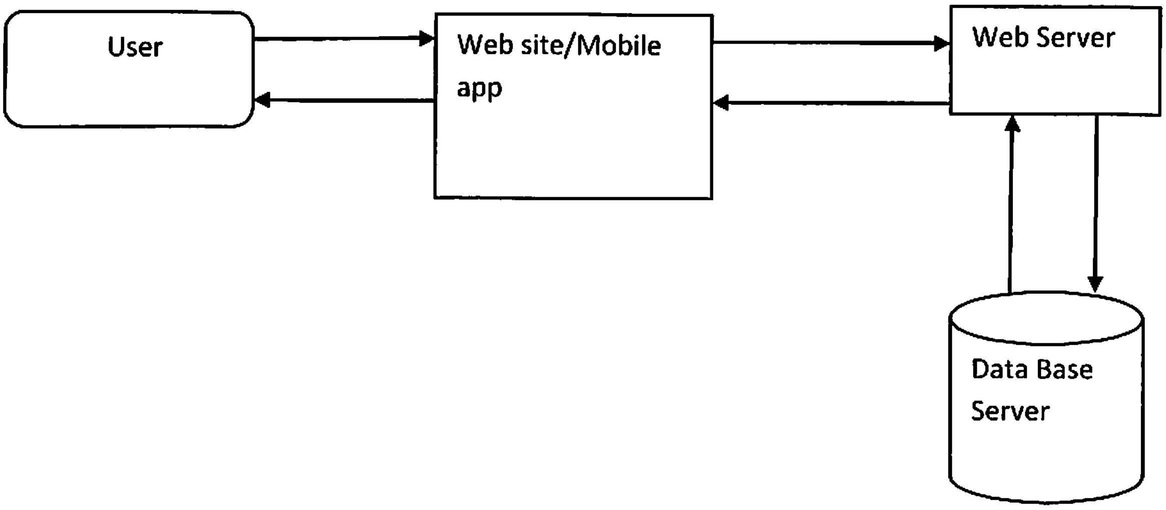


Fig. 4.2 Fast map tag creation module

b. **Fast-MapTag Based Address Searching Module**

Finders can obtain navigation directions for their destinations using Fast-MapTag services through a mobile-phone application [20]. The direction finding process involves the Finder interacting with the Fast-MapTag Finding Module as described below.

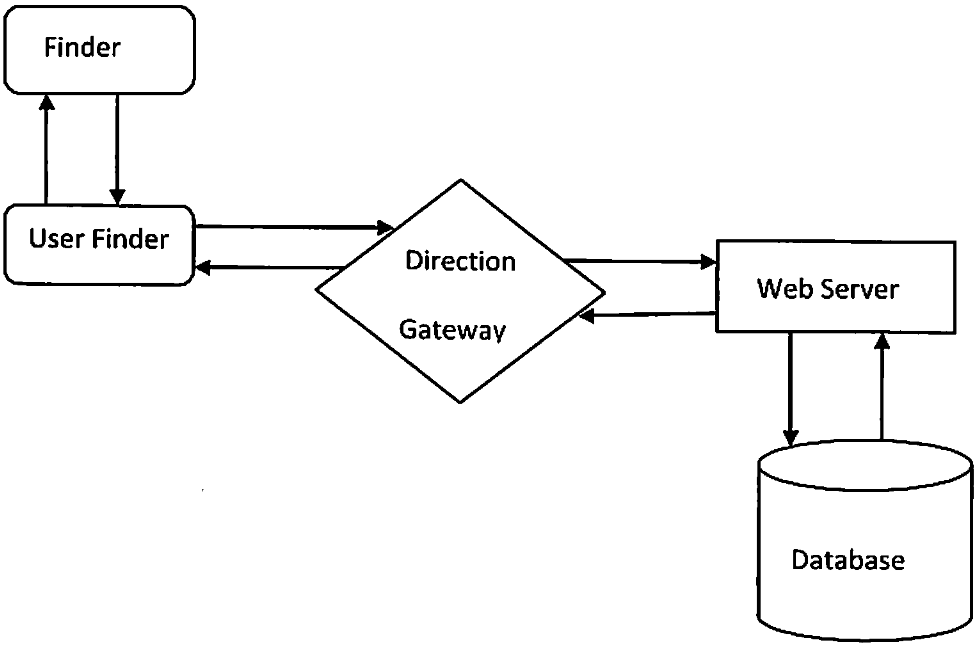


Fig.4.3 Fast map tag searching module

The Finder can get directions through mobile device if they have the desired destination Fast-MapTag. The Finding module which is associated in finding directions is described as below:

1. The user enters the destination Fast-MapTag in the Input Module, which in turn queries the Database Search Module to retrieve the custom directions. These directions are processed through the Business Logic Module and displayed using the Output Display Module.
2. The user can enter the starting location as present location or by using a different fast map tag.
3. The Input Module validates the Fast-MapTag, clarifies and/or corrects any errors or incomplete details and thereon, pushes the Fast-MapTag to the Database Search Module for querying corresponding navigation path.

4. The output module will display the user with clear navigation path for the destination in a short format which is easily understood by the user.
5. To economize the service time, extend usage/battery of the mobile device, and other Finder-driven customizations, the Address Finding Module enables several short-cuts and options as follows:

a. Optional download of text-only directions, image-based map-data, meta-data, landmark data, Findee provided photographs etc. Default options can be adjusted by Finder based on their preferences.
b. Optional storage of recently used Directions, including Source and Destination pairs, so that no processing is required to review the record.
c. Optional storage of recently used Destination or Source Fast-MapTag much like in an address-book form. This allows quick retrieval for finding directions in future.
d. Save last direction or activity on default screen so as to keep it visible until user removes it
e. User preferences for not storing any history for privacy or other reasons.

4.2.3. Fast-MapTag Web Server

a. **Map Management Module** – to manage queries to internal or external mapping and direction creation service such as Google Maps®.
b. **Cloud Database Module** – to store all Fast-MapTag relation information

a. Map Management Module

This module interfaces with one or multiple conventional web-based or similar mapping services such as Google Maps® and can provide one or more of the following but not limited to these information

- Map details of destination
- Point to point directions from a source point to a destination point
- Satellite images of the desired location, as an option
- Traffic information
- Landmark information

b. Cloud Database Module

This module stores the Fast-MapTag and associated data required by the Fast-MapTag Server. This includes, but not limited to the following featuers:

- Stores all user account details
- Stores all Fast-MapTags – both user created, system created and public
- Stores all temporarily created maptags
- Stores all user provided meta-data associated

4.2.4. Fast-MapTag SMS-based Address Search Service

Even if the user does not have or wish to use the Fast-MapTag mobile application or if the user is using a phone which does not support the same, the Finder can get directions through their mobile device if they have the desired destination Fast-Map Tag. This system will have 2 sub-components

i. Fast-MapTag SMS Server
   - The system consists of an SMS Server that receives SMS messages sent on a pre-defined phone number.
   - The SMS Server passes on the information to the Business Logic Unit

ii. Business Logic Unit
   - This module interprets the request before sending a query to Fast-MapTag Server
   - The SMS text is converted into a data-query in the form required by the Fast-MapTag Server

iii. Network Interface with Fast-MagTag Web Server
   - The system is networked over the internet, with a Fast-MapTag Server
   - The Fast-MapTag Server sends a response to the SMS Server, which in turn responds to the Finder looking for details.

Chapter 5

System Process Flow

- Findee's Data Flow
- Finder's Data Flow

5. SYSTEM PROCESS FLOW

The overall concept of creating Fast-MapTag System is discussed in the picture below and described through the following steps:

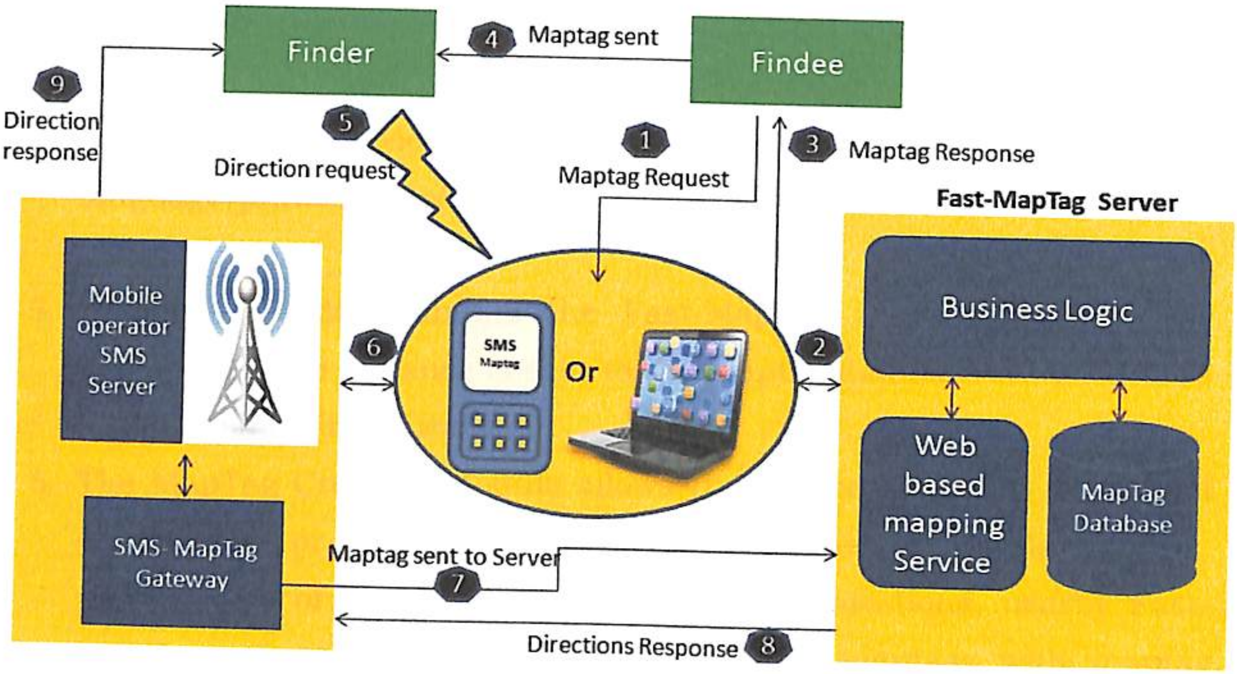


Fig.5.1 Fast map tag system flow

The Fast-MapTag App offer people to create and link their location tags to any web based mapping service such as Google maps. This custom Fast-MapTag is created as a unique and definite code-word or code-phrase that people use to uniquely tag locations that need to be searched via digital mapping services. Without losing the generality, we refer to the people residing in these destinations/ location as 'Findee' and the person who searches for destination/location as 'Finder' in this document.

5.1. *Findee's* Data Flow

1. The Findee first uses the Fast-MapTag Creation Module to create a maptag in Fast- MapTagApp application for which they use a custom-created website using a web browser in one instance else, or their mobile phone to install a custom-created mobile phone application in another instance.

2. The Findee enters his/her desired address in the address bar of the website or the mobile application to create their Fast-MapTag.
3. Fast-MapTag App or web-service is connected to one or more new or existing web based mapping service such as the Google map via the Internet to fetch address map to the Findee.
4. Findee uses the MapTag creation module enables to create a unique tag for their location in manner similar to creating email addresses on free web-mail services such as Gmail. The MapTag is a unique language phrase chosen by the Findee and is either not reserved or already in use.
5. Once the Findee submits the Fast-MapTag in Fast-MapTag App application, the instance is sent to the maptag database which checks for the availability of the tag.
6. The MapTag Creation Module allows the Findee to upload additional information or MapTag - Metadata such as nearby landmarks, photographs of their location and/or nearby locations, nearby Fast-MapTag references etc. while creating their maptags with maptag repository.
7. The custom MapTags so created, can be optionally communicated to 3rd party conventional mapping services such as GoogleMaps® so as to help improve the level of information available with such services.

The Fast-MapTag Creation Module provides for a hierarchical model of providing direction-finding meta-data, which is wholly customizable by the Findee. The core concept is to build a hierarchy of increasing detail and data-size, for helping find the destination. The basic idea is that the Finder can obtain more and more details from an automated system, as long as they have uncertainty about their destination. The system is designed to provide the minimal amount of information that can be delivered most speed and cost-effectively. The process flow for findees is given below

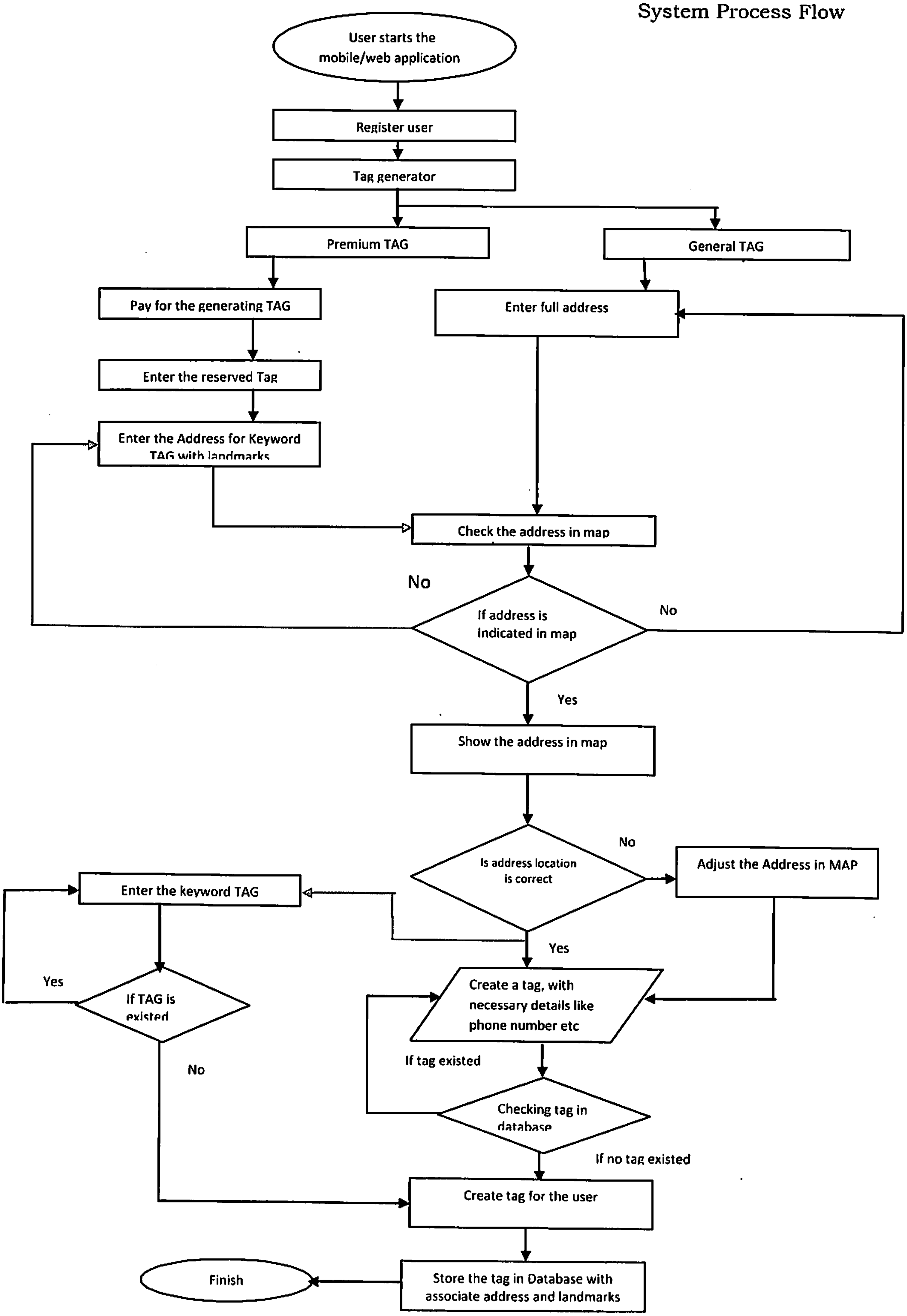

User starts the mobile/web application
Register user
Tag generator
Premium TAG
General TAG
Pay for the generating TAG
Enter the reserved Tag
Enter the Address for Keyword TAG with landmarks
Enter full address
Check the address in map
No
If address is Indicated in map
No
Yes
Show the address in map
No
Is address location is correct
Adjust the Address in MAP
Enter the keyword TAG
Yes
Yes
If TAG is existed
Create a tag, with necessary details like phone number etc
If tag existed
No
Checking tag in database
If no tag existed
Create tag for the user
Finish
Store the tag in Database with associate address and landmarks

Fig.5.2. Findee's Data flow

5.2. *Finder's* Data Flow

1. The Finder who needs to locate a particular Findee obtains the Findee's Fast-MapTag separately (outside the use of this system). The Finder uses the Fast-MapTag Finding Module either on the custom-Fast-MapTag website or through the custom Fast-MapTag mobile phone application or through SMS service on a mobile phone.
2. While using the custom Fast-MapTag Application, the Finder uses the App to request directions to a destination Fast-MapTag, from either the current location or from another user-assigned location.
3. In both above instances (2 and 3), Fast-MapTag server parses the request syntax and retrieves the directions from a combination of conventional mapping services as well as custom-created Fast-MapTag algorithms.
4. In one instance, if only one destination MapTag is provided, the Fast-MapTag provides directions from the current location of the mobile phone being used by the Finder. The location of the mobile phone is obtained through conventional methods such as mobile phone built-in GPS or through Wi-Fi and/or cellular tower location triangulation.
5. In another instance, where 2 Fast-MapTags are provided, the Fast-MapTag Server provides directions from the second (origin) tag to the first (destination).
6. In serving all direction requests, the Fast-MapTag Server uses a standardized shorthand format to minimize the time/size/cost of the message transmitted.
7. In mobile SMS MaptagApp application mode, the Finder sends requests through text inputs instead of using an SMS message.
8. The Fast-MapTag location requests are designed for intelligent and simple user interactions. The Fast-MapTag Server is capable of natural language processing similar to text-based internet searches, where common English spelling errors and order of words can be accommodated without requiring re-entry of complete information. For example if the word "Home" or "Road" is mis-spelt, the system is

designed in an intelligent enough manner to avoid an error or require re-entry.

9. The Fast-MapTag Server is also designed to intelligently and automatically interpret Finder's situational requirements. In one instance the Finder can simply enter one Fast-MapTag to fetch directions from current location to destination. In another instance, if Finder enters two Fast-MapTags the Server will provide directions from first Fast-MapTag location to second Fast-MapTag location. If the Finder sends multiple requests to same MapTag within a short period of time, the Server will interpret it as requirement for successively more details.

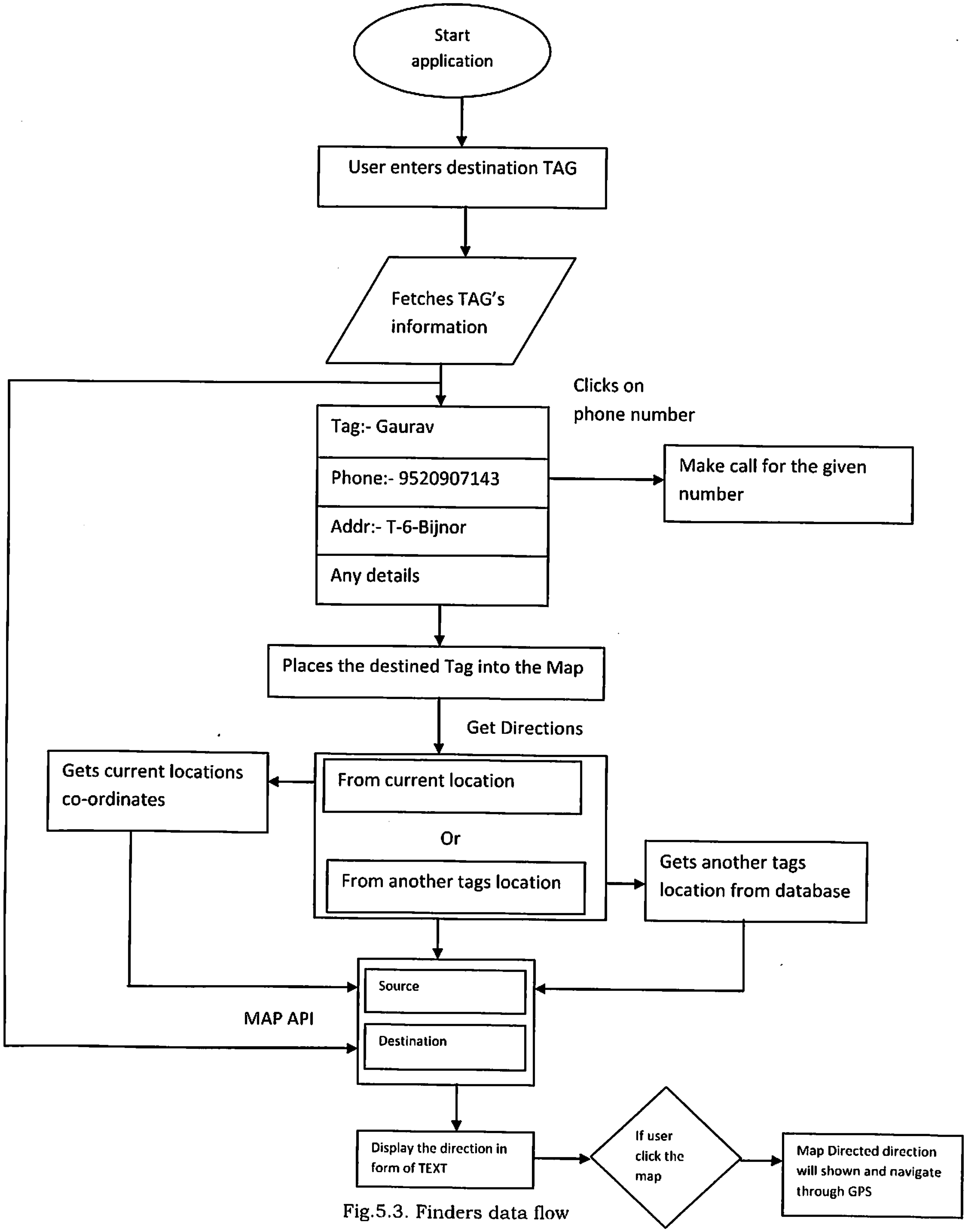

Start application
User enters destination TAG
Fetches TAG's information
Clicks on phone number
Tag:- Gaurav
Phone:- 9520907143
Addr:- T-6-Bijnor
Any details
Make call for the given number
Places the destined Tag into the Map
Get Directions
Gets current locations co-ordinates
From current location
Or
From another tags location
Gets another tags location from database
Source
Destination
MAP API
Display the direction in form of TEXT
If user click the map
Map Directed direction will shown and navigate through GPS

Fig.5.3. Finders data flow

Chapter 6
Integrating & Implementing Modules

- Database design
- Data transaction analysis in Android
- Database implementation

6. DESIGN AND IMPLEMENTATION

6.1. Database Design

After creation of tag from the front end, it is required to store them for future uses. Now database part comes to light. It is the most critical part because in android data flow from application to server and server to application is a tedious process. In android, we can create the database by two ways which given below:

1. SQlite database
2. Server side database

6.1.1. SQlite database: This database is inbuilt in android platform. We can use this database only for the standalone applications. Main drawback of this database is that it is private to application and users of other application can-not access it [14].

Following code for retrieving data using SQLite database:

```
/**
 * displays data from SQLite
 */
private void displayData() {
        dataBase = mHelper.getWritableDatabase();
        Cursor mCursor = dataBase.rawQuery("SELECT * FROM "
                        + DbHelper.TABLE_NAME, null);

        userId.clear();
        user_tagName.clear();
        user_latName.clear();
        user_lngName.clear();
        user_addressName.clear();
        user_phoneName.clear();
        if (mCursor.moveToFirst()) {
                do {
```

```
        userId.add(mCursor.getString(mCursor.getColumnIndex(DbHelp
er.KEY_ID)));

        user_tagName.add(mCursor.getString(mCursor.getColumnIndex
(DbHelper.KEY_TAGNAME)));

        user_latName.add(mCursor.getString(mCursor.getColumnIndex(
DbHelper.KEY_LATNAME)));

        user_lngName.add(mCursor.getString(mCursor.getColumnIndex
(DbHelper.KEY_LNGNAME)));

        user_addressName.add(mCursor.getString(mCursor.getColumnI
ndex(DbHelper.KEY_ADDRESSNAME)));

        user_phoneName.add(mCursor.getString(mCursor.getColumnIn
dex(DbHelper.KEY_PHONENAME)));
                } while (mCursor.moveToNext());
            }
            DisplayAdapter disadpt = new
DisplayAdapter(Tagretrival.this,userId, user_tagName, user_latName,
user_lngName,user_addressName,user_phoneName);
            userList.setAdapter(disadpt);
            mCursor.close();
        }
```

6.1.2. Server Side database: Involving client server architecture in our application we need a server side platform for managing all the server side activities, so for our usage we implemented WAMP which provides Apache, MySQL and Php for Windows. For performing the basic functionalities in our applications we majorly deal with two components as MySQL for database management whereas Php for communication between database and application.

Following code used to request server based functionality:

```
/**
      * Creating new tag
      * */
    protected String doInBackground(String... args) {
       String tag = edit_tag.getText().toString();
       String comments = edit_comments.getText().toString();

       String phonenumber = edit_phonenumber.getText().toString();


       // Building Parameters
       List<NameValuePair> params = new ArrayList<NameValuePair>();
       params.add(new BasicNameValuePair("tag", tag));
       params.add(new BasicNameValuePair("comments", comments));

       params.add(new                    BasicNameValuePair("phonenumber",
phonenumber));



       // getting JSON Object
       // Note that create tag url accepts POST method
       JSONObject json = jsonParser.makeHttpRequest(url_create_tag,
            "POST", params);
```

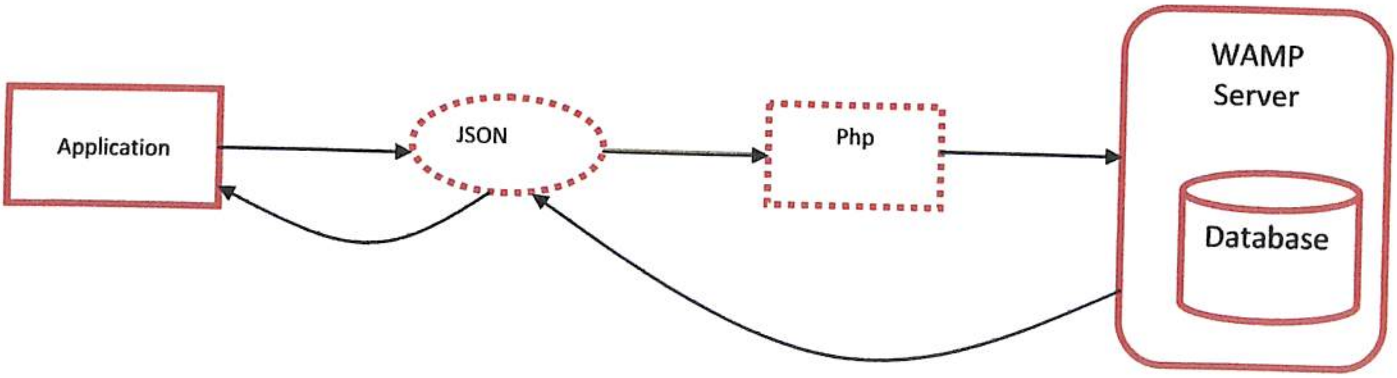


Fig.6.1 Intermediates used during data transaction

6.2. Data Transaction Analysis in Android

For the involvement of data transaction over the network, AsyncTask Class type objects are used. Whenever an AsyncTask object is created, the execution of it behavior is done in a separate UI thread. Then, we need to add the following unimplemented method:

6.2.1. onPreExecute( )

- Throughout our applications, we have displayed a progress dialog here.
- The dialog shows a message while the execution of doInBackground() is complete.

6.2.2. doInBackround( )

- A new thread is created, data to be sent to the server is added as a Name-Value Pair object encoded in a list .
- Using an object of JSON Parser Class, the response method is called, with three arguments being supplied: the URL, method type & the parameters to be sent.
- Response method returns a JSON Object.
- The object is then parsed to get information about:
  - ✓ Successful Execution of script on server,
  - ✓ Error message, if any,
  - ✓ Data such as tag list, location list etc.

6.2.3. onPostExecute( )

- Post execution tasks are performed here i.e. which are supposed to be performed after the execution of this block.

6.2.4. JSON Parser Class

For interchanging the data over network we have used JSON Object(JavaScript Object Notation).It is a lightweight interchange format. It plays the part of mediator between application and server. It has specific method which tells the response of server. The prototype of this method carries three parameters, two of string type and one of list type. String type parameters are basically the address of Php file and other parameter can be either of "GET" or "POST" [22]. Whereas the list parameter signifies the elemental entities used for the application.

- Http Client is instantiated.
- If "GET" method type is selected, the user is trying to get some viable data back from the remote server and as parameters, it is sending some data in params. The query is encoded in an Http Get type object.
- If "POST" method type is selected, the user is just trying to send some viable data to the remote server and as parameters, it is sending some data in params. The query is encoded in an Http Post type object.
- The response of the client's execution of this object is stored in an Http Response type object.
- Response thus received is parsed into entities.
- The content received is extracted from the response object through the getContent() method executed on the response by an Http Entity object.
- The content is an input stream. And using Buffered Reader, the content received, thus, is translated into Strings.
- The strings are finally encoded into a JSON object.

General format for JSON Parser class:

```
public class JSONParser {

    static InputStream is = null;
    static JSONObject jObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    // function get json from url
    // by making HTTP POST or GET mehtod
    public JSONObject makeHttpRequest(String url, String method,
            List<NameValuePair> params) {

        // Making HTTP request
        try {

            // check for request method
            if(method == "POST"){
                // request method is POST
                // defaultHttpClient
                DefaultHttpClient httpClient = new DefaultHttpClient();
                HttpPost httpPost = new HttpPost(url);
                httpPost.setEntity(new UrlEncodedFormEntity(params));

                HttpResponse              httpResponse              =
httpClient.execute(httpPost);
                HttpEntity httpEntity = httpResponse.getEntity();
                is = httpEntity.getContent();

            }else if(method == "GET"){
                // request method is GET
                DefaultHttpClient httpClient = new DefaultHttpClient();
                String   paramString   =   URLEncodedUtils.format(params,
"utf-8");
                url += "?" + paramString;
                HttpGet httpGet = new HttpGet(url);

                HttpResponse              httpResponse              =
httpClient.execute(httpGet);
                HttpEntity httpEntity = httpResponse.getEntity();
                is = httpEntity.getContent();
            }

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        try {
            BufferedReader reader = new BufferedReader(new
InputStreamReader(
                    is, "iso-8859-1"), 8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();
            json = sb.toString();
        } catch (Exception e) {
            Log.e("Buffer Error", "Error converting result " + e.toString());
        }

        // try parse the string to a JSON object
        try {
            jObj = new JSONObject(json);
        } catch (JSONException e) {
            Log.e("JSON Parser", "Error parsing data " + e.toString());
        }

        // return JSON String
        return jObj;

    }
}
```

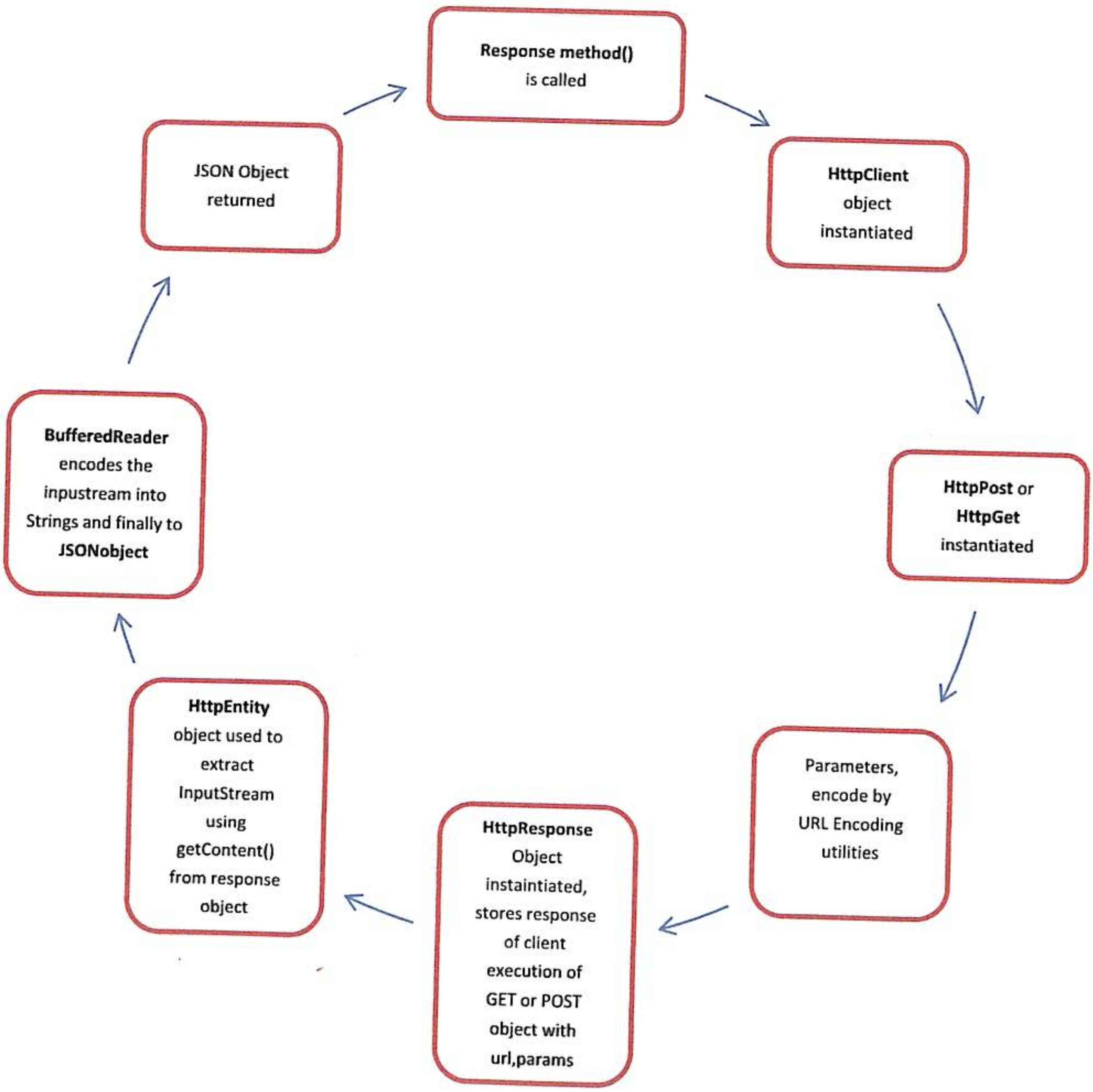

Response method() is called
HttpClient object instantiated
HttpPost or HttpGet instantiated
Parameters, encode by URL Encoding utilities
HttpResponse Object instaintiated, stores response of client execution of GET or POST object with url,params
HttpEntity object used to extract InputStream using getContent() from response object
BufferedReader encodes the inpustream into Strings and finally to JSONobject
JSON Object returned

Fig6.2 Response method() execution cycle in Android

6.3. Database Implementation:

This module stores the Fast-MapTag and associated data required by the Fast-MapTag Server. This includes, but not limited to the following features:

1. Stores all user account details
2. Stores all Fast-Map Tags – both user created, system created
3. Stores all temporarily created maptags
4. Stores all user provided meta-data associated

For efficiently performing the above stated tasks the database module is being divided into below modules:

- Tag Generation Module
- Tag Retrieval Module
- Tag Updating Module

6.3.1. Tag Generation Module

This module basically deals with the functionality of generating user defined Tag's from the front end and then employing database into account by storing the Tag along with its metadata (phone number, address) and provides user an easy application of interaction with database. So in order to implement following are the two approaches:

- Create Tag and save with phone number
- Create Tag and save with address

6.3.1.1. Create Tag and save with phone number

For creating the tag, enter the tag, enter the phone number, enter the comment and save by phone number in database. If tag has already exited, it will prompt that tag has already exited.

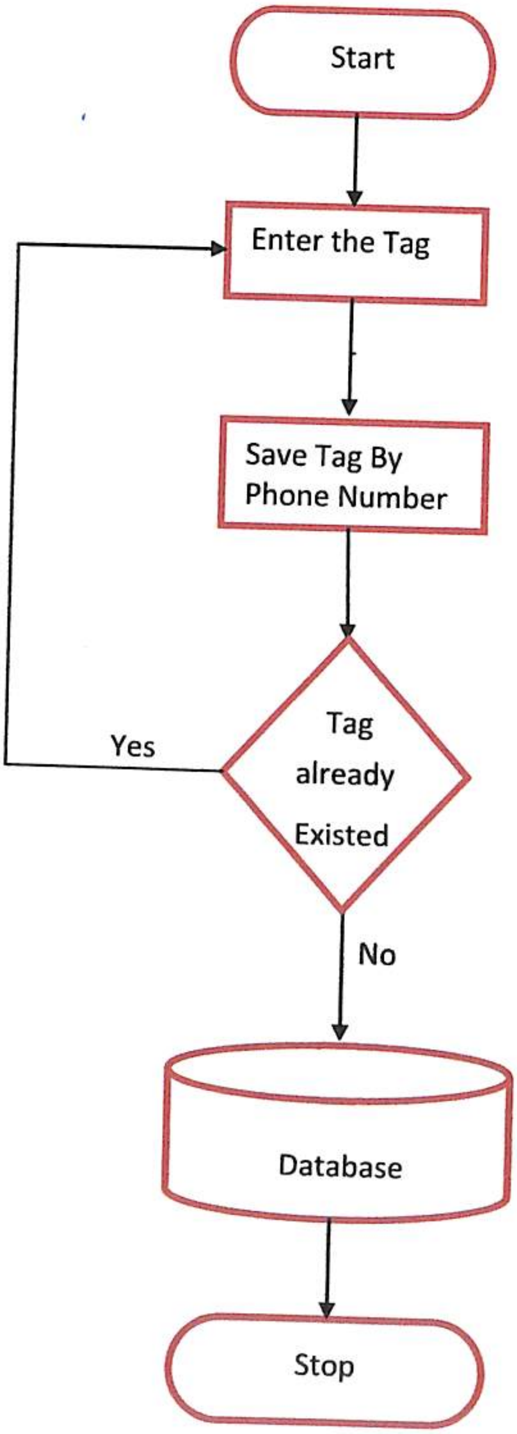


Fig 6.3 Flow chart for creating tag and save with phone-number

Following PHP code responsible for creating new tag with phone number :

```
$result = mysql_query("INSERT INTO products1(name, number, description)
VALUES('$name', '$number', '$description')");

    // check if row inserted or not
    if ($result) {
        // successfully inserted into database
        $response["success"] = 1;
        $response["message"] = "Product successfully created.";

        // echoing JSON response
        echo json_encode($response);
    } else {
        // failed to insert row
        $response["success"] = 0;
        $response["message"] = "Oops! An error occurred.";

        // echoing JSON response
        echo json_encode($response);
    }
}
```

6.3.1.2. Create Tag and save with address

For creating tag, first enter Tag, phone number, comment and save tab by address. Then a map will open and type address and check that it is showing the correct address or not. If it is not, drag the marker, point it to right place. Then press the next button and all information save into the database. If tag is already existed, it shows the message that tag is already existed and tells to user to change tag name. After changing the tag name, save the tag .After saving the tag, it goes into database for checking uniqueness of tag. If tag name is unique, it saves into the database along with its metadata.

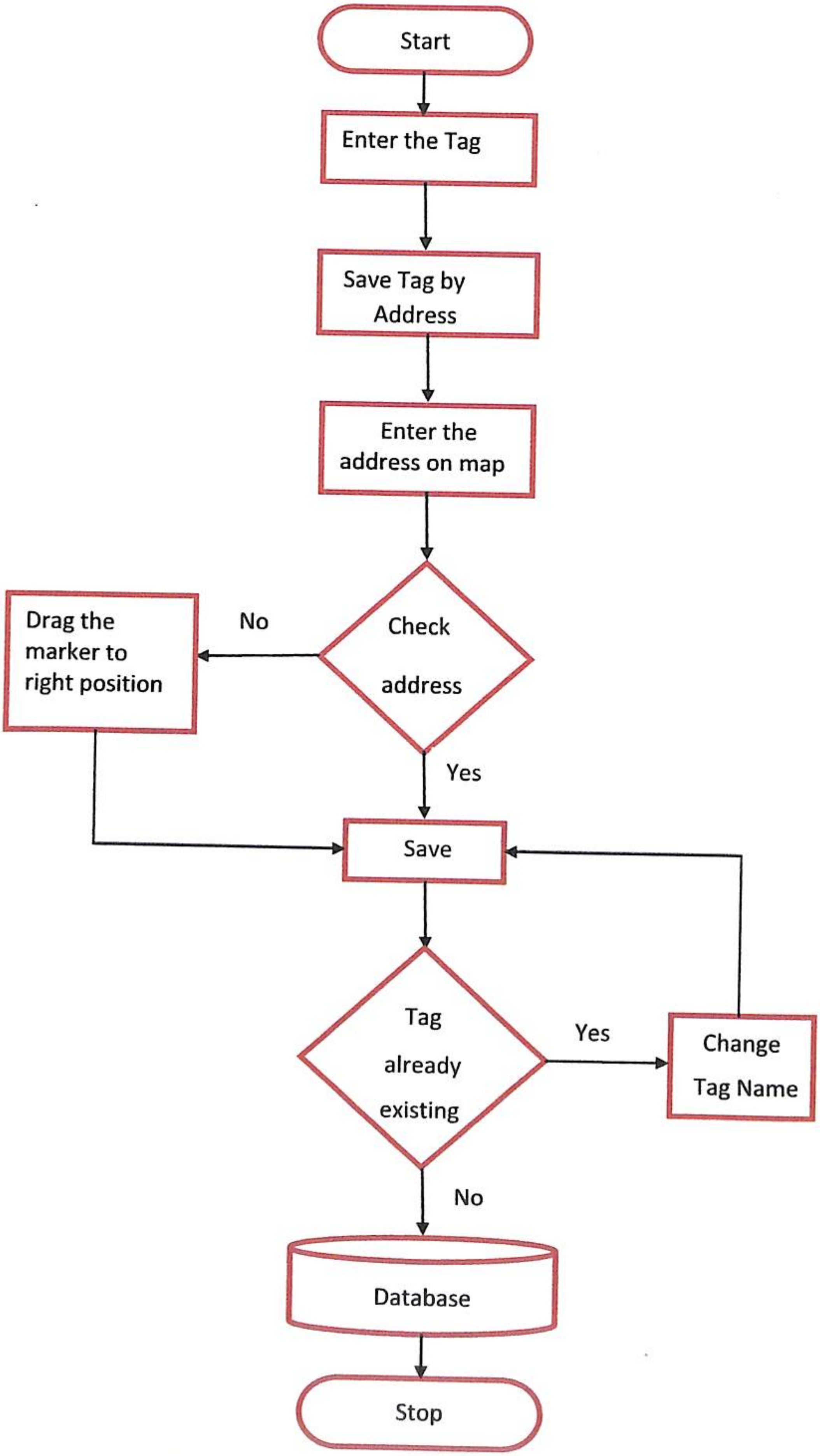


Fig 6.4 Flow chart of creating tag and save with address

Following PHP code responsible for creating tag with address:

```
$result = mysql_query("INSERT INTO products1(name, price, description)
VALUES('$name', '$price', '$description')");

    // check if row inserted or not
    if ($result) {
        // successfully inserted into database
        $response["success"] = 1;
        $response["message"] = "Product successfully created.";

        // echoing JSON response
        echo json_encode($response);
    } else {
        // failed to insert row
        $response["success"] = 0;
        $response["message"] = "Oops! An error occurred.";

        // echoing JSON response
        echo json_encode($response);
    }
}
```

6.3.2. Tag Retrieval Module

This module allows the user to get all information associated with a specific tag. To perform retrieval operations simply enter the tag name and search. It will search in the database and display the information about tag.

Following code used for retrieving a particular code:

```
  $result = mysql_query("SELECT *FROM products WHERE pid = $pid");


   if (!empty($result)) {

      // check for empty result

      if (mysql_num_rows($result) > 0) {


         $result = mysql_fetch_array($result);


         $product = array();

         $product["pid"] = $result["pid"];

         $product["name"] = $result["name"];
```

```
        $product["price"] = $result["price"];
        $product["description"] = $result["description"];
        $product["created_at"] = $result["created_at"];
        $product["updated_at"] = $result["updated_at"];
        // success
        $response["success"] = 1;

        // user node
        $response["product"] = array();

        array_push($response["product"], $product);

        // echoing JSON response
        echo json_encode($response);
    } else {
        // no product found
        $response["success"] = 0;
        $response["message"] = "No product found";

        // echo no users JSON
        echo json_encode($response);
    }
```

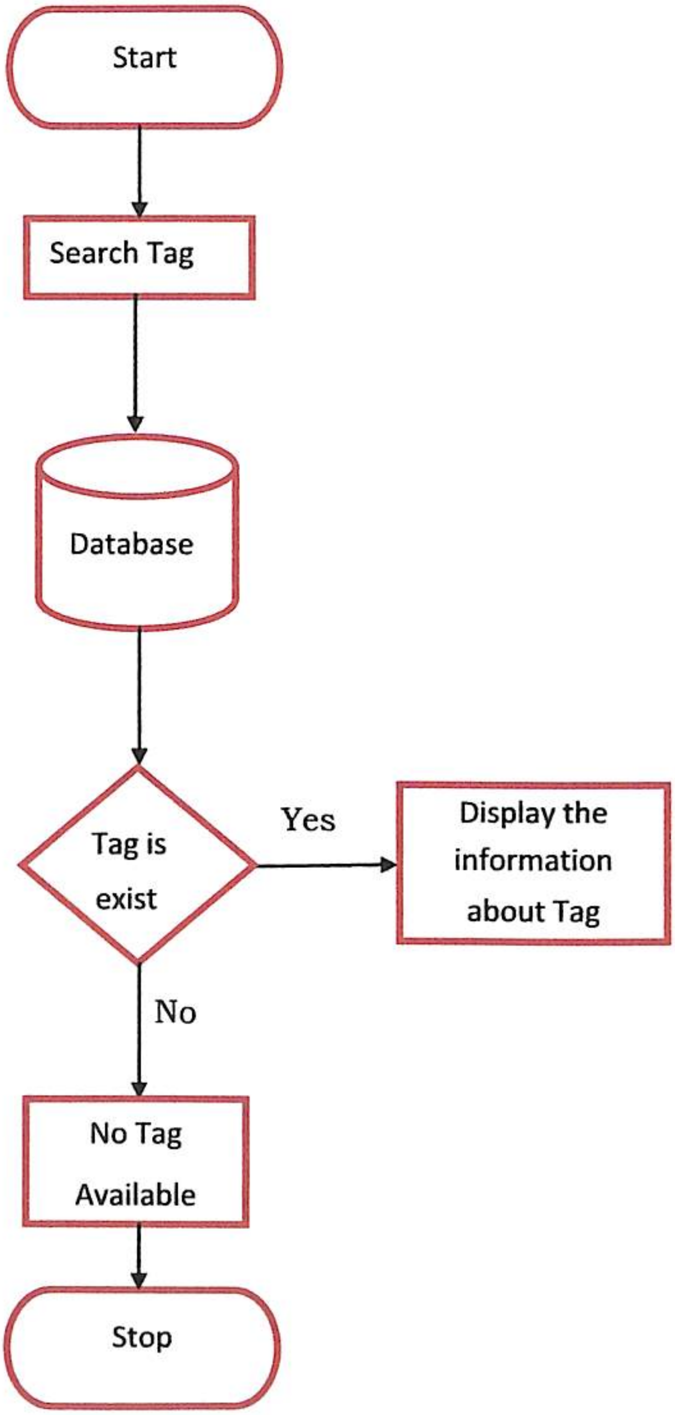


Fig 6.5 Flow chart of Tag Retrieval form database

6.3.3. Record Updating Module

Considering the scenario when user changed their address or phone number and if he wants to update that information associated with tag, then this module helps the user to update tag's information and it will reflect in the database.

Following code used for updating a tags data:

```
  $result = mysql_query("UPDATE products SET name = '$name', price =
'$price', description = '$description' WHERE pid = $pid");

    // check if row inserted or not
    if ($result) {
        // successfully updated
        $response["success"] = 1;
        $response["message"] = " successfully updated.";

        // echoing JSON response
        echo json_encode($response);
    } else {

    }
} else {
    // required field is missing
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";

    // echoing JSON response
    echo json_encode($response);
}
```

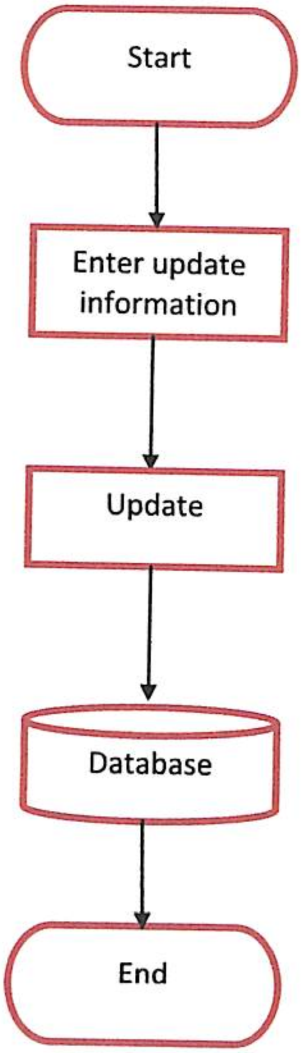


Fig 6.6 Flow chart of record updating information in database

Chapter 7

Usability Testing

- Background
- Usability evaluation of new prototype for mobile device
- Result/Analysis
- Conclusion

7. Usability Testing

7.1 Background

This chapter presents the usability assessment of the prototype proposed. Usability evaluation techniques have been applied here to assess the usability and its improved usability has been presented in comparison with the most popular and widely used social networking applications.

Usability is the key parameter for evaluating any application or a product, as it describes the degree of easiness in using an application or a product. The International Standards Organization (ISO) has defined the usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [26]. In general, usability is about the quality of the interface and it will be determined by how well the user interface is organized to present the available functionalities to the users, in a more convenient and interactive way [27] [28]. There are number of ways to evaluate the usability of the digital libraries, among them Analytical and Empirical techniques are more popular. Analytical evaluation will be carried out by the usability professionals by using the established theories and methods which includes Heuristic Evaluation [29], Cognitive Walkthrough [30] and Claims Analysis [31]. Empirical techniques involve the user for usability evaluation. Various empirical techniques include usability inspection, focus groups, and questionnaires [32], think aloud [33], log analysis [34] etc.., We have selected the empirical methodology for the usability evaluation of the UI presented in this thesis.

7.2 Usability Evaluation of New Prototype for Mobile device

7.2.1 Methodology

Mobile application usability evaluation consists of various steps and methods. We have chosen Empirical evaluation methodology for assessing the usability of the newly built UI system. We have considered the both qualitative and quantitative approaches to assess the results of the experiment.

Empirical evaluation encompasses various usability attributes which have been identified by many researchers for evaluating. Among these, learnability, low error rate, memorability, effectiveness, efficiency and satisfaction are the significant usability attributes for evaluation.

7.2.2 Subjects

Subjects were from the educational institute, India with an open invitation to the identified user groups (students, researchers, faculty members etc.) from all the disciplines. We received 20 responses for this invitation, among 10 are expert users and 10 are novice users. The expert group (6 males / 4 females, ages: 21-45) consisted of two faculty members, two research students and six graduate students. All these expert group members are having the substantial experience with various online social networking, android applications and other contemporary websites on the World Wide Web. The novice group (8 males / 2 females, ages 21-45) consisted of one faculty member, three research students and six graduate students from the School of Languages and School of Arts. This entire novice group is consisted either with the no experience or inconsiderable experience with the online searching and other World Wide Web services.

7.2.3 Experimental design

Four sessions were administered in a sequence during the experiment: pretest, training, formal tasks, and post-study. Pretest session was provided a brief over view on the present experiment process to the participants and the subjects were told that they would be interrupted between questions to allow the experimenter to ask some questions related to the experiment. During the training session, a demonstration was done on different applications to the participants. The dataset used in the training was not related to the formal tasks.

After the training session, subjects were required to answer a set of 10 questions (see Appendix-A). These questions were developed by the present researcher to ensure that the questions did not unduly favor any one product and that users would need to explore the provided application fully to find the answers.

All subjects were tested on the same configuration of the systems with the same network connectivity at our development place. Subjects were then given a set of tasks and asked to attempt all questions in the presented order. The subjects were encouraged to verbalize their thoughts.

After five tasks, subjects took short break and then proceeded as before on the second interface. Order of presentation was counterbalanced across all the subjects with half the experts and half of the novices starting on the Standard applications, and the other half of both user groups starting on the new prototype. Order of questions was randomized for each subject.

After the completion of this task, the questionnaire for tasks was distributed among the subjects, so that they could describe their satisfaction level for accomplishing the tasks.

7.2.4 Usability Evaluation Questionnaire

The usability evaluation criteria questionnaire was formulated based on the various established questionnaires [27] [35] [36] to explore different factors involved in mobile application usability evaluation. This questionnaire contained information that gave the authors quantitative results concerning usability factors such as simplicity, visibility, memorability, learnability, error prevention, effectiveness, efficiency, interactivities and satisfaction. The copy of the questionnaire is attached in Appendix-B.

7.3 Results/Analysis

Results and analysis are presented for both the experiment and usability questionnaire. Two-sample t-test has been employed to determine the impact of the interface on usability.

The following tables summarize the user ratings for usability questionnaire in percentage wise for Standard application as well as for New prototype.

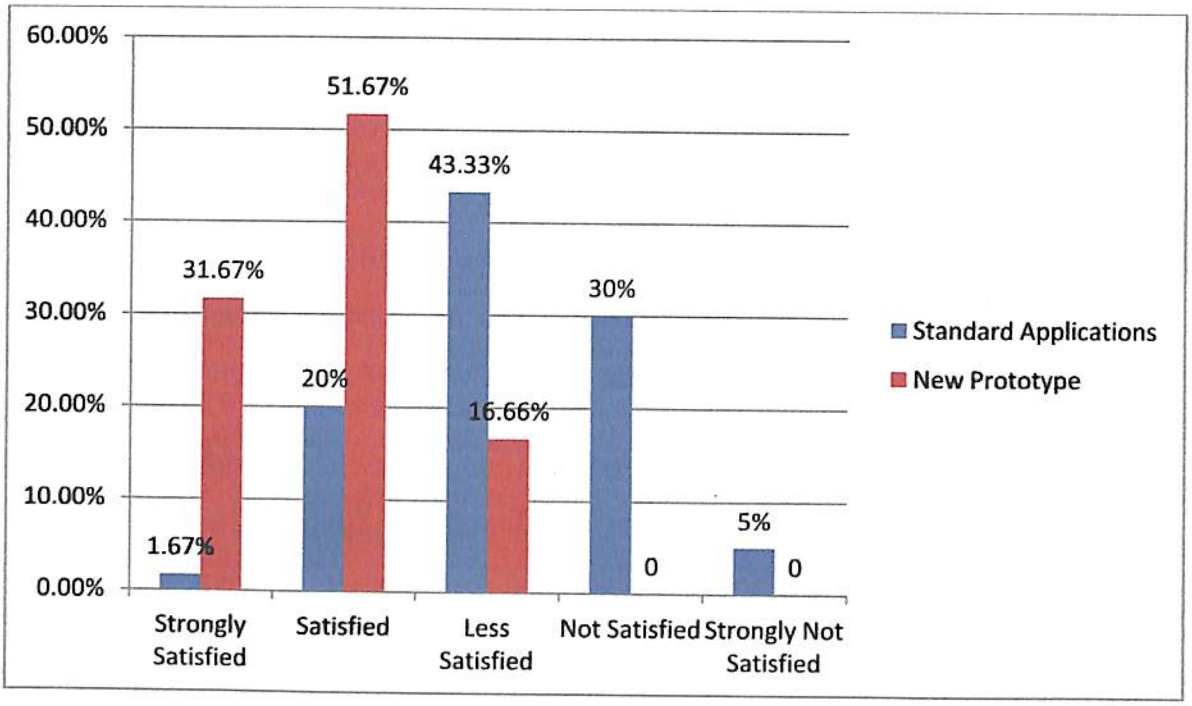
Table 7.1: Summary of user responses for Standard Applications (n=20)

Usability Evaluation Criteria	Strongly Satisfied (5 points)	Satisfied (4 points)	Less Satisfied (3 points)	Not Satisfied (2 points)	Strongly Not Satisfied (1 point)
Learnability	-	20%	30%	40%	10%
Low error rate	-	10%	10%	60%	20%
Memorability	10%	70%	10%	10%	-
Effectiveness	-	10%	70%	20%	-
Efficiency	-	-	70%	30%	-
Satisfaction	-	10%	70%	20%	-
Total	1.67%	20	43.33%	30%	5%

Table 7.2: Summary of user responses for New Prototype (n=20)

Usability Evaluation Criteria	Strongly Satisfied (5 points)	Satisfied (4 points)	Less Satisfied (3 points)	Not Satisfied (2 points)	Strongly Not Satisfied (1 point)
Learnability	40%	50%	10%	-	-

Low error rate	30%	50%	20%	-	-
Memorability	40%	20%	40%	-	-
Effectiveness	20%	70%	10%	-	-
Efficiency	30%	60%	10%	-	-
Satisfaction	30%	60%	10%	-	-
Total	31.67%	51.67%	16.66%	0%	0%



Graph 7.1: User Rating for Standard Applications and New Prototype

The above data shows that the users are very much satisfied with new prototype comparatively with Standard applications. However, the following statistical analysis will show in depth details for each usability criteria of both the interfaces.

7.3.1 Learnability

To assess the Learnability of both Interfaces, we have selected the user ratings for the statements 2, 3 and 4 from usability questionnaire. The following tables present statistical analysis for Learnability feature of the both interfaces.

Hypothesized difference (D): 0

Significance level (%): 5

Sample variances for the t-test: Assume equality

Table 7.3: Summary of Statistics for Learnability feature of New Prototype and Standard

Variable	Observations	Obs. with missing data	Obs. without missing data	Min.	Max.	Mean	Std. Deviation
User Rating for Learnability | New Prototype	20	0	20	3.0	5.0	4.100	0.718
User Rating for Learnability | Standard	20	0	20	1.0	4.0	2.100	0.852

t-test for two independent samples / Upper-tailed test:

95% confidence interval on the difference between the means:

Table 7.4: Two sample T test for New Prototype and Standard Applications Learnanility feature

Difference	2.000
t (Observed value)	8.025
t (Critical value)	1.686
DF	38
p-value (one-tailed)	<0.0001
Alpha	0.05

In table-7.3, the mean value of new prototype (4.1) is higher than the mean value Standard application (2.1) and in table-7.4 computed p-value(<0.0001) is lower than the significance level alpha=0.05. The above statistical analysis shows that Learnability of the prototype is better than the Standard applications.

7.3.2 Memorability

To assess the Memorability of both Interfaces, we have selected the user ratings for the statements 7 and 10 from usability questionnaire for

Statistical analysis. The following tables present the Statistical analysis of Memorability feature of the prototype and Standard applications.

Hypothesized difference (D): 0

Significance level (%): 5

Sample variances for the t-test: Assume equality

Table 7.5: Summary of Statistics for Memorability feature of New Prototype and Standard

Variable	Observations	Obs. with missing data	Obs. without missing data	Min	Max	Mean	Std. Deviation
User Rating for Memorability | New Prototype	20	0	20	3.0	5.0	4.0	0.918
User Rating for Memorability | Standard	20	0	20	2.0	5.0	3.8	0.768

t-test for two independent samples / Upper-tailed test:

95% confidence interval on the difference between the means:

Table 7.6: Two sample T test for Memorability feature of New Prototype and Standard Applications

Difference	0.200
t (Observed value)	0.748
t (Critical value)	1.686
DF	38
p-value (one-tailed)	0.230
Alpha	0.05

In table-7.5, the mean value of prototype (4.0) is higher than the mean value of Standard application (3.8) and in table-7.6 computed p-value(0.230) is higher than the significance level alpha=0.05. As per the above statistical analysis there is no significant difference between prototype and Standard applcation in respect to Memorability feature.

7.3.3 Effectiveness

To assess the Effectiveness of both Interfaces, we have selected the user ratings for the statements 1, 12 and 13 from usability questionnaire for Statistical analysis. The following tables present the Statistical analysis for Effectiveness feature of the prototype and Standard applications.

Hypothesized difference (D): 0

Significance level (%): 5

Sample variances for the t-test: Assume equality

Table 7.7: Summary of Statistics for Effectiveness feature of New Prototype and Standard

Variable	Observations	Obs. with missing data	Obs. without missing data	Min	Max.	Mean	Std. Deviation
User Rating for Effectiveness | New Prototype	20	0	20	3.0	5.0	4.100	0.553
User Rating for Effectiveness | Standard	20	0	20	2.0	4.0	2.900	0.553

t-test for two independent samples / Upper-tailed test:

95% confidence interval on the difference between the means:

Table 7.8: Two sample T test for Effectiveness feature of New Prototype and Standard Applications

Difference	1.200
t (Observed value)	6.868
t (Critical value)	1.686

DF	38
p-value (one-tailed)	<0.0001
Alpha	0.05

In table-7.7, the mean value of prototype (4.1) is higher than the mean value of Standard applications (2.9) and in table-7.8 computed p-value(<0.0001) is lower than the significance level alpha=0.05. As per the above statistical analysis the Effectiveness feature of prototype is notably significant over the Standard applications.

7.3.4 Satisfaction

To assess the Satisfaction of both Interfaces, we have selected the user ratings for the statements 11 and 14 from usability questionnaire for Statistical analysis. The following tables present the Statistical analysis for Satisfaction feature of the prototype and Standard applications.

Hypothesized difference (D): 0

Significance level (%): 5

Sample variances for the t-test: Assume equality

Table7.9: Summary of Statistics for Satisfaction feature of New Prototype and Standard

Variable	Observations	Obs. with missing data	Obs. without missing data	Min	Max	Mean	Std. Deviation
User Rating for Satisfaction | New Prototype	20	0	20	3.0	5.0	4.20	0.616
User Rating for Satisfaction | Standard	20	0	20	2.0	4.0	2.90	0.553

t-test for two independent samples / Upper-tailed test:

95% confidence interval on the difference between the means:

Table 7.10: Two sample T test for Satisfaction feature of New Prototype and Standard Applications

Difference	1.300
t (Observed value)	7.029
t (Critical value)	1.686
DF	38
p-value (one-tailed)	<0.0001
Alpha	0.05

In table-7.9, the mean value of prototype (4.2) is higher than the mean value of Standard applications (2.9) and in table-7.10 computed p-value(<0.0001) is lower than the significance level alpha=0.05. As per the above statistical analysis the Satisfaction feature of prototype is elevated over the Standard applications.

7.4 Conclusion

In this chapter, usability evaluation for Standard as well as new prototype has been presented. Further, the improved usability of the New Prototype over Standard interface was also presented with statistical analysis.

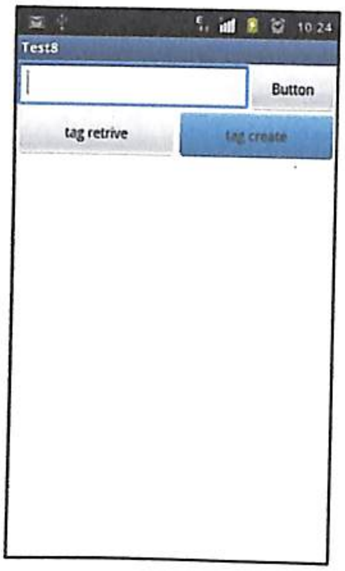
Chapter 8
Output Screenshots

- Tag creation screen shots
- Tag Retrieval screen shots:
- Record updating screen shots

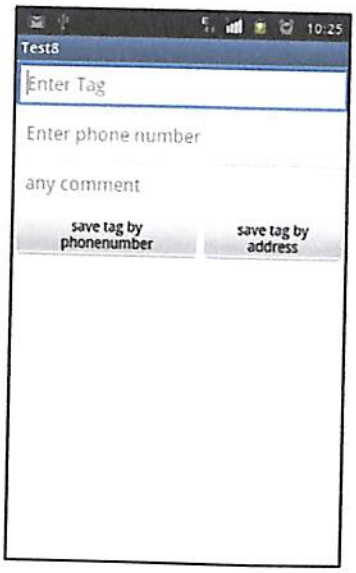
8. OUTPUT SCREEN SHOTS

8.1. Tag creation screen shots

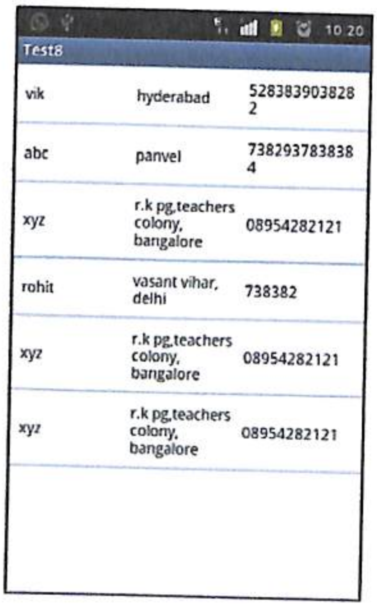
For creating tag user click the tag create button. After clicking that button new screen opens where user enters tag name along with information like phone number, any comment etc and save the tag either phone number or address and it will save in database.



After pressing tag create button then prompts for metadata about tag.



Now user has choice to save tag either by phone number or by address which results storing in database and can be viewed as

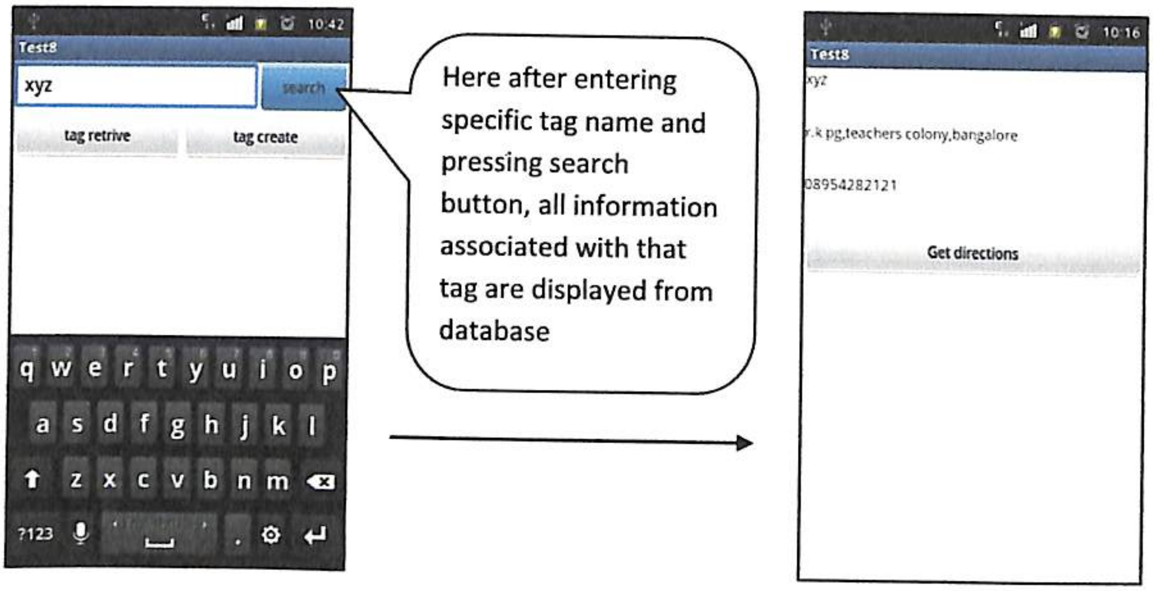


vik	hyderabad	5283839038282
abc	panvel	7382937838384
xyz	r.k pg,teachers colony, bangalore	08954282121
rohit	vasant vihar, delhi	738382
xyz	r.k pg,teachers colony, bangalore	08954282121
xyz	r.k pg,teachers colony, bangalore	08954282121

Fig 8.1 Screen shots of Tag creation and saving into database

8.2. Tag Retrieval screen shots:

If the user wants to see information about tag, it will get simply by entering the tag name and press search button and the information about the tag fetch from the database and display to the user.



xyz

Here after entering specific tag name and pressing search button, all information associated with that tag are displayed from database

xyz

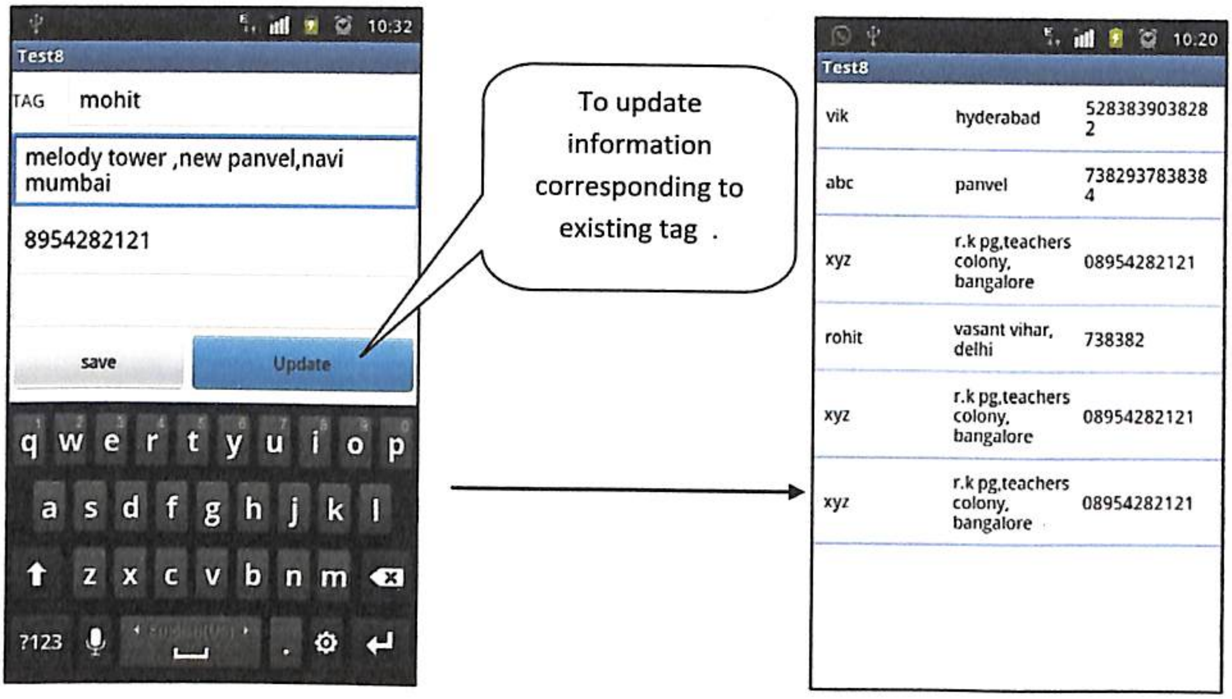
r.k pg,teachers colony,bangalore

08954282121

Fig 8.2 Screen shots of tag retrieval from database

8.3. Record updating screen shots

If user has changed the phone number and other information and wants to update the information, simply filling the update information and press the update button and it will update in the database.



TAG mohit

melody tower ,new panvel,navi mumbai

8954282121

To update information corresponding to existing tag .

vik	hyderabad	5283839038282
abc	panvel	7382937838384
xyz	r.k pg,teachers colony, bangalore	08954282121
rohit	vasant vihar, delhi	738382
xyz	r.k pg,teachers colony, bangalore	08954282121
xyz	r.k pg,teachers colony, bangalore	08954282121

Fig 8.3 Screen shots of record updating into database

Chapter 9
Product Usability

- Find me now
- Call a cab
- Direction to any events
- Direction to any events
- No need to type long address while driving
- Phone contact Synchronization
- Commercialised Tags
- Delivery Services

9. PRODUCT USABILITY

There are several use cases and application that can be used by this application. The important use cases that are essential are described below:-

9.1 Find me now

Findees in locations that they are unfamiliar with, can provide directions to potential Finders through a 1-click-process on their mobile phone Application that creates a temporary tag and sends it to the Finders in one of the following methods:

(i) With 1-click the Fast-MapTag Application can generate a temporary map-tag that is a simple English or vernacular phrase (few words) and which is associated with the precise geo-location of the Findee's current location. This phrase can be verbally or otherwise communicated to the Finder (e.g. via email or SMS).

(ii) By providing the Finder's mobile number or Finder's email address, the Findeee can, with a 1-click process the Fast-MapTag Application can automatically send a temporary map-tag to the Finder

(iii) By providing the Finder's Fast-MapTag username, the Findeee can, with a 1-click process create and send a maptag automatically to the mobile Application of the Finder.

Using any of these or some combination of these methods, the Findee can provide precise location and hence directions, to the Finder, even without having any knowledge of the location or nearby landmarks.

9.2 Call a cab

Findees in locations can call a cab, and provide directions to cab driver through a 1-click-process on their vehicle GPS system, that creates a temporary tag and sends it to the Finders in one of the following methods:

1:- The user calls a cab operator and asks for a cab service.

2:- The cab service operator asks for the present location tag or tag which is already created. The operator sends this tags detail to the cab driver nearest to the tags location.

3:- The cab driver gets the direction to the customers location through his GPS embedded in the cab.

9.3 Direction to any events

In India daily thousands of events keep going on like auction, sale, fuctions etc. The event organiser can create a temporary tag for their event which not only shows the direction to reach the event but also shows the necessary details about the events like timing, programmes etc. The event tag has to be present with their advertising pamphlet so that people can know about the event tag.

9.4 No need to type long address while driving

While driving it is very hard to type the full address into the map and get the direction. With the help of fast map tag the user can easily type a 3-4 alphabet tag into the mobile or vehicles GPS and reach the destined direction.

9.5 Phone contact Synchronization

Once the application is installed the app will synchronize with phone contact and shows the contact list in the application. The contact list in app will show the names of other contacts that are using the same application. So even by help of phone contact also user can get the directions to person listed in phone contacts.

9.6 Commercialised Tags

Through this application one can create a tag for the commercial purposes also. The commercial shop can create their tag and stores necessary information like phone number and shop timing. User can directly reach to the Shops tag without asking anyone. The user can be fully dependent on mobile application as sometimes human tends to misguides the user.

9.7 Delivery Services

For delivery services such as pizza delivery, couriers etc., the method provides for an easier way to create automated training and help applications, cutting down the time of the trainers and supervisors as well as improving overall customer satisfaction.

Chapter 10

Result and Discussion

10. RESULT AND DISCUSSION

By performing various surveys and market research it has been found that for country like India where all locations are not mapped in the map. This type of application will be much more helpful for the user. It can be helpful in commercial way or it can be helpful in general way. The advantages will for both sides i.e. for findee and finder.

Findee's advantages:-

- Can easily communicate their location with simple natural language words.
- Can use maptag in their email signature for easy communication of physical addresses
- Reduction in time for Findee in helping Finders with last minute direction requests and/or delayed meetings
- Reduction of effort in providing directions repeatedly to all Finders seeking the Findee by referring them to the online Fast-MapTag.

Finder's advantages:-

- Convenience: Simple 1-click process from entering address to locating directions
- Most time-efficient - removes the hassle of long process in opening mapping applications and/or enabling GPS to find out directions for a destination
- Augments automated mapping services with currently practiced landmark driven navigation process followed conventionally in countries like India.
- Removes the need of internet to find navigations to the destinations as basic directions can also be obtained via an SMS service.

Chapter 11
Conclusion

11. CONCLUSION

Fast map tag is much easier and convenient application to get the destination's direction especially in places like India, where maximum locations are not mapped in existing maps. This application helps in reaching the destination address möst accurately other than existing navigation applications. Moreover it can be used in both commercial and general ways and giving advantages to both sides. The commercial advantages are like advertising, event scheduling, and courier services etc. The major advantage of this application is one can get to the findees address by using findess mobile number and can also get the necessary landmark details of the destination.

Appendix A

Tasks for Usability Evaluation.

1. Create your account in Application
2. Login into your account
3. Select and promote a user
4. Browse for the profile picture
5. Search the usernames based on location
6. Pick the Image files from above search activity
7. Change the Interface Language
8. Refer any username to your friend
9. Logout
10. Search the usernames based on age limits

Appendix B

S.No	Statement	Strongly Satisfied (5)	Satisfied (4)	Less Satisfied (3)	Not Satisfied (2)	Strongly Not Satisfied (1)
1	It assists me to get my results in effective manner					
2	It was easy to learn to use					
3	It boots up my knowledge related to my interested field					
4	Exploring the new features by trial and error is					
5	It is time saving when I am using it					
6	It gives me results according to my desired search					
7	It is simple and easy to use					
8	It Provides efficient search techniques					
9	This system shows positive feedback at every step					
10	It is consistent and its is not difficult to remember all steps					
11	I should recommend this to my colleague.					
12	It is reliable and gives me information according to my needs.					
13	The functions/facilities of the system is effective, facilitates me to complete my tasks					
14	I am satisfied with its use.					

References

[1] Meier, R.Professional Android 2 Application Development. Indianapolis, IN. Wiley Publishing, Inc

[2] .Bishop, T. (2012, March 6). “Google Play replaces Android Market, cosolidates Google’s media marketplaces”. Retrieved from http://www.geekwire.com/2012/google-playreplaces-android-market-consolidates-googles-media-marketplaces

[3] what is Android? (2012). Retrieved March 10, 2014 from http://developer.android.com/guide/basics/what-is-android.html

[4] Siddartha Sreenivasa Reddy, TRIP TRACKER APPLICATION ON ANDROID, Master thesis, May 2011.

[5] Israfil Coskun and Osman Celik. Challenges of Mobile ApplicationDevelopment,2010. http://developer.smartface.biz/challenges-of-mobile-application-developementn74.html.

[6] “Wonobo”,www.wonobo.com/india, Retrieved on Fbruary,2014.

[7] G. Inc. (2010, Nov.) Android Developer Guide. [Online]. http://developer.android.com/guide/basics/what-is-android.html

[8] A. D. Team. (2007, Nov.) YouTube presentation. [Online]. http://www.youtube.com/watch?v=Mm6Ju0xhUW8

[9] S. Hashimi, S. Komatineni, and D. MacLean, "Pro Android 2," in Pro Android 2. New York, US: Apress, 2014, ch. 1, pp. 10-11.

[10] Eclipse Foundation, Java development tools, http://www.eclipse.org/jdt/,2009. online: June 2009.

[11] Steele, J. (2011). The Android Developer's Cookbook. Boston, MA. Pearson Education, Inc. The Apache Software Foundation. (n.d.). “Chapter 2. Connection management”. Retrieved from http://hc.apache.org/httpcomponents-client-ga/tutorial/html/connmgmt.html

[12] Android Developers. Google. Tools. http://developer.android.com/guide/developing/tools/index.html

[13] Aman Singhal, Place Me: Location Based Mobile App for Android Platform, November 22, 2010

[14] Android Developers. SQLiteOpenHelper, 2010.http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html

[15] Android Developers. Android Location Package, 2010.http://developer.android.com/reference/android/location/package-summary.html

[16] Sergey Ivanenko,Agatha Wong, Working with Geocoding APIs[pdf],ESRI development summit, 2008

[17] Njunjic, Ivan, "Development Techniques for Android Platform Mobile Device Application" (2012). Master's Theses and Doctoral Dissertations. Paper 394.

[18] Wikipedia.org. Android Operating System, 2011, http://en.wikipedia.org/wiki/Android_(operating_system), accessed Mar. 2014.

[19] Rick Rogers and John Lombardo. Android Application Development, 2009.http://androidapps.org.ua/i_sect13_d1e11121.html

[20] Android GUI Architecture, ListView Backgrounds: An Optimization. (n.d.). Retrieved February 10, 2012 fromhttp://developer.android.com/resources/articles/listview-backgrounds.html

[21] Wikipedia. Dalvik virtual machine. http://en.wikipedia.org/wiki/Dalvik_virtual_machine, 2009. online: June 2009.

[22] Vogel, L. (2008). “Apache HttpClient - Tutorial”. Retrieved March 21,2014fromhttp://www.vogella.de/articles/ApacheHttpClient/article.html

[23] Vogel, L. (2010a). "Android HTTP Access - Tutorial". Retrieved March 14, 2014 from

[24] http://www.vogella.de/articles/AndroidNetworking/article.html

[25] Vogel, L. (2010b). "Android Threads, Handlers and AsyncTask - Tutorial". Retrieved January 10, 2014 from http://www.vogella.de/articles/AndroidPerformance/article.htm

[26] SQLite Official webpage. http://www.sqlite.org/about.html

[27] Jeng, J., (2006). What is Usability in the Context of the Digital Library and How can It Be Measured?. Technical Report. The School of Communication and Library Studies, New Jersey.

[28] Nielsen, J., (2003). Usability 101: Introduction to Usability. Available at: http://www.useit.com/alerbox/20030825.html

[29] Oronzo, P., Enrica, M., Sebastiano, B., (1999). Multimedia in Distance Education: Effects of Usability on Learning, Interacting with Computers, PP 37-44.

[30] Nielsen, J., and Mack, R, (1994). Heuristic Evaluation. In J. Nielsen and Mack, Usability Inspection Methods. John Wiley.

[31] Wharton, C., Rieman, J., Lewis, C. and Polson, P, (1994). The cognitive walkthrough method: A practitioner's guide. In J. Nielsen and Mack, Usability Inspection Methods. John Wiley.

[32] Sutcliffe, A. and Carroll, J. M., (1999). Designing claims for reuse in interactive system design. International Journal of Human-Computer Studies, 50.

[33] Chin, J.P. Diehl, V.A. and Norman, K.L. Development of an instrument measuring user satisfaction of the human-computer interface. CHI'88 Proceedings. (1988) 213 - 218.

[34] Nielsen, J. (1993). Usability engineering. San Diego, CA: Academic Press.

[35] Seonho Kim (2008), Visualizing Users, User Communities, and Usage Trends in Complex Information Systems Using Implicit Rating Data, Ph.D thesis, Virginia Polytechnic Institute and State University.

[36] Brooke, J. (1996). SUS: A Quick and Dirty Usability Scale. In: P.W. Jordan, B. Thomas, B.A. Weerdmeester & I.L. McClelland (Eds.), Usability Evaluation in Industry. London: Taylor & Francis. Available at http://www.cee.hw.ac.uk/~ph/sus.html.

[37] Lewis, J. (1995). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. International Journal of Human-Computer Interaction, 7 (1,) 1995, 57-78. Available at http://www.acm.org/~perlman/question.cgi?form=CSUQ