TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

KANIZ FATEMA

BIGBANG: AN APPLICATION TOWARDS BUILDING BUSINESS ECOSYSTEM THROUGH THE INTEGRATION OF MYGEOTRUST PRIVACY PLATFORM

Master of Science thesis

Examiner: Prof. Kari Systä
Examiner and topic approved by the Faculty Council of the Faculty of Computing and Electrical Engineering on 4th May 2016

ABSTRACT

Kaniz Fatema: BigBang: An application towards building business ecosystem through the integration of MyGeoTrust privacy platform
Tampere University of Technology
Master of Science Thesis, 46 pages
June 2016
Master’s Degree Programme in Information Technology
Major: Software Engineering
Examiner: Prof. Kari Systä

Keywords: Micro business, Android, MyGeoTrust, constructive research.

Micro or small business often defined as the business that has less than five employees including the owner himself. Additionally, it mostly operate in a certain geographical area and the consumer base that surrounds its’ vicinity. Even with their limited focus and capacity such businesses combined carries a large share of overall economy of a county. According to statistics, in the USA for instance, such businesses host 41.3 million jobs either directly or indirectly.

However, to publish and commercialize these business services are not trivial task due to mainly economical constrains. These businesses often use cheap means of advertising to reach their consumers. Such initiative often includes, launching business websites of their own using mostly free templates and releasing promotional images or videos through the sites.

This thesis is an initiation to support such micro businesses and their consumers. The target is to build a platform, which will provide both the stakeholders a common ground to maximize their interests and benefits. For instance, the businesses can launch their business offers, view and analyze the rivals, set business strategy. On the other hand, consumers could have a virtual megamall of vibrant services. Furthermore, this platform is integrated with the MyGeoTrust privacy service to study the usability and applicability of the service under a number of pragmatic scenarios.

PREFACE

This thesis work was done in the Department of Pervasive Computing as part of partial fulfillment of MSc degree.

I would like to thank Prof. Kari Systä for considering me worth for this work and his continuous support and guidance during the thesis work. I would also like to thank Dr. Robert Guinness for his initial guiding during this work.

I am thankful to almighty and grateful to my husband for supporting me to complete my study.

I would like to thank my daughters, my parents, my brothers and my friends to be always by me.

Tampere, 16.5.2016

Kaniz Fatema

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Micro/Small businesses along with their consumer base represent the most significant share of the economic demography of many dominant world economies, including USA and the UK. According to the research produced by the Small Business Administration (SBA), micro businesses are usually the job creators in USA [1][2]. Another report on the track states that micro businesses create 64% of all the net new jobs in USA during the period between 1993 and 2009 [1]. This is largely expected as this business segment constitute about 92% of all the businesses in the USA, employed 50% of employees in the private sector and pay 44% of the total private payroll in the country [2]. With over 90% of consumer support of all classes, micro businesses possesses survival rate comparable to that of large businesses.

However, due to limited capital and manpower such business often lags behind in promoting their business services and providing means to their consumers a comprehensive access support. This thesis is thus targeted in shaping a framework that would offer such business and their consumers a single point reference of interaction for maximizing mutual benefits.

The application, which is intended to be implemented within the scope of this thesis, is named **BigBang**. The underlying motive is three fold: (a) Study and implement a viable solution to bridge small/micro businesses and its' customers, (b) integrate the privacy platform MyGeoTrust [3] within the solution implementation to better ensure user privacy, and (c) study the pragmatic applicability of MyGeoTrust.

Under the first motive, the intention is to provide a viable solution platform to two key components of a business: the business itself (specifically, small or micro businesses within a defined geographical area) and the consumers. For the businesses of such scale this application will offer the opportunity to expose their business services to a larger community of consumers. Through this solution platform a business can freely register within a category of business, unveil their business service, offers and promotions, can get the options to interact with customers (e.g., contacting, booking, buying, location services and others). A business could also get the opportunity to assess their business rivals, and fix their business strategy accordingly, which in turn ensures a healthy means to ensure best value added services to the consumers.

On the consumer side, this solution will provide a single point reference to get access to a fascinating and diverse domain/area of business services. At any given time, they could effectively access to all the services that are offered by the nearby business service providers, get their offers, compare, favorite, log, communicate, subscribe for new promotions / offers, navigate themselves to the business and others. In short, it will provide a client / consumer the virtual mega mall service where they could get access to their desire services. Figure 1 summarizes the overall concept of this solution platform.

Within the scope of the second motive, the privacy platform MyGeoTrust [3][4] will be integrated within the solution platform. According to the mandate of MyGeoTrust project, this platform should provide privacy and transparency in crowdsourcing user location data as opposed to native platforms, e.g., Google. As, the proposed solution platform should use user location information to offer e.g., navigation services, therefore this integration would gain higher consumer confidence.

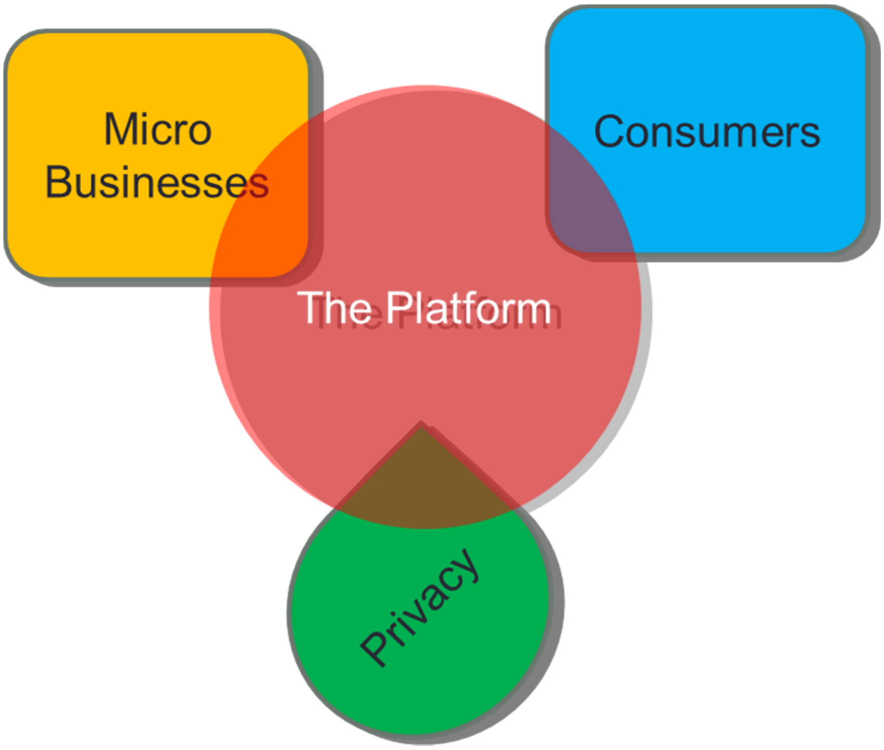


Figure 1 Overall concept of BigBang Platform

Finally, the usability and applicability of MyGeoTrust platform will be studied under a number of pragmatic scenarios. For instance, the integration process of the platform with third party applications, API standard, pragmatic usage and implementation standard are studied.

This thesis is organized as follows: in Chapter 2 the background of this thesis is presented. This includes a discussion on the Micro businesses, their overall impact on economy and a detail discussion on the MyGeoTrust project. Chapter 3 presents the detail specification of BigBang application that is implemented within the scope of this thesis. Chapter 4 is devoted in discussing the methodology and study setup. Implementation detail of BigBang application containing discussion on architectural design, underlying rationale of certain choices, and the integration process with MyGeoTrust service are presented in Chapter 5. Chapter 6 outlines the analytical detail of MyGeoTrust that are observed during implementation and evaluation. Finally, future directions and concluding remarks are presented in Chapter 7 and 8, respectively.

2. BACKGROUND

This chapter plots the background of this work. First, an overarching description on microbusiness is presented. Then, the MyGeoTrust service is elaborately discussed that is integrated as part of the BigBang platform to provide privacy and security to users' data.

2.1 Micro Business

Microbusiness or Small business is defined as an enterprise having fewer than five employees, including the owner [5]. Additionally, such businesses often reside within a given geographical location and serving consumers in their vicinity. However, business of such scale plays a pivotal role while considering their combined impact on overall economy of a country. For instance, micro firms represent 99.7% of all the employer firms in the USA [1]. They employed more than 50% of employees in the private sector and pay 44% of the total private payroll [2]. They offer 26 million direct jobs, while indirectly influencing another 1.9 million jobs and inducing 13.4 million jobs. Therefore, as a whole this business segment hosts a total of 41.3 million jobs in the USA [5]. This is represented in Figure 2.

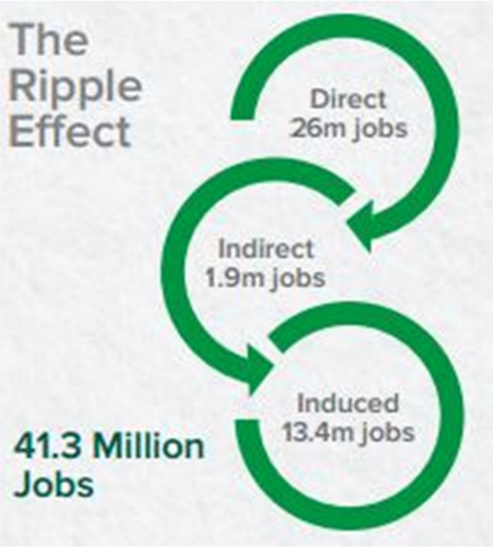


Figure 2 Micro business economy in USA [5]

During the last 15 years, micro businesses created 64% of all the job opportunities in the USA. Additionally, the survivability rate of this kind of business is high. About 70%

of all businesses tend to survive for a minimum of two years, 50% survive for five years, one-third stays in business for at least a decade, and a fourth of all start-ups last for 15 years or more [2]. These survival rates are comparable to major industries and states, which have the similar survival rate.

Finally, for the past decade micro businesses represent a growing trend. Despite of economical depressions, new businesses are coming up every year with great ideas and a sizable percentage of them survive for a minimum of two years [2]. Survival changes increases with the willing to work hard to realize it.

Due to such benefits, consumers and common people often favor such businesses. More than 90% of all consumer classes in the USA positively support micro businesses, and believe that they have a deep impact in the economy through job creation and taxes [1][2].

2.2 Micro Business Promotion

When it comes to promote the business services, it is observed that they predominantly rely on online-based solutions. As can be seen from Table 1, 90% of the means of business promotion relates to online services, including launching company website (44.2%), email communication (34.4%) and promotional picture or video releasing on website (16.7%). Reason behind such choice might be the cost effectiveness of online services.

Table 1 How small businesses promote their products online [15]

Company website	44.2%
Email marketing	34.4%
Videos/still pictures on website	16.7%
Banner ads	10.0%

Furthermore, cheaper means are often chosen in creating and launching business websites. For instance, majority of the efforts of launching business website consists of doing it oneself (82.2%) or using cheaply available alternatives, e.g., using off-the-shelf PC software (55.9%) or editors (35.7%) or basic templates (7.6%) freely provided by the ISP, among others. Table 2 offers the detail listing of the alternatives popularly used in USA.

Table 2 Small Business Website Development Method [15]

"Do It Yourself" (net)	82.2%
Standard off-the-shelf software for use on PC	55.9%
HTML editor	35.7
% Basic templates provided by others	12.5%
Basic template provided by ISP	7.6%
Browser-based applications provided by ISP or others	2.1%
"Do It For Me" (net)	24.6%
Website design firm/consultant	17.1%
Friends/others	5.0%
ISP or ISP partners	3.0%
Service provider that provided name registration and other services	1.0%

These statistics and observations that boldly indicate the dominance of cheap but effective means to promote business services and reaching large volume of consumers lies the basis of this thesis. This thesis is thus committed to offering a platform of benevolent benefits for both the businesses and the consumers.

2.3 MyGeoTrust Project

The overarching focus of MyGeoTrust project [3][4] is to ensure the security and privacy of individual's data while using mobile devices that offers crowdsourcing of data with very little control over those data. For instance, companies such as Google and Apple offer single point license agreements that users must have to accept to use the products powered by them. For instance, mobile devices powered by Android or iOS, several applications using location services, such as Google maps. However, these licenses offer these companies the legal right to share individual's location information with third parties of their choice (e.g.,[6][7]). At bare minimum this collected information includes the location updates derived from Global Navigation Satellite Systems (GNSS), and the identity and strength of available radio networks, such as cellular networks and wireless local area networks (WLAN). Such data with individual's personal information (often termed as crowdsourced mobile data) are vital asset for many different market segments, such as transportation, advertising, urban planning, media and entertainment, real estate, health, and education [3][8].

The above developments have both positive and negative implications. On the positive side, a mobile device gains the ability to understand, and predict the user's preferences and offers services accordingly [9]. Additionally such information can also benefit entire user community as well or even the general public in many non-obvious ways [10]. On the negative side, third parties (e.g., companies, individuals or surveillance organizations) who gained access of these data can exploit it in many undesirable ways. Users often do not have any trace of who is viewing, monitoring and tracking them.

Privacy concerns are increasingly becoming an issue for mobile device consumers [3]. A 2012 study examined the attitudes of US consumers concerning mobile phones and privacy by conducting a nationwide survey [11]. The survey asked several questions related to mobile phones and location privacy. A plurality (46%) answered that user location information should not be tracked and kept by the service providers. Additionally, 92% responded either that they would definitely or probably not allow to user their location information to tailored commercial adds to them.

In March 2015, a research project under the name MyGeoTrust has been launched to address some of the privacy concerns related to geo-tagged data collected by mobile platforms and the applications that run therein[3]. This project has received funding from the Strategic Research Opening program of the Finnish Funding Agency for Innovation, which seeks bold visionaries and a multidisciplinary approach to tackling challenging industry issues. The two organizations involved in the project are the Finnish Geospatial Research Institute and the Faculty of Law at the University of Helsinki [3].

The driving force of MyGeoTrust project is the belief that it is possible to develop an approach for protecting privacy in using geo-tagged context-aware services that supports the core ethical principle of autonomy, authority, transparency and control of the individual over his personal data [12].

2.3.1 Privacy Concern in MyGeoTrust

The aim of MyGeoTrust project is to define a platform for geospatial services where ensuring the user's privacy is at the core of the platform. In brief, following privacy concerns should be addressed by the platform:

(a) Collection of personal data: currently the platform (as well as many third party applications) running on user device has the right to collect both user location information and users' identity traces [13].

(b) Association of location data with users' identity: it is not traceable how the location information is associated with users' personal information, e.g., identity, timestamp [13].

(c) Legal authority over personal data: The platforms or service providers have full authority over users' personal data through a very generic licensing agreement [14].
(d) User right over personal data: Individuals do not enjoy any right to monitor, access or control the usage of their own data [14].

In order to encounter these privacy concerns, three design principles are adopted to design MyGeoTrust. In related literature these principles are defined as the core properties of any privacy protected system. In brief these principles are,

Privacy by Design:

The mandate of Privacy by Design is to ensure privacy and gain personal control over one's information through privacy protective system design [17] [16]. In order to achieve this, seven fundamental principles for Privacy by Design are proposed that need to be embedding into the design specifications of the system [17]. MyGeoTrust is committed to implement each of these principles, which are summarized as follows,

(a) The system must be Proactive not Reactive; Preventative not Remedial in protecting user privacy.
(b) The system should offer Privacy as its default settings.
(c) Overall system design must embed privacy.
(d) In all aspect of system functionality it must respect User Privacy.
(e) Data usage within the system should be transparent to the end user.
(f) The user should have access to the mechanism of controlling his data.
(g) The system must offer full functionality and the end-to-end security without any sacrifice.

User Decides:

As articulated in the Mobile Privacy Disclosures by FTC [18], the term User Decides or Consumer Choice, refers to the provision where users has the right to choose in case of a transaction that is not consistent with the business-consumer agreement. However, being a proactive and preventive system by nature, MyGeoTrust defines this principle at a fine grained level. According to the mandate, MyGeoTrust platform will leave decisions about how the platform operates, to the greatest extent possible, in the hands of the user. This especially applies to privacy-related decisions [3].Therefore, users should be given the control over all the configurable options of the system that relate to privacy centric services.

In addition to such control, users should have the sole authority over their data (Data ownership). Under this authority of data ownership, the user will retain the right to delete his or her data; have the rights to grant transfer of his or her data to third parties or to restrict such transfers; have the right to view and otherwise access his or her data in a comprehensible form; and so on.

Trust through Transparency:

As reported in the Mobile Privacy Disclosures by FTC [18], Trust through Transparency should force Companies to disclose details about their collection and use of consumers' information. Therefore, MyGeoTrust must offer its intended consumers clear disclosures which explain in plain terms (without any legalese or technical jargon) the followings,

(a) Preciously explain and present to the users what personal information will be collected, retained and shared.
(b) Explain what legal authority users have as part of the consumer's choices. It must also present how such authority can be exercised by the user.
(c) Being a proactive system, it must provide periodic reminders to its users when their location information is being shared.
(d) If a new service is going to be introduced over the location information previously collected, the user should be adequately notified beforehand with an opportunity to exercise their authority of choice.

2.3.2 Project Goal

The research project MyGeoTrust has been initiated to challenge the status quo regarding collecting, delivering and using crowdsourced mobile data, due to the privacy issues presented in earlier section. The project is intended to develop a mobile middleware stack and a server application to address users' privacy issues. The middleware implements a privacy and security solution that runs as an android service. Within this service the privacy model is implemented as a set of customizable privacy profiles representing different privacy settings [3]. Through these profiles and their fine grained customization facility users can control the access to various mobile OS resources, such as location services, Wi-Fi, Bluetooth as well as the type of sharing that occurs from mobile applications that utilize this middleware (e.g. anonymous sharing, sharing within a private group, no sharing at all, etc.) [1]. This middleware is termed as MyGeoTrustClient stack and is discussed in following section.

2.3.3 Current Implementation

The MyGeoTrustClient stack as presented in the previous section is intended to serve two primary purposes: (a) ensure users privacy through a set of customizable profiles that allows the users to control the usage and sharing of devices resources among third party applications, and the collection and usage of the data by those applications; and (b) offers a set of standard API's that allows third party applications to use device resources in an standard, traceable way.

Figure 3 presents the architecture of the stack, which follows layered architectural pattern. At the very top level resides the UI layer that is used by the user to configure the settings in a privacy profile, or create new profile with customized privacy settings or switching between profiles, and creating user accounts.

Within the data crowdsourcing module, location data and other crowdsourced data, such as Wi-Fi scans and sensor data, are collected and forwarded to the MyGeoTrust server. However, such data sending is controlled by the user set privacy settings, which includes for instance how frequently data will be sent, should it be anonymized, what data it should contain and with whom this data can be shared.

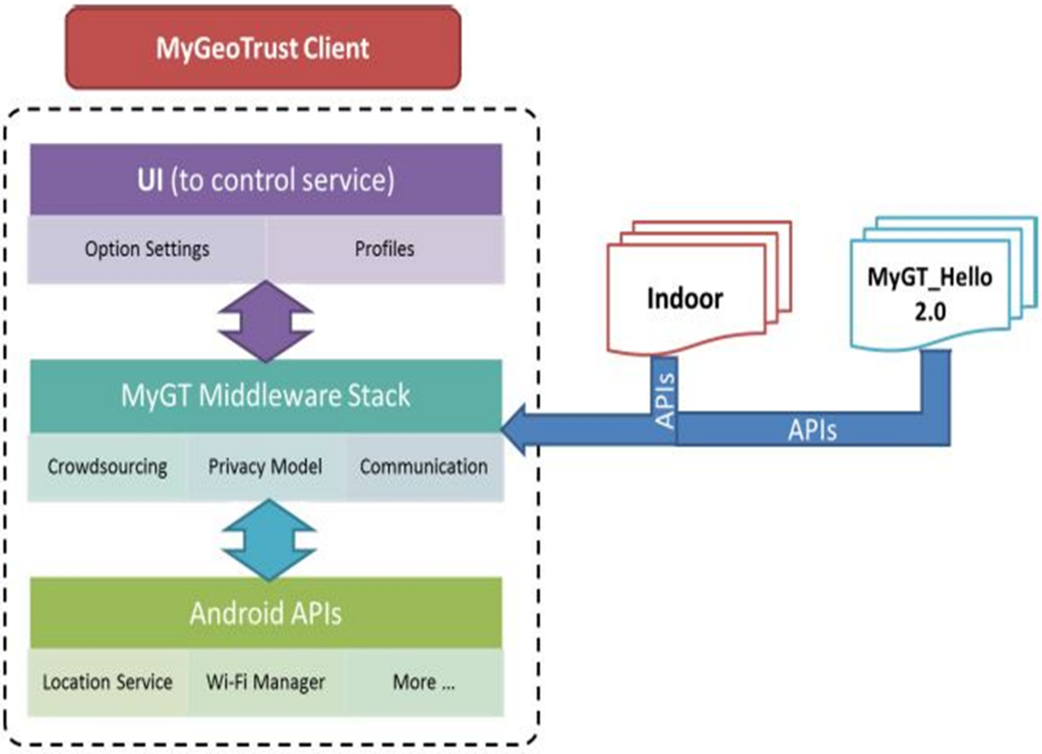


Figure 3 MyGeoTrustClient Stack Architecture[1]

The communication module is responsible for exposing APIs for third party applications to bind with the service and gain access to system resources through the service. Many of these APIs extend corresponding Android native APIs, such as the Android location APIs, but they control resource usage according to the privacy settings in the profile. For instance, a third party application bound to the service can gain access to GPS only if user allows it through the currently active profile settings. Therefore, the third party applications do not have any explicit access right of system resources (as opposed to current practices in Android OS). Instead, they granted or denied access right of a certain resources by the user at runtime through the privacy settings of the service.

Finally, the third layer consists of the Android native APIs used by the MyGeoTrust middle layer. This layer is platform-specific and thus subject to change with the platform and specific version in use.

3. GOALS OF THE RESEARCH

This chapter elaborates the features of BigBang application that need to be implemented as part of this thesis and the requirements in integrating the MyGeoTrust service.

The overarching goal of the application is to bridge business with their potential consumer base in a cheap, efficient and convenient way. Primary focus is to target the businesses that are located in a given geographic location and having a very focused business services, for instance, saloons in Espoo, or car repair shops in Hervanta. The consumers therefore would be the people living the vicinity of these businesses or who are travelling to a location where they are in need for such business services. This thesis termed these businesses as micro / start-up business. The overall concept of the application is portrayed in Figure 4.

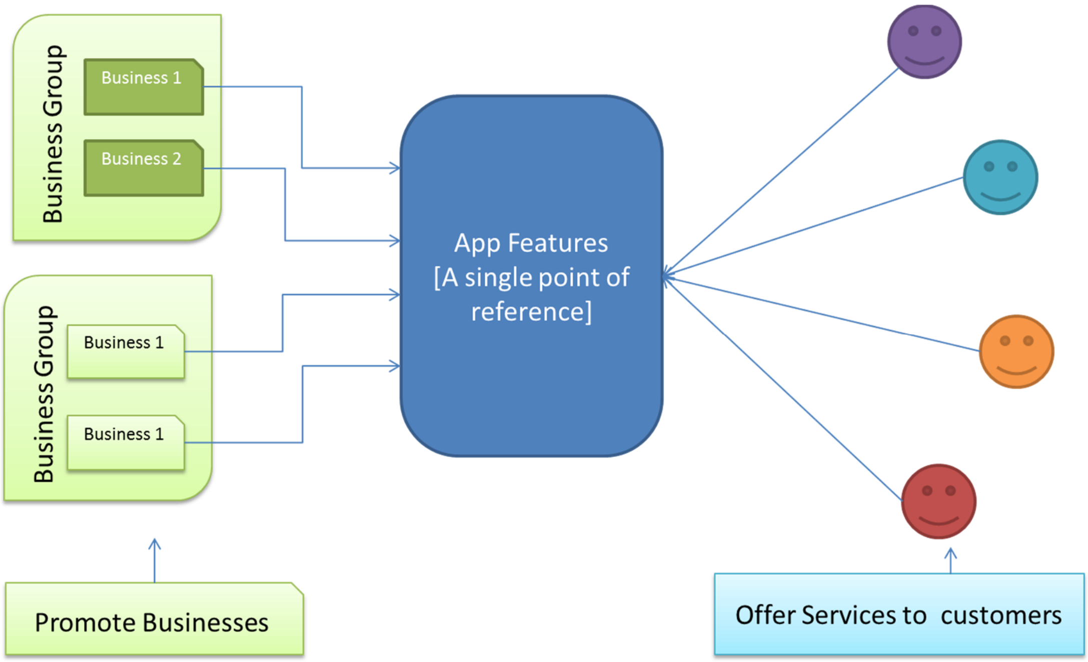


Figure 4 BigBang application services

As per this figure, small businesses can register and expose their business through the application and the consumers using the application can get access to those services. Therefore, for the businesses the platform offers opportunity to expose business services to a wide range of consumers / clients / customers, which is otherwise impossible for a business of such scale. Through this platform a business for instance, can freely register within a category of business, unveil their business service, offers and promotions, can get the options to interact with customers (e.g., booking, buying, contacting, registering, location services and others). Additionally, they also get the opportunity to assess their business rivals, and fix their business strategy accordingly, which in turn ensures a healthy means for best value added services to the consumers.

On the other hand, the consumers of the business services, this platform will provide a single point reference to get access to a fascinating and divertive domain/area of business services. At any given time, they could effectively access to all the services that are offered by the nearby business service providers, get their offers, compare among available options, favorite their preference, subscribe for new promotions / offers, navigate themselves to the business and many more. The bottom line is that it would provide a consumer the virtual mega mall service where they could get access to their desired services.

Rest of this section is dedicated in describing the application features in general, and specific need for the integration to MyGeoTrust.

The end product within the scope of this thesis should offer the following features for both the consumers and the businesses. Further enhancement of the platform will be carried out in future, a discussion on which is presented in Chapter 7.

(a) For Consumers:

- Customer can register to get personalized services, e.g., bookmarking and search/browsing history.
- Categorically search for service businesses services. For instance, cloth stores, or grocery shops nearby.
- Search shops within a given category.
- View shop information, product listing, address and contact detail of a business.
- Bookmark/Favorite a shop.
- View search history.
- Communicate with the shop: call, email, or browse website (if any).
- Get a custom map navigation service to get turn-by-turn navigation from customer's current location to the shop.

(b) For a business:

- A business could register for a business account.
- A business can unveil their business service, offers and promotions on its own business page.
- A business can get the options to interact with customers (e.g., contacting, booking, buying, location services and others).

(c) The navigation service:

The navigation service should offer at-least the followings:

- Plot shop location on the map while user selects the navigation option on a shop’s page.
- Plot driving route on the map from customer’s current location while requested for a route to the shop.
- Show the distance and the total time required to travel to the shop from users’ current location.
- Start turn-by-turn navigation while customers request for such.
- During real-time navigation it will show distance from the next turn to better help the user in navigation.

(d) MyGeoTrust integration:

Integration of MyGeoTrust service should verify the followings,

- Test how well the BigBang as a representative of third party application integrates with MyGeoTrust service.
- Test the usability of the API’s exposed by MyGeoTrust service. This should include the verification of the understandability of the API’s signature, its documentation and the robustness of the API’s.
- Test the usability of the services’ privacy settings in relation to the user of a third party application that is implemented on top of the stack, e.g., the BigBang application.
- Test the efficiency (in terms of, CPU, memory, and battery consumption), and performance in demanding use cases (e.g., searching for location and navigation).

4. STUDY SETUP

This chapter presents the methodology followed in this thesis. The chapter begins with a theoretical discussion on the methodology, then presents its adoption within the scope of this thesis and finally lists the technologies used in implementing the application.

4.1 Basic Methodology

The primary nature of this work is to construct a software product in order to address a real life need. To be precise, a software product (BigBang) will be designed and implemented to address the need of microbusinesses and their consumers. Additionally, a privacy service MyGeoTrust will be integrated with the software to evaluate its usability under real life scenario. Therefore a constructive research approach is a good fit for this study.

By definition, a constructive research approach (CRA) offers a methodology that creates innovative constructions to solve real world problems and thus contributes to the field of study where it is applied [20][21]. This research approach is widely used within the paradigm of software engineering and computer science [19].

According to [20][21], an ideal model of CRA should encompass the following building blocks: (a) find a practically relevant (real-life) problem which requires solution; (b) obtain an understanding on the topic and on the problem; (c) design on innovative artifact that is intended to solve the original problem; (d) implement and demonstrate that the solution works; (e) make a theoretical contribution through carefully linking the solution to existing theoretical knowledge; and (f) examine the scope of applicability.

In practice, the steps do not follow each other in a simple sequence; rather the process is iterative and sometimes also recursive.

In relation to this model of CRA, both the applications within the scope of this thesis are intended to serve or solve real world needs, adopting innovative approaches to solve such needs and in turn offer intellectual contribution to the focused domain in-terms-of solution proposals and lessons learned.

4.2 Methodological breakdown of the Thesis

According to the goal of this thesis, the focus domain of the BigBang application is to support a very focused domain of business services and their consumers that are confined within a given geographical location. Other platforms with similar concept, e.g., Amazon [22], EBay [23], AliBaba [24], are targeted for global businesses and are not intended for small-scale startup businesses. Therefore, the problem domain of this thesis is a real-life and relevant need. To address this need, a design and prototype implementation is presented. To better support user's personal geo-tagged data privacy, MyGeoTrust service is integrated with it. Then an evaluation is carried out to test the pragmatic feasibility of MyGeoTrust. Therefore, the overall workflow of the thesis adhere the guidelines of Constructive research approach.

4.3 Technical Specification

This section presents the technical specifications and underlying rationale that forms the backbone of BigBang implementation. For the client implementation, Android platform is selected, whereas for the server side implementation, MySQL and StrongLoop framework are nominated.

Android:

For the client side implementation, Android platform [25] has been selected. The main motive behind such selection is the huge market share of Android at present time. According to netmarketshare [26] android holds a staggering of 60.39% mobile operating system market (see Figure 5), which make/made it an inevitable choice for this thesis.

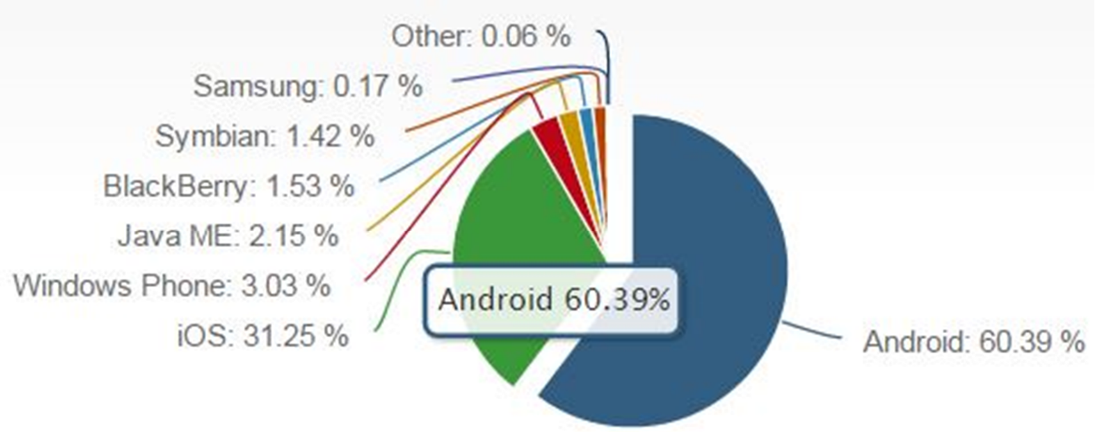


Figure 5 Market Share of Android Operating System [26]

Third party libraries:

In addition to the Android platform, several open source third party libraries are used to ease the development of the client application. A list of these libraries and their intended use are presented in Table 3.

Table 3 List of Third party libraries used

Library Used	Purpose
EventBus	For callback and event management.
EasyDialog	For Custom dialog display for long running events.
Picasso	For image display and associated memory management.
Android Image Slider	For image sliding.
Fabmenu	For animated floating menu items.

Following technologies and platforms are used to implement the server side functionalities.

MySQL:

MySQL is an open-source relational database management system (RDBMS). It is the second most popular and widely used RDBMS system and the most widely used open-source client–server model RDBMS [27][28]. Due to its open source license, MySQL enables the cost-effective delivery of reliable, high-performance and scalable Web-based and embedded database applications [27]. These benefits of MySQL make it a good fit for BigBang.

StrongLoop framework:

The StrongLoop framework (currently acquired by IBM and released under the name IBM API connect) is used to create high-quality, scalable and secure APIs for application servers, databases, enterprise service buses (ESB) and mainframes [29]. Therefore this framework is used in this project to expose REST API to connect, query the database in a secured way.

5. IMPLEMENTATION

This chapter presents the implementation details mostly concentrating on the architectural design, component interactions, and its integration with MyGeoTrust project.

5.1 The System Architecture

The overarching architecture of BigBang application follows client-server architectural design. This is an obvious choice, as the platform requirements require a server to store data, which will be accessed and manipulated through client application. For instance, user can create and manage user account in the server through the android client application. The basic building blocks of this architecture are shown in Figure 6, where the left side presents the android client application architecture and the right side presents the server.

On the client side, the architectural design is predominantly influenced by the MVC (Model-View-Controller) design pattern imposed by Android studio. Therefore, all the packages within the client application (see the right side of Figure 6) are organized according to this design pattern. A brief discussion on the fundamentals of MVC architecture is presented bellow to refresh readers’ memory before proceeding with client side architectural detail.

The MVC is a software design pattern for developing applications. A Model View Controller pattern is made up of the following three parts:

Model - The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

View - A presentation of data in a particular format, triggered by a controller's decision to present the data.

Controller - The controller is responsible for responding to user input and performs interactions on the data model objects. The controller receives the input; it validates the input and then performs the business operation that modifies the state of the data model, which in turn updates the corresponding views.

MVC is popular as it isolates the application logic (the model) from the user interface layer (the view) and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response.

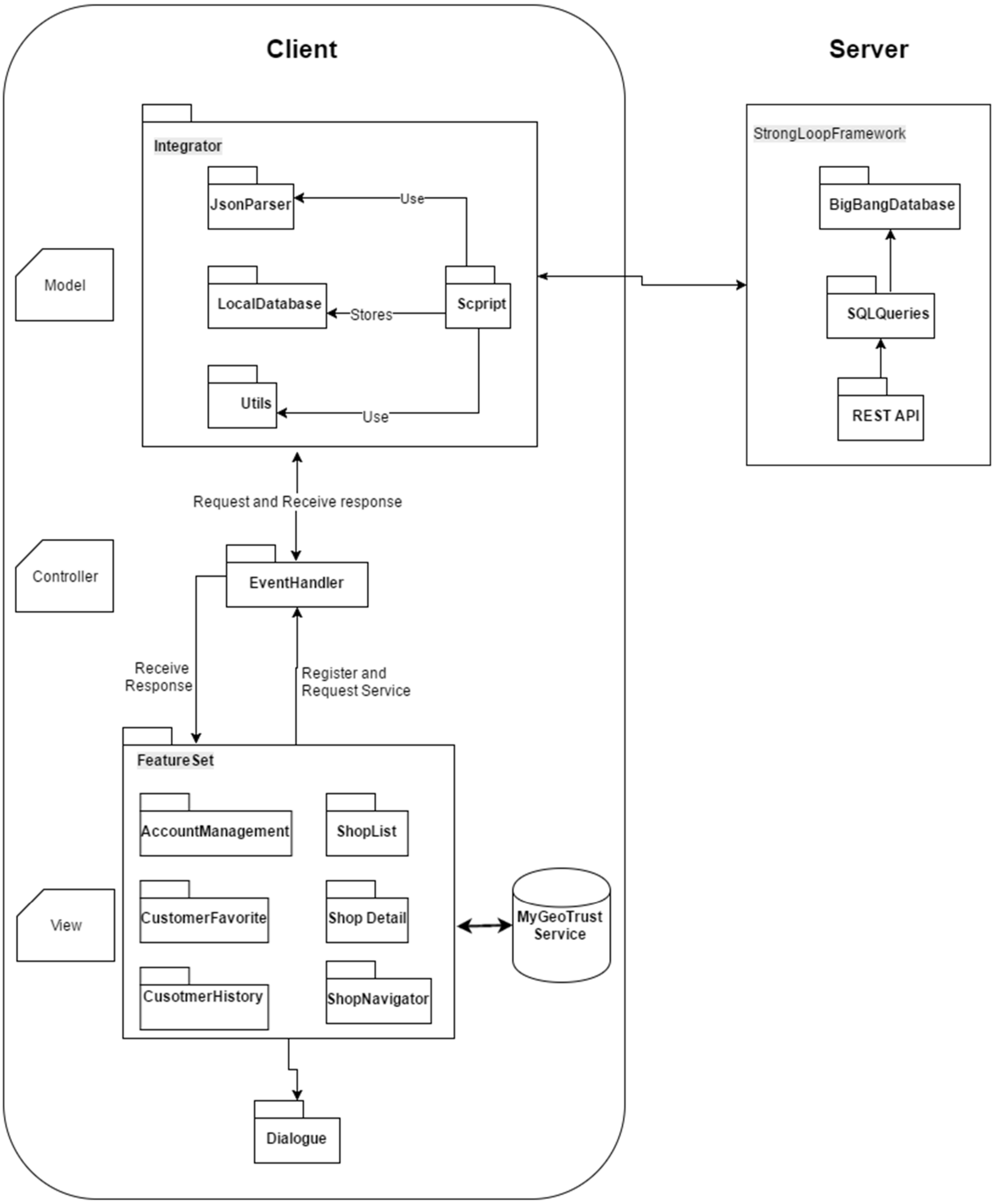


Figure 6 The overall architecture of BigBang platform

In relation to this idealistic model of MVC pattern, the **View** of the client application consist all the graphical interfaces and associated elements that are responsible for rendering data for the user. This includes displaying the data requested by the user, update data upon user input through the view and updating the view accordingly. As can be seen from Figure 6, all views are organized under individual java packages based on their intended functionalities. For instance, **ShopList** package is responsible to render shop lists under a selected category; the **ShopDetail** package is responsible for displaying a shops description and associated communication and navigation options; whereas the **ShopNavigator** package is responsible for providing real time navigation service to a selected shop; the **CustomerHistory** and **CustomerFavorite** packages are responsible for storing and retrieving browsing history and favorite list under an user account. All these feature specific packages are grouped under the **FeatureSet** package. The Dialogue package contains customizable dialogue to inform user graphically that a time-consuming background task is under processing. For instance, while reading list of shops and their details from the server under a certain category, a dialogue will appear to notify the user that the request in currently being processed. This dialogue package is used by each of the packages listed under **FeatureSet** package.

The **Model** of this application, as per the definition of MVC pattern, is responsible for the followings: (a) initiating network calls using REST APIs to exchange data between client and the server; (b) parsing, storing or updating data in the clients' local database and (c) notifying the View through the Controller. Upon receiving request from the **Controller**, the **Script** package executes an asynchronous network call using appropriate REST API. For instance, if user request for list of shops under Cloth category then a REST API to retrieve list of shops under that category will be initiated by the Script package. Upon receiving response (in JSON format) from the server the Script package parses it using the **JSONParser** package. The parsed information will then be stored in corresponding local database. This database is implemented in **LocalDatabse** package. In relation to the above example, the Script package will receive a JSON response holding shop list information, which is then parsed by the JSONParser package and stored in the shop list database. Finally, the controller will be notified on the completion of the task, which in turn will inform the corresponding view to update.

The Controller of this application consists of the **EventHandlers** that are responsible for registering callback request by the Views, and delivering the request to the Model for execution. Upon receiving the notification of completion from the Model, the Controller informs the View using the corresponding callback request. For example, consider that user taps a shop in the shop list view. Then this view will register for a callback event to the Controller and will pass the shop name and id with detail information will be retrieved. The shop detail will be displayed by the ShopDetail view. The Controller upon receiving the call back registration and the arguments, forwards the request to the Model. The Model then takes necessary action to retrieve the information, as presented in the previous paragraph. Upon completion, the Model notifies the Controller, and the Controller informs the **ShopList** view. This view then opens the ShopDetail view with the retrieved shop detail.

The integration of the client with MyGeoTrust service is done through the View package. This arrangement is shown in Figure 6 and Figure 7. Every Android application has a launching or main view (typically known as MainActivity) to launch the application, and for BigBang application the launching activity is the ShopList. Within this activity, the application binds to the MyGeoTrust service. This means, for the first launch of BigBang application, it sends a bind request to the MyGeoTrust service. The service in response starts itself (if not already running) and returns a reference to it. This reference is then used by the client application for further communication with the service.

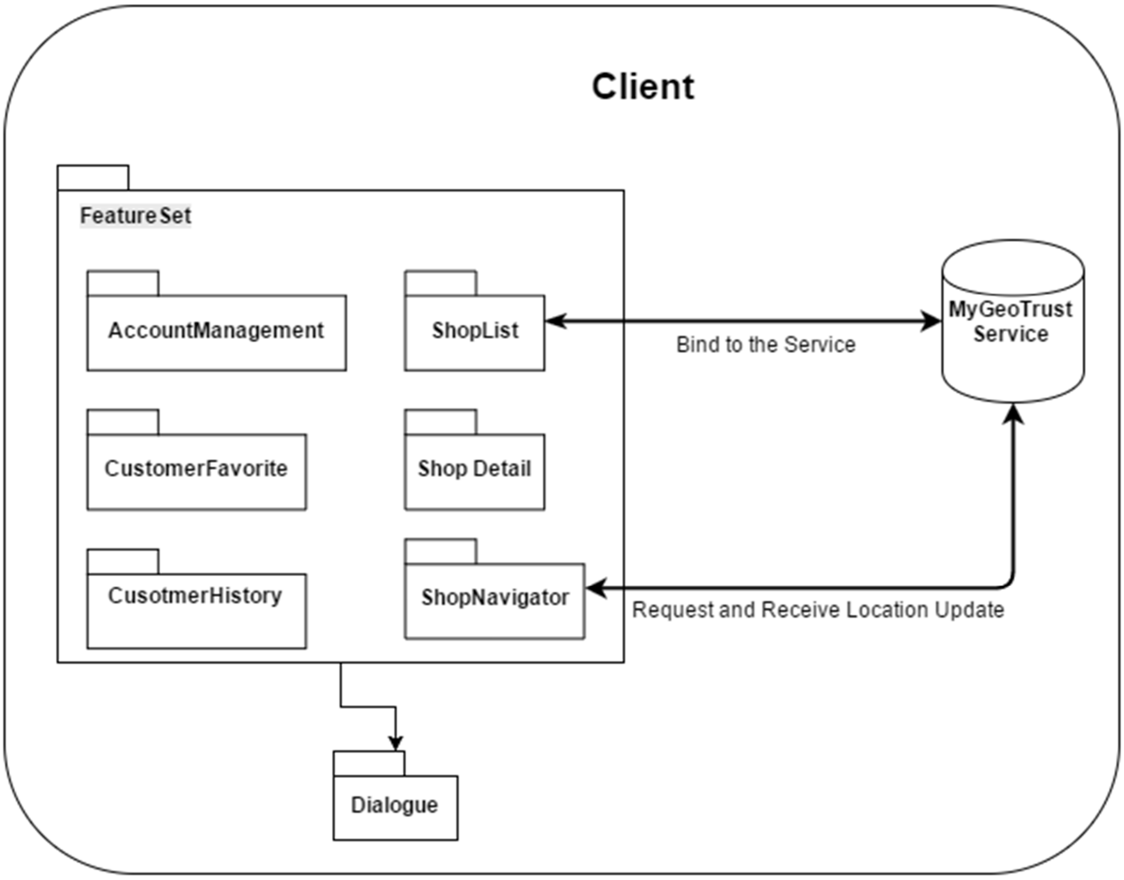


Figure 7 Integration process with MyGeoTrust

Now the APIs to get location update for real time navigation is implemented in the **ShopNavigation** package. User can open the navigation view in this package from the shop detail view by clicking a navigation button. The navigation view then opens the Google map and plots the shop location on it. If user ask for a turn-by-turn navigation then this navigation view will register for a continuous location update to the MyGeoTrust service, and in return the service will send location update (corresponds to user current location) to the view. The view will then use this information to navigate user through the route. Figure 7 presents in brief this overall arrangement. A detail discussion on this is presented in Section 5.5.

5.2 The Database design

This section describes the current design of the database in order to implement the application features with hints for future extension. Figure 8 presents the database as an Entity-relationship diagram.

The BigBang_Users table is used to create and manage user accounts. The StrongLoop framework for the implementation of token-based authentication uses several fields in this table. This authentication mechanism is/has been discussed further in next section. Among others, this table stores user first and last name, unique email address, hashed password and token. This table will be used to handle both Business and consumer accounts in future.

Tables holding personalized information for both business and the consumers should have a relationship with this User table. In the current implementation, users browsing history and favorites list are associated with a user account. Therefore, these two tables (e.g., User_History and User_Favorite) are linked to the BigBang_Users table. Both these table holds a mapping between userId and shopId with a timeStamp filed in History table to indicate when that user visited the shop page. These tables have one-to-many relationships with User table.

The Shop_Category table holds the categories under which a business can be hosted. New categories can be added or edited in this table, which in turn will be populated dynamically in the client application.

Shop / business general information under a certain category are listed in the Shop table. This table holds information like shop Id, name, shop type, website address, email address, phone number among others. Due to obvious reason, a one-to-many relationship holds between Shop_Categroy and Shop table.

In order to manage additional information regarding the business / shop necessary information will be added as an additional table and linked to the Shop table. Current implementation has a Shop_Address table and a Shop_Opening_Hour table, which holds fine-grained detail of a shop in relation to its geographical location and opening hours over the week. These tables have a one-to-one mapping with the Shop table.

Consistency within the database is maintained by imposing cascade dependency among the table relationships. Therefore, any modification within the database (either delete or update) keeps the entire database in a consistent state. For instance, if a user entry is removed from the User table then their corresponding entries from User_History and User_Favorite table will be removed. This is analogous to the transaction concept of database.

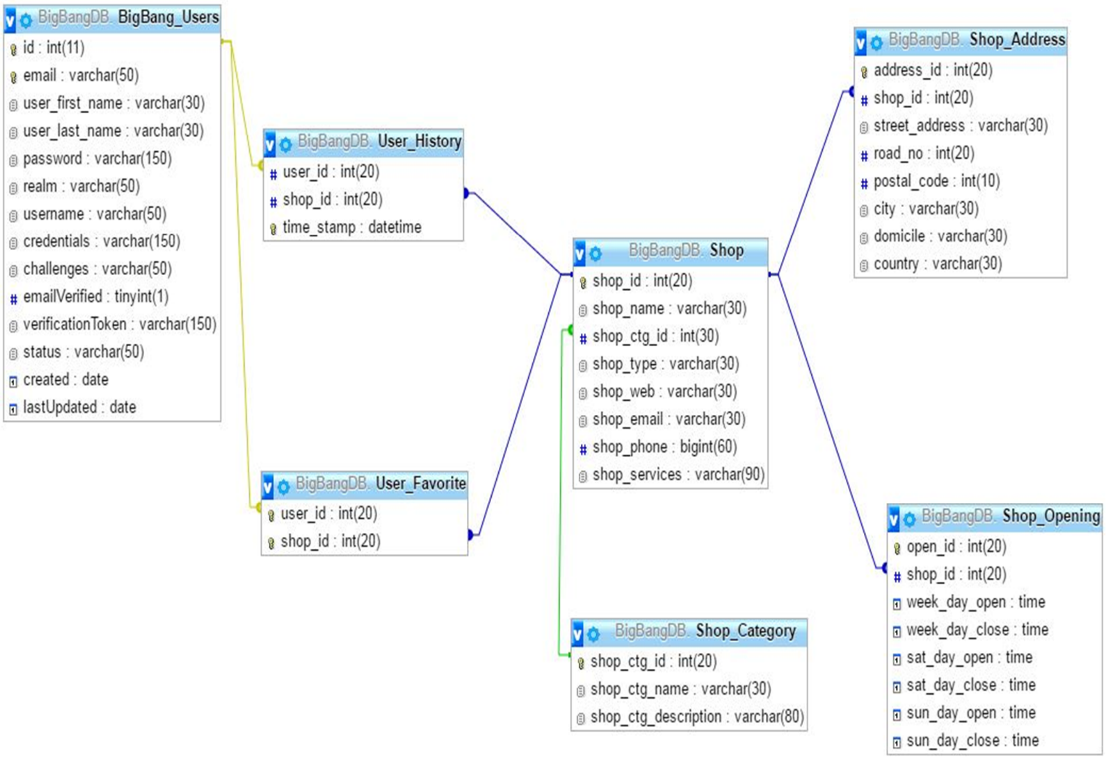


Figure 8 ER diagram of BigBang database

5.3 StrongLoop Integration

StrongLoop framework is used to implement the followings: (a) to integrate the database with the client, (b) to Expose REST APIs to execute database queries, (c) to create custom REST APIs to execute customized queries and (d) to efficiently manage user authentication and personalization. This section offers an elaborated discussion on these topics.

5.3.1 Database Integration

In order to integrate StrongLoop with the MySQL Database following procedure need to be followed.

First, a server application needs to be created. Instruction for creating a StrongLoop application can be found behind the link [30]. Once the project is created, the next step is to connect this project with the MySQL database. This can be done through command line interface or using StrongLoop web interface. Both approaches execute the same steps, which includes, providing the database credentials, providing the MySQL database name to be connected, selecting and installing the MySQL driver, and providing a name for the database to be used by StrongLoop. A detail documentation of these steps can be found here [30].

Once the connection is established the next step is to discover model. This step basically maps each MySQL table to a corresponding StrongLoop model. These models are then used to create and expose defaults REST APIs by StrongLoop. Detail on model discovery can be found in the given link [30].

5.3.2 Exposing and Using Default REST APIs

While the models are discovered successfully, StrongLoop will create and expose default REST APIs that corresponds to each table in the MySQL database table. Such APIs offers all HTTP operations, including GET, POST, PUT and DELETE with wide range of SQL queries within a given table. The parameters accepted by these operations or response returned by the query is by default in JSON format. For example, consider the Shop table in the BigBang database. A GET API exposed by StrongLoop that queries this table is

http://149.202.48.57:3000/api/Shops

Executing this API will return detail of all the shops present in the Shop table. Customization can be achieved by retrofitting query parameters with this API. For instance, detail of the shop having shopId = 1 can be retrieved using the following API,

http://149.202.48.57:3000/api/Shops/1

5.3.3 Creating and Exposing Custom REST API

Custom REST APIs for complex queries can be created in different ways. For instance, a custom complex query that joins multiple tables in the database can be done using either (a) StrongLoop wrapper methods that generate raw SQL implicitly, or (b) writing the raw SQL query manually and exposing it against a given REST URL. In this thesis the second approach is adopted as it provides more control from developers' perspective. Figure 7/10 presents the code snippet that implements a raw SQL query to retrieve all the shops along with their address and opening hour detail under a given shop category. A representative query can be,

Find all the shops, their general information, detail address and weekly opening hours under Cloth category.

```
module.exports = function(ShopCategory) {
     ShopCategory.categorywiseShopInfo = function (id, cb) {
        var ds = ShopCategory.dataSource;
        var sql = "SELECT Shop.shop_name, " +
                             "Shop_Address.street_address,
                              Shop_Address.road_no,
                              Shop_Address.postal_code, Shop_Address.city,
                              Shop_Address.country, " +
                             "Shop_Opening.week_day_open,
                              Shop_Opening.week_day_close,
                              Shop_Opening.sat_day_open,
                              Shop_Opening.sat_day_close,
                              Shop_Opening.sun_day_open,
                              Shop_Opening.sun_day_close " +
                      "FROM Shop, Shop_Category, Shop_Address, Shop_Opening "
                      +
                      "where Shop_Category.shop_ctg_id = " + id  +
                      " AND Shop_Category.shop_ctg_id = Shop.shop_ctg_id " +
                      "AND Shop.shop_id = Shop_Address.shop_id " +
                      "AND Shop.shop_id = Shop_Opening.shop_id";

      ds.connector.query(sql, function (err, ShopCategory) {
            if (err) console.error(err);
            cb(err, ShopCategory);  });
      };

     ShopCategory.remoteMethod(
        'categorywiseShopInfo',
        {
            accepts: [ {arg: 'id', type: 'number', required: true}],
            http: { path:'/categorywiseShopInfo/:id', verb: 'get' },
            description: 'Get the list of shops under a given shop category
                          identified by the category id',
            returns: { arg: 'categorywiseShopInfo', type: 'array'}
        } );};
```

Figure 9 Exposing Custom REST API for custom query using RAW SQL

In order to implement such queries in StrongLoop using approach (b) following needs to be performed:

Within the StrongLoop project a Java Script module is created for each model that corresponds to a MySQL table. The ShopCategory.js represents ShopCategory model. Within this script file create a function, which will contain the raw SQL to be executed. In Figure 9, the function named categorywiseShopInfo implements the above example query. This function is then exposed against a REST API, which is configured using the remoteMethod() function. As can be seen in Figure 9, the rest API is exposed as /categorywiseShopInfo/:id URL. The required parameter is the shop category id, which is a number, and the operation / verb type is GET. The response format is set as a JSON array with the array name as categorywiseShopInfo.

Executing this method will expose a REST API, which looks like the following,

http://149.202.48.57:3000/api/ShopCategories/categorywiseShopInfo/id

5.3.4 Token based Authentication and API protection

To implement a token based user authentication (e.g., user login) and exposing secure REST APIs that requires token based authentication, the User table of the MySQL database needs to be integrated with the User model generated by the StrongLoop project.

In order to perform this integration, the User table (BigBang_User table for this project) in MySQL database needs to have all the columns with the exact naming and types as in the User model. These fields are shown in BigBang_User table in Figure 6/9. While the integration is successful, StrongLoop exposes PUT or POST REST APIs to insert new user account information with hashed password, among others.

Once a user account is created using either PUT or POST REST APIs, the users can use their credentials to login. This login is done using Login REST API. While login is successful an access toke that corresponds to the user account is returned. This token is valid for a preset amount of time, after which user needs to re-login to renew the token. This token is then used to access secured REST APIs that requires access token to validate a user’s authority to access the API. A representative example of such API would be to access user’s transaction information.

5.4 MyGeoTrust Integration

This section describes what API set from MyGeoTrust service is used along with its integration procedure.

5.4.1 MyGeoTrust API usage

Table 4 summarizes the APIs that are used in this project along with their purpose and usage scenario.

Table 4 Tentative API usage scenario

API	**isBound()**
Purpose	Checks if the application is ready bound to the service or not.
Usage Scenario	Check whether the client is already bound before initiating bind service call.

API	**BindService(context, context)**
Purpose	Bind the given application context with the service. It also starts the service if not already started.
Usage Scenario	Request to bind the client application with the service.

API	**unbind(context)**
Purpose	UnBind / unregister the given client from the service.
Usage Scenario	Unbind from the service when the application terminates.

API	**onServiceBind(boolean b)**
Purpose	Callback methods returning the status of service bind request with BindService(..). In order to get this callback, the class needs to implement ***IMyGtServiceBinder*** interface.
Usage Scenario	Receives response on bind request status. If the bind is successful, the application starts.

API	**requestLocationUpdates(long minTime, float minDistance, IMyGtLocationListener listener)**
Purpose	Request the service for continuous location update with the given time interval and distance.
Usage Scenario	Request for location update for plotting user current location on the map and also to initiate real time navigation.

API	**removeLocationUpdates(String lilstenerID)**
Purpose	Removes the client from getting continuous location update
Usage Scenario	Stop location update while done. E.g., while user current location is received or user reaches to their destination.

API	**onLocationChanged(Location location)**
Purpose	Callback method getting location update when there is a location update received by the service. In order to get this callback, the client needs to implement the *IMyGtLocationListener* interface.
Usage Scenario	Receive the location update to plot user current location on the map. During real-time navigation use the location update to move user location marker along the root and calculate user distance from the next turn to re-arrange the map.

API	**isGPSAllowedInProfile()**
Purpose	Returns whether GPS is allowed by the current active profile settings.
Usage Scenario	Used to check if the GPS is allowed by the current profile before requesting for location update. If not then ask user to allow it.

API	**isProviderEnabled()**

Purpose	Returns whether GPS is turned on in the device.
Usage Scenario	Used to check if the GPS is turned on in the device before requesting for location update.

5.4.2 MyGeoTrust Integration

The first step in integrating MyGeoTrust service as part of a third party application is to import the library from Maven repository [31] using Gradle build system [32].

The next step is to bind to MyGeoTrust service using the BindService() api/API call. The bind request status will be returned in onBind() callback method. If the bind is successful then the service will be up and running (if not already running) and the callback will get an affirmative response. While bind is successful the third party application can invoke and use the APIs exposed by the service.

Application can unbind from the service at any given time by calling the unbind() API.

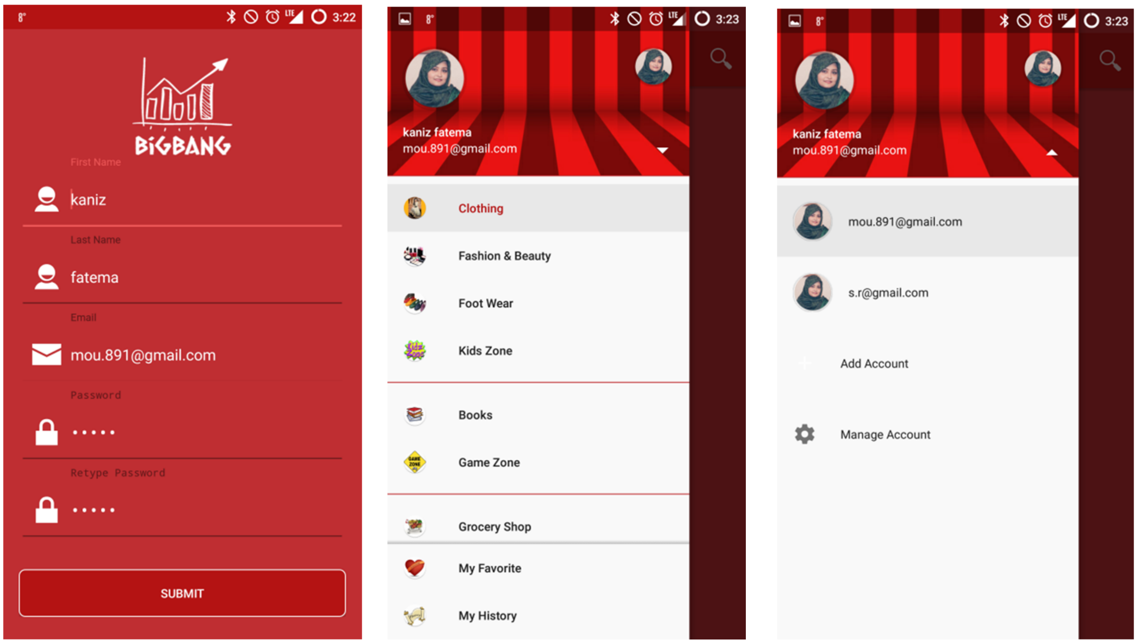
It is worth noticing that there is only one instance of MyGeoTrust service running in the device and serving all applications that bind to it. The service shuts down automatically while there is no application bound to it.

5.5 BigBang User Interface and General Application Flow

With the first boot up of BigBang application, it asks user to create an account. This is a compulsory step to proceed with the application, as both business and consumer can get personalized services, which require user account. For instance, a business will have their personalized business page to promote business offerings, can communicate with users who register to be notified, whereas consumers can favorite a business or browse their history with others. Users are allowed to create multiple accounts and manage them using the same interface, as shown in Figure 10(a).

When the account is successfully created the application runs with the account as the default active user account and populates all the categories under which a business can be launched in a Navigation drawer (Figure 10(b)). Favorite list and history associated with a user account can be viewed from the bottom two options on this drawer.

Additionally, this navigation drawer provides an interface to create a new account, manage / update an existing account or switching between accounts, as shown in Figure 10(c). User’s personalized information is updated with the switching of user account on this interface.



First Name
kaniz

Last Name
fatema

Email
mou.891@gmail.com

Password
•••••

Retype Password
•••••

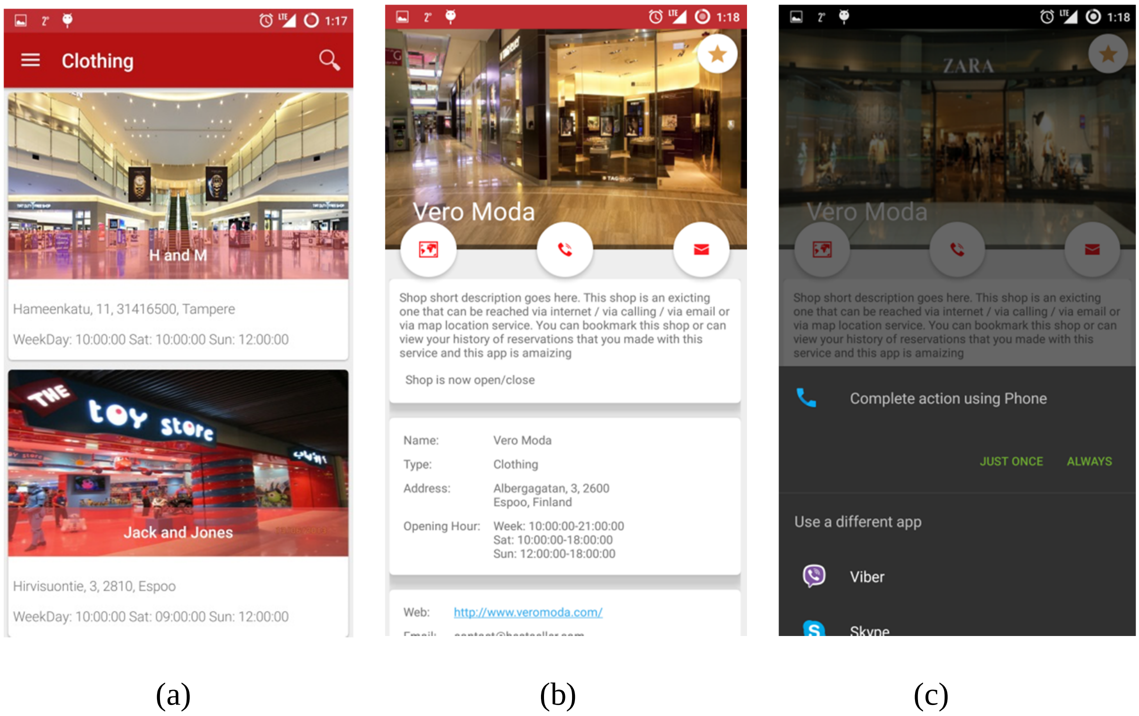
(a) (b) (c)

Figure 10 (a) User account create or manage interface, (b) Business category list interface with user current active account on top, (c) User account create or manage interface

Tapping on a business category in the interface shown in Figure 10(b) opens the list of shops under that category in the main interface. This is shown in Figure 11(a). For each shop this list shows information including shop name, full address and opening hour during the week.

Tapping on a shop on this list would navigate to the shops' business page (Figure 11(b)). This page holds the detail for the shop. The current implementation includes the following information, shops promotional picture gallery and description, calculate the opening hour and notify user whether the shop is currently open or not, shops full contact information (e.g., name, address, opening hour, website address, phone number, email), and listing of all the products that are available on the shop.

In addition to the above information, the shops' business page allows users to communicate with the shop in following ways, (a) make a phone call, (b) email to the shop, (c) navigate to the shops personal website, (d) add this shop in users favorite list, (e) open custom navigation service to navigate to this shop from users current location. All these options are made available through a set of self-descriptive floating buttons. A sample use of a phone call option is shown in Figure 11(c).



Hameenkatu, 11, 31416500, Tampere

WeekDay: 10:00:00 Sat: 10:00:00 Sun: 12:00:00

Hirvisuontie, 3, 2810, Espoo

WeekDay: 10:00:00 Sat: 09:00:00 Sun: 12:00:00

Shop short description goes here. This shop is an exicting one that can be reached via internet / via calling / via email or via map location service. You can bookmark this shop or can view your history of reservations that you made with this service and this app is amaizing

Shop is now open/close

Name:	Vero Moda
Type:	Clothing
Address:	Albergagatan, 3, 2600 Espoo, Finland
Opening Hour:	Week: 10:00:00-21:00:00 Sat: 10:00:00-18:00:00 Sun: 12:00:00-18:00:00

Web: http://www.veromoda.com/

Complete action using Phone

Use a different app

(a) (b) (c)

Figure 11 (a) List of Businesses under Clothing category, (b) Detail of a Business (Business home page), (c) In-App communication with the business

Now tapping on the navigation button (the button with a map icon in Figure 11(b)) would open the navigation window, as shown in Figure 12(a). This window plots the shop's location on the map with a bubble marker and provides option for getting the driving route from user current location to the shop and real-time turn by turn navigation service. These service options are provided under a floating menu as shown in Figure 12(a).

Tapping on the Get Route menu option on this window will first ask for user current location. While user's current location is acquired, it will then request for the driving route between users current location and the shop. Upon receiving the route information it is plotted on the map. Additionally it displays the total distance and time required to travel to the shop. All these features are demonstrated in Figure 12(b).

While the root is received, user can initiate real-time navigation by tapping the Start Navigation menu option. This will start the turn-by-turn navigation as shown in Figure 12(c). During this navigation the application will take continuous location update to plot user's current location along the route (shown by the arrow location marker in Figure 12(c)). As the user moves along the route, the location marker will converge towards the next turn in the route. The application will show the distance between user's current location and the next turn on the top right corner of the window. While user reaches within 20 meter of the next turn, the map will be rearranged to show the next turn in the route. This process will continue until user reaches to their destination. The navigation stops automatically once the destination is reached.

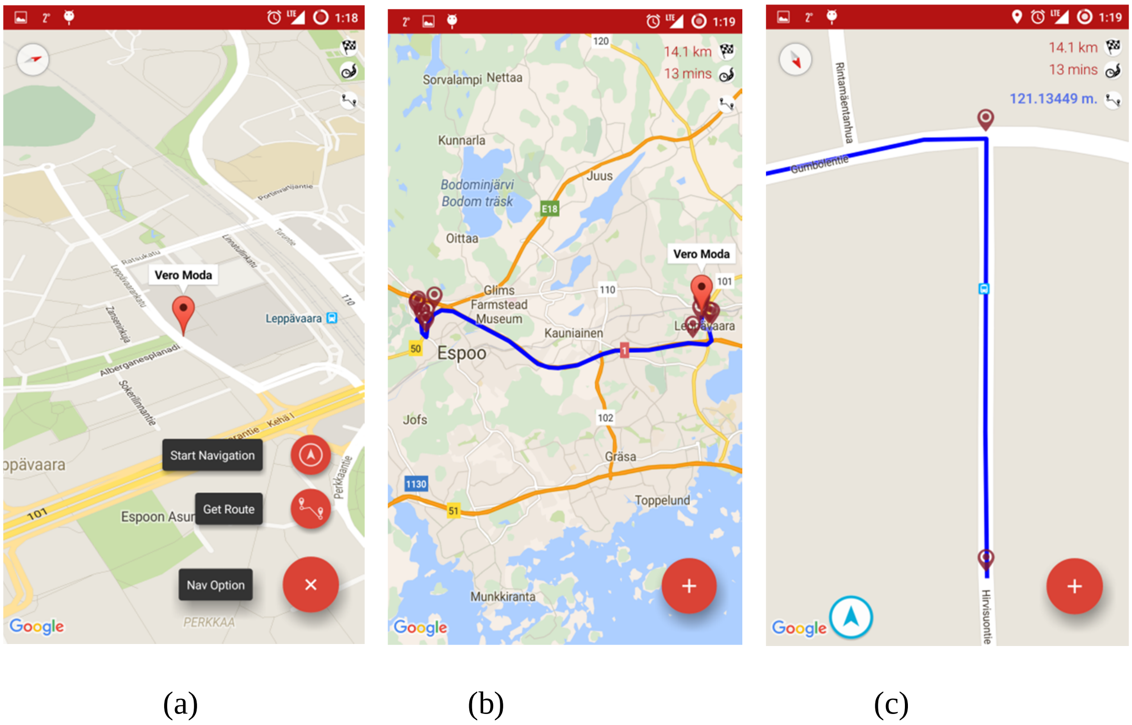
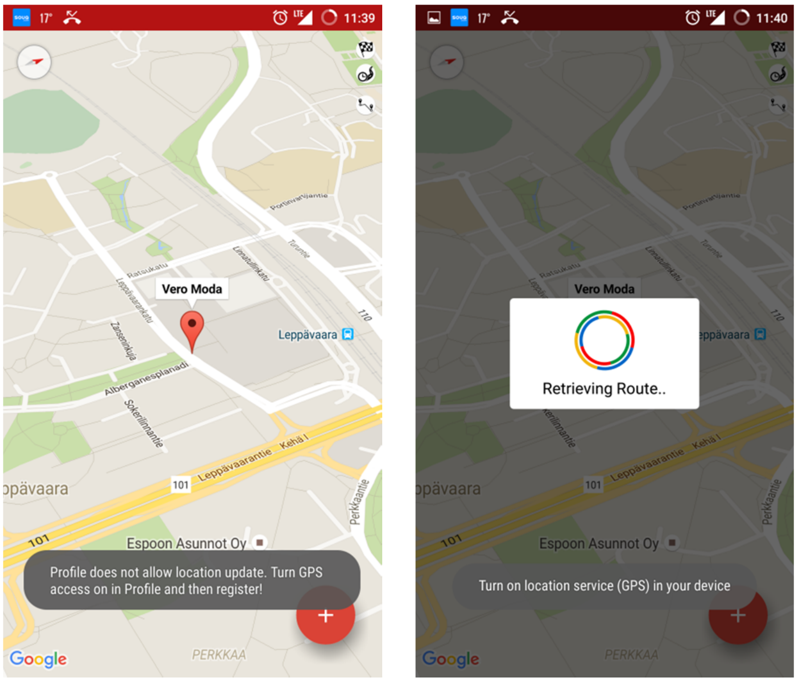


Figure 12 (a) Navigation window showing the shop location on map with navigation options, (b) Driving route plan from users' current location to the shop, (c) Turn-by-turn real time navigation service

Point to be noted here is that the navigation service, as described above will only work if the location service is permitted by the current privacy profile set in MyGeoTrust service and the GPS settings in the device.

To guide user appropriately, the application will prompt user appropriate messages regarding the privacy settings and the device settings. For instance, while asking for route to the shop, if the application finds that location service is disabled in current profile settings, then it will prompt user a message asking for allowing the service in MyGeoTrust (as shown in Figure 13(a)). Similarly, if the GPS is turned off in the device while requesting for location update for the navigation, similar Toast message will be prompted (see Figure 13(b)).



Profile does not allow location update. Turn GPS access on in Profile and then register!

Retrieving Route..

Turn on location service (GPS) in your device

(a) (b)

Figure 13 (a) Location Service not allowed in Current Profile message, (b) GPS not turned on in the device

6. ANALYSIS

Study on the usability of MyGeoTurst service reveals mixed result. This chapter discuss in detail both pros and cons that are encountered during the integration and usage of the service as part of BigBang application. Additionally, suggestions on future improvement on the service are documented.

6.1 Pros

There are several positives noted while using MyGeoTrust service. This section throws light on those.

6.1.1 Standard Library Integration Process

The integration process of MyGeoTrust service as part of a third party application (e.g., the BigBang) follows the standard and well anticipated approach. There are two different approaches available, for instance (a) download and compile the library from Maven repository, or (b) manually include it in the project and compile it. The earlier approach is temporarily unavailable due to development. The latter is presented below,

First, the library (i.e., mygtlib-debug.aar) needs to be downloaded from project's git repository and include it in the projects lib directory.

Second, update the Gradle build file of the application to compile this library with the project. Following code snippet (in Figure 14) illustrates this step.

```
//Include the libs folder in the directory list
repositories {
    flatDir {
        dirs 'libs'
    }
}

dependencies {
    //compile the library with gradle build
    compile(name:'mygtlib-debug', ext:'aar')
}
```

Figure 14 APPs gradle file modification to compile MyGeoTrust library as part of BigBang application

6.2 Standard API

In terms of API signature, all the APIs exposed by the MyGeoTrust service follow the standard. The API naming is clear and representative of their intended purpose. No further documentation is needed for clarification. The argument lists for these APIs are also concise yet informative enough to interpret. For instance, consider the location update API presented in Table 4.

```
requestLocationUpdates(long minTime, float minDistance, IMyGtLocationListener
listener)
```

The method name requestLocationUpdates signifies that the API is intended for continuous location update. While the argument list indicates that it needs to have the minimum time interval at which the location update will be sent to the caller, the minimum accuracy required in location update and a reference to the caller which implements the callback to get location update.

6.2.1 Correspondence with Android Native API

In addition to standard signature, the APIs also mimic their counterpart from native Android. For instance, the location update API discussed in previous section is almost identical to its' Android counterpart, which makes it trivial for the developers who have experience in using Android native API to get used to MyGeoTrust. This observation holds for other APIs exposed by the service.

In addition to this, implementing interfaces in order to get callback methods are almost identical in most cases. For instance, the onLocationChanged() callback method to receive location update (as described in Table 4) is identical to its Android counterpart.

6.3 Cons

There are several issues noticed regarding the integration and performance of the service, which are listed below.

6.3.1 Unnecessary Thread Creation

Use of continuous location updates API from MyGeoTrust service (fifth API listed in Table 2) instead of Android native one resulted in unnecessary UI thread creation to update the user interface. As a demonstration of this issue, the following code snippet is presented. The method animateMarker() presented in Figure 17 is used to move user location marker smoothly on the map while navigating along the route from user current location to a shop (similar to google map navigation service). This is done within a thread using interpolation. As arguments, this method receives the marker icon and user’s current location to move the marker from previous to the given location.

In order to move the marker, a request for location update need to be registered with update frequency as presented in Section 5.4.1. In response, the location update will be received in onLocationChanged() callback method. For this location update, if Android native API is used, then the animateMarker() method can be called directly from onLocationChanged() callback method without the creation of any UI thread. This is shown in line 6 of Figure 15. The location marker in the animateMarker() is moved within a new thread (line 10 through 19 in Figure 17) and Android automatically handles the call from a callback method to a thread, which updates the UI without needing to create any additional threat.

```
 1    @Override
 2
 3    Public void onLocationChanged(Location location)
 4    {
 5      LatLng currentLocation = new
 6      LatLng(location.getLatitude(), location.getLongitude());
        animateMarker (navigationMarker, currentLocation, false);}
```

Figure 15 Code snippet to call animateMarker method from Android native onLocationChanged callback method

```
 1    @Override
 2    Public void onLocationChanged(Location location)
 3    {
 4       LatLng currentLocation = new LatLng(location.getLatitude(), loca
                                            tion.getLongitude());

 6       runOnUiThread(new Runnable() {
 7                    @Override
 8                    public void run() {
 9                        animateMarker (navigationMarker, currentLocation,
10    false);
                      }
         });}
```

Figure 16 Code snippet to call animateMarker method from MyGeoTrust onLocationChanged callback method

However, if MyGeoTrust API is used for this location update instead, then an additional UI thread needs to be created in onLocationChanged() callback within which animateMarker() has to be called. This is shown in Figure 16 (line 6 to 10). This is because, onLocationChanged() method in MyGeoTrust is a wrapper callback method.

This method does not have the same thread handling mechanism to that of Android native one. Therefore, Android throws an UI thread exception while animateMarker() method is called within the callback. This is inconvenient from developers perspective because this scenario is neither obvious for the developers to resolve nor desirable from standard development practices.

```
public void animateMarker(final Marker marker, final LatLng toPosition,
final boolean hideMarker) {
        final Handler handler = new Handler();
        final long start = SystemClock.uptimeMillis();
        Projection proj = mMap.getProjection();
        Point startPoint = proj.toScreenLocation(marker.getPosition());
        final LatLng startLatLng = proj.fromScreenLocation(startPoint);
        final long duration = 1000;
        final Interpolator interpolator = new LinearInterpolator();

        handler.post(new Runnable() {
            @Override
            public void run() {
                long elapsed = SystemClock.uptimeMillis() - start;
                float t = interpolator.getInterpolation((float) elapsed
/ duration);
                double lng = t * toPosition.longitude + (1 - t) *
startLatLng.longitude;
                double lat = t * toPosition.latitude + (1 - t) *
startLatLng.latitude;
                marker.setPosition(new LatLng(lat, lng));

                if (t < 1.0) {
                    // Post again 16ms later.
                    handler.postDelayed(this, 16);
                } else {
                    if (hideMarker) { marker.setVisible(false); }
                    else { marker.setVisible(true); }
                }
            } }); }
```

Figure 17 Method to move location marker smoothly on the map using interpolation

6.3.2 An Android Service

Currently MyGeoTrust is implemented as an Android Service, which in simplest terms is an Android application that runs background without necessary having an UI. However, a Service in Android is governed by the same rules and guidelines like any other applications. It includes for instance, the application lifecycle events, access permission right, resource consumption and Android OS monitoring and control over it.

MyGeoTrust being a nucleus for controlling privacy and security of crowdsourced data, needs master privileges (or root access) and need to be persistent like the Operating System itself. This is required because the service is the control point of resource access right of all the applications that run over it. However, being a service, it first needs to gain the access right of the resources for itself. With the current version of Android (API 23) user can revoke such access permission at any given time, which makes current MyGeoTrust implementation vulnerable. Because, with no access right on a resource (for instance, location service), how could MyGeoTrust provide location update to the applications?

Additionally, MyGeoTrust consumes resources like the other applications, and are exposed to the threat of being terminated or stopped by the Operating system in crisis situation. Even through such case of termination is slim as per android documentation, however, for a demanding service like this, it might occur if every possible resource intensive tasks are not handled and implemented efficiently.

6.3.3 Unified UI Style

From users' experience perspective switching back and forth between the BigBang application and MyGeoTrust service to initiate location update is daunting in demanding situation. For instance, consider the following usage scenario, where user in a real hurry to get to a shop and he is in a heavy traffic with few seconds to start the navigation service. Additionally, consider that the location service is not allowed in the privacy profile and the GPS is turned off in the device.

In this user scenario, while trying to start the navigation service, the user has to open MyGeoTrust profile settings to allow location update along with turning on the GPS in the device. Therefore, the process imposes an additional step, which is also not too straight forward (i.e., launch the service UI, open profile settings, navigate to the option settings, change and save it, and return back to BigBang application) in addition to conventional process (i.e., just turning on the GPS from taskbar). This is really cumbersome for users while in a stressed scenario.

Additionally, a change in the profile settings does not take into effect in many cases, e.g., user allows location update in the current profile settings but the service denies access to it. The exact scenario in which such malfunction occurs is hard to reproduce. However, it has to be resolved as it is not desirable from user experience perspective in any circumstances.

6.4 Recommendations

Recommendations for MyGeoTrust service for further improvement is boiled down to the following proposition,

MyGeoTrust privacy settings should be part of the Operating System (e.g., part of conventional Android Settings) rather than a standalone Android Service.

The discussion hereafter is provided in support of the above proposition and in relation to the Cons presented before.

According to the mandate of MyGeoTrust privacy design, the service should acquire the sole authority over all the system resources that in some way relates to users' privacy. Such resources include for instance, GPS, Wi-Fi, microphone, Contact Info, Phone Storage, Bluetooth and others. All third party applications that bind to this service would request permission to this service for accessing certain resources. Therefore, individual application does not get any access right (contrary to conventional practice) rather rely on the service to get access. Users can grant permission to this application through service privacy settings. A control of this scale requires supreme authority (popularly known as Root permission) to work effectively.

However, MyGeoTrust in its current form is implemented as an Android Service. By definition, a service in Android is a background application which may run standalone or in relation to other client application. Therefore, it has a lifecycle and access right similar to any other application in Android, as discussed before. This raises the first design and implementation concern. Because, Android OS reserves the soul authority to control this service like any other application. For instance, it can terminate or kill the service if its' decides so without any prior notice. Therefore, it is practically impossible to retain control over other applications with such vulnerability of its own lifecycle.

The second concern rises from the fact that an application can request access permission for resources either during its installation process or at runtime (only for API level 23). While granted by the user, the application retains this access right forever or until user revokes it (only for API level 23). Being an Android service, it is very difficult for MyGeoTrust to keep trace of which applications are gaining access right bypassing the service, and abusing such rights. Additionally, user can revoke access right of the service itself (either intentionally or unintentionally) like any other application which would make it practically unusable.

The third concern is the resource consumption while the service is running, analogous to any other application. One of the main tasks of the service is to crowdsource geotagged user data, which requires use of GPS and Wi-Fi in the device. As improper use of these resources (e.g., using GPS for long time and in high frequency) can consume significant battery power and CPU time, therefore adequate intelligence need to be developed to ensure efficient use of these resources. However, this task is not trivial to implement.

Finally, the service in its current form is exposed to issues listed in the previous section. This includes for instance, unnecessary thread creation, and low user experience. These also raise concern on the usability of the service.

According to our understanding and evaluation the above discussed issues can be effectively remedied if MyGeoTrust is implemented as part of the Operating System. The settings of Android OS can be effectively extended to incorporate the privacy model of MyGeoTrust. There are several pragmatic reasons backs this proposal, which includes but not limited to the followings,

First of all, this idea of embedding privacy settings or user control of resource access by the application is not a new idea. There are several wrapper implementations of Android Native OS that offers such features to a certain extent. For instance, consider the Oxygen operating system developed by OnePlus on top of Android 5.1 (acronym Android Lollipop). This operating system lists all the permissions granted to each application running in the device under App permissions settings tab. In this tab, user can not only view for each application, what permissions it currently has and when it last used those permissions, but also can revoke or grant permission at runtime. Therefore, user can track whether an application has abused permission through the verification of last access time.

This feature can be matured further to show user statistics on when an application uses certain permission while the user does not use the App. Additionally, user can register for explicit warning and notifications for apps abusing access permission frequently. The OS settings can also suggest user with possible remedy in order to encounter such misuse.

Secondly, the data crowdsourcing feature can extend the Android data crowdsource mechanism (for instance, crowdsourcing while user using the navigation service), which will make it a robust and efficient system in terms of resource management.

Thirdly, being part of the OS, the privacy model will enjoy root privilege over all applications and not vulnerable to be terminated / shut down by the OS.

Fourthly, user can control permissions per application basis rather than one-size-fits-all model of current MyGeoTrust service. This offers fine-grained control over applications.

Fifth, Android native APIs can be used as it is instead of creating wrapper methods. This is beneficial in many ways, including unnecessary UI thread creation problem, serving multiple applications who requests for location or Wi-Fi scans simultaneously, along with many other issues related to concurrency.

Last but not the least, user experience can be enhanced in using these privacy settings. There are several reasons supporting this opinion, which includes, (a) privacy settings are integrated within the conventional settings, which is easily conceivable for the user, rather than habituating with an entirely new system, (b) user has a single point of reference to change the privacy settings that keeps current process of turning toggling access permission almost as it is. These will reduce the overhead of switching back and forth between applications.

7. FUTURE DIRECTION

There are several directions in which this application and its feature list can be extended and matured. This chapter discussed some of the prominent areas based on current progress and envisioned targets.

7.1 Application Feature

Currently this application allows generic user registration with token-based identification. Any user can create account and login with their credentials to view custom browsing history and favorite list. This feature will be matured further to offer business owners to carry out the followings,

(a) Create own business account.
(b) Launch a business portal against the account.
(c) Upload information to the business portal with information, e.g., business description, offers, product listing, sending notification to the registered customers.

In addition to that, each shop / business portal (e.g., the shop detail view in current implementation) should offer additional features, which should include but not limited to

(a) In-app booking and purchase of products.
(b) Categorical listing of a shop services.
(c) Option for shop owner to give promotional or seasonal offers on selected products.
(d) Customers can register for specific products / services of a shop to get notification (e.g., promotional offers).

On customers' part, the application will be extended to accommodate features like,

(a) Collecting points on in-app purchases.
(b) Provide constructive feedback, product ratings on shop services.
(c) Can compare related services from different service providers (i.e., businesses).

7.2 The Navigation Service

Current navigation service offers basic turn-by-turn navigation from user's current location to the shop in question. It also provides some useful real time information in relation to navigation. However, this navigation service should be matured further to meet real world need of the users. This must include the followings,

(a) Alternate routes from source to destination.
(b) Provide different route types to select from, e.g., walking, cycling.
(c) Automatic route recalculation. This is a must service that is required in real time situation, for instance while user misses a turn.
(d) Route should be plotted more user-friendly way (like google navigation) along with supporting information, e.g., which way to turn from next point, traffic information and others.

7.3 Secured Communication

Current implementation of the server communication uses only HTTP protocol, which is not a secured way. Next release should use HTTPS as this protocol is built to offer authentication of the website and associated web server with which one is communicating, which protects against man-in-the-middle attacks [33]. Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with or forging the contents of the communication [33].

8. CONCLUSIONS

This thesis is devoted to offer a conceptual platform and a workable implementation of it, which is targeted to micro businesses and their consumer base. Under the name of BigBang, this platform should offer small businesses that are located in a certain geographical area with defined services and their consumer base, a single point reference of interaction. The point of argument here is that such solution would lead to build a healthy business ecosystem where small business can advertise their services with an opportunity to compare with their rivals to provide better offers. Moreover, due to detersive business services launched within this platform, each business get exposure to a large consumer base, which is otherwise, would not be possible. On the other hand, for the consumers, this platform would be the virtual mega mall that brings together all the services that located around their vicinity.

Additionally, this thesis explores the opportunity to integrate BigBang platform with the privacy service named MyGeoTrust. The motive is to offer the users of BigBang a secured and trustworthy means in using location based services, e.g., the turn-by-turn navigation service. However, as MyGeoTrust is in beta stage, this thesis thus makes a study on the pragmatic applicability of this service.

The contribution of this thesis can be summarized as follows: (a) it offers a design and partial implementation of a platform for micro businesses, (b) it implements a workable version of customized navigation service that allows users to get turn-by-turn navigation to their desired shop, (c) it integrates the privacy service MyGeoTrust and performs feasibility study, and (d) it offers recommendations for future improvements of MyGeoTrust with a blue print of future goal for BigBang platform.

REFERENCES

[1] B. Headd, An Analysis of Small Business and Jobs, US. Small Business Administration, 2010, available at https://www.sba.gov/.

[2] Micro Business Statistics and Trends, 2013, available at http://microexec.com/2013/02/27/micro-business-statistics-and-trends/.

[3] R.E. Gunness, H. Kuusniemi, j. Vallet, T. Sarjakoski, J. Oksanen, M. Islam, M. Syeed, H-M. Halkosaari, P. Kettunen, M. Laakso, M. Rönneberg, MyGeoTrust: A patform for trusted crowdsoruced geospatial data, Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015), Pages 2455-2469, September 14-18, Tampa Convention Center, Tampa, Florida, 2015.

[4] MYGEOTRUST, last visited on May, 2016. Available at http://mygeotrust.org/ .

[5] Bigger than you think: The economic Impact of Micorbusinesses in the United States, last visited onMay, 2016, Available at http://microenterprisealabama.org/wp-content/uploads/2014/09/Bigger-Than-You-Think-The-Economic-Impact-of-Microbusiness-in-the-United-States-copy.pdf.

[6] Apple Inc. “Privacy Policy”. Apple website, September 17, 2014 version. Available: http://bit.ly/AppplePrivacy.

[7] Google Inc. (2014). “Google Terms of Service”. Google website, April 14, 2014 version. Available: http://bit.ly/GooogleTOS .

[8] Oksanen, J., Sainio, J. and J. Westerholm, 2014. Where to ride a bike? Methods for deriving privacy-aware heat maps from mobile sports tracking application data. Mobile Tartu 2014 - Mobile data, geography, LBS. 3rd International Conference and PhD Summer School, 1-3 July 2014, Tartu, Estonia.

[9] Chen, R., and R. E. Guinness, 2014. Geospatial Computing in Mobile Devices. Norwood, MA: Artech House, 208 pages.

[10] Bergman, C. and J. Oksanen, 2014. Conflation of OpenStreetMap and public Sports Tracker data for automatic routing. Mobile Tartu 2014 - Mobile data, geography, LBS. 3rd International Conference and PhD Summer School, 1-3 July 2014, Tartu, Estonia.

[11] Urban, J. M., Hoofnagle, C. J., and Li, S. (2012). Mobile Phones and Privacy, BCLT Research Paper Series, UC Berkeley Public Law Research Paper No. 2103405. Available at SSRN: http://ssrn.com/abstract=2103405 .

[12] Onsrud, H.J., 2003. Privacy in the Use of Spatial Technologies: Ethics as a Driver of Technological Research Priorities, NSF Confidentiality Workshop, May 12-13, 2003, Washington D.C. Available: http://www.urban.org/nsfpresentations/index.html .

[13] Bao, J., 2009. Privacy protected location based service. Masters Thesis. 2014. Auburn University, Alabama, United States.

[14] Dobson, J., 1998. Is GIs a Privacy Threat? GeoWorld, July 1998.

[15] Small Business at a Glance, last visited on May, 2016. Available at https://www.entrepreneur.com/page/216022 .

[16] Sokolovay, K., Lemercier, M., Boisseau, J., 2014.Privacy by Design Permission System for Mobile Applications. The Sixth International Conferences on Pervasive Patterns and Applications, PATTERNS 2014.

[17] Cavoukian, A. , Privacy by design: The 7 foundational principles, Information & Privacy Commissioner, Ontario, Canada, 2009.

[18] FTC Staff Report, Mobile Privacy Disclosures: Building Trust Through Transparency, Federal Trade Commission, USA, February, 2013.

[19] E. Kasanen, K. Lukka, and A. Siitonen. The constructive approach in management accounting research. Journal of Management Accounting Research, 1(5):243–263, 1993.

[20] K. Lukka. Konstruktiivinen tutkimusote: luonne, prosessi ja arviointi. Rolin, K., Kakkuri-Knuutila, M.-L., Henttonen, E. (eds.) Soveltava Yhteiskuntatiede ja Filosofia, Gaudeamus, Helsinki, pages 111–133, 2006.

[21] K. Piirainen and R. Gonzalez. Seeking constructive synergy: Design science and the constructive research approach. Design Science at the Intersection of Physical and Virtual Design, pages 59–72, 2013.

[22] Amazon, Available at www.amazon.com

[23] EBay, Available at www.ebaly.com

[24] Alibaba, Available at www.alibaba.com

[25] Android Developers, Available at http://developer.android.com/index.html

[26] Mobile/Tablet Operating System Market Share, NewMarketShare, https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qpsp=68&qpnp=2&qptimeframe=Q, 2016.

[27] MySQL, http://www.oracle.com/us/products/mysql/overview/index.html

[28] DB-Engines Ranking, http://db-engines.com/en/ranking/relational+dbms , Retrieved 12 February 2015.

[29] StrongLoop, https://strongloop.com/, Last visited May, 2016.

[30] StrongLoop getting started, available at https://strongloop.com/get-started/.

[31] The Central Repository, available at http://search.maven.org/.

[32] Gradle, available at http://gradle.org/.

[33] HTTPS Everywhere, last visited May, 2016. Available https://www.eff.org/https-everywhere/faq