DEGREE PROJECT IN COMMUNICATION SYSTEMS, SECOND LEVEL
STOCKHOLM, SWEDEN 2014

Social Proximity Indicator Application Enhancing Personalization for the User

A User Centric Multimodal Smartphone based Social Proximity Indicator

DIPESH MITTHALAL

KTH ROYAL INSTITUTE OF TECHNOLOGY
INFORMATION AND COMMUNICATION TECHNOLOGY

Social Proximity Indicator Application Enhancing Personalization for the User

A User Centric Multimodal Smartphone based Social Proximity Indicator

Dipesh Mitthalal

Master of Science Thesis

School of Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden

26 June 2014

Examiner: Prof. Gerald Q. Maguire Jr.
Industrial Supervisor: Christopher Howson (Technicolor R&D, Rennes, France)

Foreword

This work is presented in partial fulfillment of the requirement for the degree of Master of Science at the KTH Royal Institute of Technology, Stockholm, Sweden. This work was carried out at the Mobile and Interacting Devices division of Media Computing Lab (MCL) at Technicolor R&D, Rennes, France.

Abstract

The smartphone has become an important part of our daily life. It takes advantage of multiple built-in sensors, along with internet connectivity, to provide a variety of services including navigation, weather forecasts, media capturing/sharing, and many more. The smartphone has become a hub for our social interactions, ranging from traditional voice communications through messaging, to online social networking.

There are endless possibilities for a new generation of applications that automatically adapts according to users' social affinities. The key enabler is to understand the social profile of the user and to realize a Social Proximity Indicator (SPI). This indicator of a user's social profile includes the user's location, preferences, common friends, frequent interactions among friends, community membership, and many more attributes. This project addresses the exploitation of the user's smartphone as a detector of these user's social contexts in order to infer the social proximity between any 2 (or more) people. The goal of this social proximity indicator is to identify the (current) set of smartphone users that would want to participate in an application specific activity, such as media sharing, group conversation, etc. SPI identifies users based on their social interactions and common preferences.

Based on this SPI, a photo sharing application is proposed and built for the Android smartphone platform. This application uses multiple physical and virtual sensors (hosted by the user's smartphone) to determine the context of the user. After inferring this context, the application initiates photo sharing among an identified group sharing this context. The application, as well as the underlying code, theoretical concepts, and results are presented in this thesis. The experiments conducted during this thesis project indicate that it is feasible to build smartphone based social proximity based applications in which proximity incorporates much more than simply physical location, thus going beyond existing applications.

Keywords: Social Proximity, Smartphone, Social Interactions, Image sharing, Virtual Sensors

Sammanfattning

Smartphone har blivit en viktig del av vårt dagliga liv, som drar nytta av de många inbyggda sensorer, inklusive Internet-anslutning, för att ge en mängd olika tjänster. Däribland navigation, väderprognoser, media fånga/delning, och många fler tjänster. Smartphone har blivit kärnan i våra sociala interaktioner. Den erbjuder allt från traditionella röst kommunikation via meddelanden, till sociala nätverk.

Det finns oändliga möjligheter för nästa generationens program som kommer att automatiskt anpassar sig till användarnas sociala tillhörighet. Den viktigaste faktorn är att förstå den sociala profilen av användaren genom att använda ett Socialt Proximity Indikator (SPI). Denna indikator på social profil innehåller användarens läge, inställningar, gemensamma vänner, täta samspel mellan vänner, gemenskap medlemskap, och många fler attribut . Detta projekt behandlar utnyttjandet av användarens smartphone som en detektor av användarens sociala sammanhang för att ansluta sig till social närhet mellan några två (eller fler) personer. Målet med denna indikator är att identifiera den (nuvarande) uppsättning av smartphone-användare som skulle vilja delta i någon applikation specifik aktivitet, till exempel mediedelning, gruppsamtal, etc.

SPI identifiera användare baserat på deras sociala interaktioner och gemensamma preferenser. En fotodelnings applikation har föreslagits och byggd för Android smartphone -plattformen, baserad på data från SPI. Denna applikation använder flera fysiska och virtuella sensorer (genom användarens smartphone) för att bestämma ramen för användaren. Efter en analys kommer programmet att initiera fotodelning mellan den identifierade gruppen som hittades i analysen. Applikationen , liksom den underliggande koden ,teoretiska begrepp, och resultaten kommer att presenteras i denna uppsats. Experimenten som genomfördes under detta examensarbete tyder på att det är möjligt att bygga smartphone baserad på SPI där närhet innehåller mycket mer än bara fysisk plats. Det gör applikationen unik än de befintliga applikationer.

Nyckelord: Social Naauml;rhet, Smartphone, sociala interaktioner, bild delning, virtuella sensorer

Acknowledgments

My sincere thanks are due to my thesis academic supervisor and examiner Prof. Dr. Gerald Q. ”Chip” Maguire Jr. (School of Information and Communication Technologies, KTH Royal Institute of Technology, Stockholm, Sweden) for his valuable suggestions and recommendations. I am very grateful for his timely feedbacks. My heartfelt gratitude to my thesis supervisor Mr. Christopher Howson (Researcher at Technicolor R&D, Rennes, France) and his team for sharing their valuable ideas and continuous feedback.
Finally, I thank all for giving me opportunity to express myself and supporting me throughout this endeavor.

-Dipesh Mitthalal

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

API	Application Programming Interface
C2S	Click 2 Share
CRUD	Create, Read, Update and Delete
GPS	Global Positioning System
GSM	Global System for Mobile
ID	Identifier
IR	Infrared
JAX-RS	Java API for RESTful Web Services
JSON	JavaScript Object Notation
OS	Operating System
OSN	Online Social Network
RADAR	Radio Detection And Ranging
RDBMS	Relational DataBase Management System
REST	REpresentational State Transfer
RF	Radio Frequency
RFID	Radio Frequency IDentification
SMS	Short Message Service
SPI	Social Proximity Indicator
SQL	Structured Query Language

TDOA	Time Difference of Arrival
UWB	UltraWideband
VoIP	Voice over Internet Protocol

Chapter 1

Introduction

The penetration of smartphones into daily lives [1] has motivated researchers to make smart and powerful uses of this device. Smartphones not only provide various aspects of information about the users, but also are offering increasingly complex services. Data from the smartphone helps in inferring the social group of the user by keeping track of calls, short messages, e-mails, locations, and other activities carried out by an user [2, 3]. Osmani et al. points out that many people habitually carry their mobile phone and this makes it an ideal tool to learn about human interactions and user behavior [4]. The smartphone's hardware and software capabilities help its user remain connected with persons not only through voice calls, but also through e-mails and messages. Users form social groups through various modes of interactions such as physical meetings, remote meetings (online messaging services), voice calls, etc.

A social group consists of people whom you are connected to, on the basis of some relationship. This relationship can be as friends, spouses, siblings, colleagues, classmates, etc. A social group can be constructed in several ways. It can be based on location proximity, it can be based on similar likes and interests, it can be based on behavior, interaction, etc. An Online Social Network (OSN) consists of many social groups and enables us to estimate the social proximity between (any two) users who participate in the same OSN.

In 2009, proximity was the buzz word in the era of social networking when Google launched its location based service named "Latitude" [5]. In 2011, there was hype concerning social proximity and several start-ups were building proximity based applications [6, 7, 8, 9]. These applications used location as a proximity measurement to provide social networking services, but each of these applications ultimately failed. A major reason is that people are not interested in sharing their information with strangers. Users found explicit check-in services in Facebook and latitude as intrusive. Google latitude required users to explicitly check-in and share their location with the rest of the world. Google Latitude's explicit check-in was retired in 2013. Color Labs' location proximity based photo sharing application called "Color" raised US$41 Million in initial funding, but the application was withdrawn in 2012. This application did not use any existing social networks (such as Facebook, Twitter, etc.) nor did it use any other data from the user's smartphone to find the user's social relationships. When a user clicked a picture it was shared with Color's application server and the picture was analyzed based on the location where it was taken and objects present in the picture, and shared with people in your vicinity. Wechat [10], a chat application with a media sharing feature, was introduced at the same time as Color, and there were 10 times more Wechat users than Color users. This shows that people often interact (messaging) with each other, to build the social affinity before hanging out together. Nokia built a proximity based social network [11] for cruise travelers by creating an *ad hoc* network of smartphones on a cruise ship and provided various services to people on board. This solution was based only on proximity of physical location.

Today location proximity is not the only factor to consider while building social proximity based applications. People interact with each other and build relationships not only by meeting at a physical point, but also by exchanging thoughts and sharing preferences through messaging applications, voice calls, or online social networks. The increasing intersection between social computing and smartphones provides many new ways for people to interact with one another. Mobile access allow users to share their thoughts and images in near real time, as well as keeping up to date about people in their social network. This way of keeping in touch with friends remotely is referred to as social proximity. There has been an increase in the number of applications which share location and other contextual information automatically [12, 13, 14]. New phones can be programmed to share a user's real-time activity with friends via social networks or activity with office colleagues through e-mail clients.

Social networks based on the location proximity of the user have always been an interesting topic for smartphone application developers. Researchers in the mobile commerce lab of Carnegie Mellon University developed a web application called Locaccino [15]. This application enables users to share their location based on user defined preferences. A similar social networking application is addressed in this thesis, but it considers social proximity *together* with location proximity. The ability to determine the social proximity between any two persons is a key enabler for many intelligent services that are expected to be seen in the future. A person X is said to be in social proximity of person Y, when the X is connected to

Y remotely through any message, or e-mail, or social networking media. Services such as media sharing, recommendation services, and advertisement services are expected to be influenced by the user's social proximity. Social proximity not only refers to the proximity of friends in a social network, but it also refers to any person with whom a user interacts, independently of whether this interaction is through instant messaging, voice calls, or e-mail services.

Imagine a photo sharing application running on your smartphone which automatically shares the picture you have taken at an event with the people in your vicinity and your acquaintances. In addition, you expect to receive images from others in your vicinity who are present at the same event. The possibilities of such an application were demonstrated in the application called color [8] which automatically shares pictures with other users of the application who are present at the same location. Therefore, a user of this application will not be surprised to receive random photo from strangers in their vicinity. Recent examples of mobile social applications include Loopt [6] which displays your friend's location on a map and dodgeball [7] which continuously senses your location and finds friends of friends within 10 block radius. These location aware applications are useful to provide different kinds of services exploiting social networking, but currently are only based on the participant's location.

1.1 Problem Statement

The fact that the earlier set of applications did not become popular with smartphone users hints that, neither physical proximity or OSN based social proximity alone is sufficient for proximity based services. The fact remains that there may be two users within close social proximity but who have different preferences. Social proximity fused with location proximity and individual preferences is necessary for user centric system. This thesis project aims to build a user centric multimodal smartphone based social proximity indicator. Also a prototype of photo sharing application based on social proximity combined with knowledge of co-located users in close physical proximity is built. We leverage data from virtual sensors to gather social proximity related information. A virtual sensor can be used to mask the explicit data sources (sensors) that provide data [16] and hence make the sensor's functionality more developer friendly. Consider the case of several people within close social proximity who would like to go fishing. One or two persons in this group are not able to go, but they would still like to receive images of their friends fishing. In this scenario location proximity based photo sharing will fail, hence we propose to fuse social proximity with location proximity along with each user's preferences. So that everyone within a defined social proximity will receive the shared images. All the required data for our social proximity indicator is captured via the user's smartphone.

To summarize, the following categories of proximity based solutions facilitate social networking:

- Application leverages geo-location of the users to provide some service(s).
- Application relies on OSN to understand a user's social graph and offer services based on social graph.
- Applications that are based on geo-location as well as OSNs and using data from both the sources to provide services to the end users.

The three types of applications described above impose restrictions on users, such as:

- Users should expect to meet strangers when the service is only location based.
- Users are assumed to be using one or more of the popular OSNs.

1.2 Motivation

The motivations for this thesis project are:

- To create a social proximity indicator which is application and ecosystem independent (i.e. not tied into Google, Apple, Facebook, etc.),
- To leverage ubiquitous smartphones as the source of user data to build the indicator,
- To offer personalized enhanced services to an end user, and
- To provide useful services to non OSN users.

1.3 Objectives

The objectives of this thesis project are:

- To build a social proximity indicator which relies on both popular OSNs (specifically Twitter and Facebook) and location proximity (specifically Foursquare and Google Latitude), but also takes into account non-OSN users who interact with people through voice calls, text messages, and who frequently meet with each other by creating a event in their personal calendar or because they collaborate through any other media processed via their smartphones in order to participate in a group activity.

- Propose a personalized model for logging and mining continuous social interactions of every individual user via their smartphone.
- Propose a novel approach to build applications that consider individual's preferences.

1.4 Contributions

This thesis project makes the following contributions:

- A comprehensive review of the existing research and an analysis of the reason for unpopularity of existing application in order to understand proximity based services and their expectations from a user's point of view.
- Construction of social graphs through social interactions.
- The implementation of a photo sharing application as a proof of concept to demonstrate the feasibility of social proximity based user centric (personalized) applications.
- To demonstrate that SPI can be used in commercial systems which cater to personalized needs. For example, movie recommender systems, media sharing applications, etc.
- Give a new dimension to social proximity outside OSNs.

1.5 Limitations

This thesis project had the following limitations:

- Image processing is outside the scope of this project, hence this project relies on the meta data explicitly made available via images as a source of information in this project.
- It is difficult to mine e-mails in the user's phone as most proprietary e-mail clients do not grant direct access to these messages. The result is that we can only mine these messages by introducing an e-mail project service - either in the smartphone or in a server in the fixed network.

1.6 Outline

The rest of this thesis is organized as follows: Chapter 2 discusses the related work regarding various dimensions of the proximity based solutions. Chapter 3 gives an overview of research methodology followed in conducting this research and introduces the proposed photo sharing application named Click 2 Share (C2S). Chapter 4 presents an overview of smartphones and justifies the reason for choosing Google's Android OS for the development of a prototype. Chapter 5 discusses the various possible indicators of social proximity. Football based contacts rating algorithm is proposed in chapter 6. Implementation details are given in chapter 7 followed by evaluation in chapter 8 and conclusion in chapter 9. Appendix A presents several use case scenarios which can take advantage of the proposed image sharing application. Appendix B gives the user agreement and survey questionnaire used for evaluating C2S. Appendix C gives the analysis of results obtained in survey.

Chapter 2

Background

This chapter gives an overview of previous work done on various aspects contributing to proximity based applications. Proximity based social networking has been the subject of extensive research. To understand human networks, a sociometer [17] was developed by Choudhury and Pentland, researchers at the Human Dynamics Laboratory in Massachusetts Institute of Technology (MIT). This device had an IR transceiver to detect humans nearby, a microphone to detect speech information, and accelerometer to detect the user's motion. Using this setup, researchers showed that human networking and social proximity can be modeled based on inputs from these sensors. Today's smartphones have all these sensors inbuilt and many more features can be achieved by integration of additional social networking services. However, in order to determine proximity researchers have explored various approaches. This chapter describes some of these different approaches to determine location, co-location, social proximity, activity, and the question of whom would I like to know.

2.1 Location

Location refers to determining a user's physical location. This gives an answer to the question "Where am I?". A comprehensive review of localization techniques is given by LaMarca and de Lara in [18]. This section presents an overview of a number of technologies and research projects concerned with inferring location. Some of these techniques require installation of an infrastructure that can used in conjunction with (or in some cases exclusively for) location. Infrastructure-based technologies can be classified into different classes based upon their core technology: Wi-Fi [19, 20, 21], GSM [22, 23, 24], GPS, ultrasound with RF [25, 26], RFID [27], UWB [28], Bluetooth [29], and TDOA based schemes [30]. These technologies leverage support from an infrastructure to locate an object in a physical space. Each of these technologies has its limitations. For example, RADAR [19], an RF based system, uses triangulation to locate a user by using signal strength information gathered at multiple RF receivers in known locations. Wi-Fi based localization exploits Wi-Fi receiver's antennas present in a physical space. It would seem that additional antennas should yield greater accuracy, but this is not the case in practice as changes in the physical environment can lead to large changes in the RF propagation, hence reducing the accuracy of the technique. Augmenting the receiver with a mechanism for time stamping the arrival of Wi-LAN radio frames can be used to increase the location accuracy, see for example [31] and [32]. In the past, several start-ups (Dodgeball [7], Loopt [6], Pantopic [33]) introduced localization based social applications, relying on infrastructure based localization support in mobile phones via Wi-Fi, Bluetooth, and GPS. These applications often offered only limited functionality and were incapable of providing a full range of social functions. Most of the solutions that have been introduced were one dimensional, in that they only supported a single function, such as micro-blogging, location based messages, location based media-sharing, etc. Due to the limitations of the underlying infrastructure the solutions were often not well received by a rapidly changing technology community. For example, RFID based co-location requires that each user carry an RFID tag all the time and requires that RFID readers be present at all locations [34]. Researchers also proposed the use of fingerprinting based wireless positioning using Wi-Fi [35, 36, 37]. Present day smartphones fuse data from more than one inbuilt sensors to improve reliability and accuracy of the location system (for example GPS, gyroscope, and compass) in [38]. In this thesis project, the user's location will be identified based on sensor fusion to increase accuracy and reliability.

2.2 Colocation

Colocation refers to the presence of one or many other persons together with a user. This co-location may lead to one or more social activities. Co-location answers the question "Who are you with physically?". Peer-based localization techniques helps to infer co-location with respect to other devices. This technique either forms a virtual map between one or more devices and infers the actual distances between the devices or infers the distances between pairs of devices in proximity. Schemes such as Humming-bird [39] and NearMe [40] are proximity based schemes which detect devices within 30 to 100 meters of each other. Beep-Beep [41] uses sound to detect users in close proximity, but does not scale when more than two devices are present. Bluetooth discovery is used by Bluehoo [42] to detect friends within range via their Bluetooth device, but a Bluetooth proximity network has lot of limitations when sensing social interactions. First, Bluetooth might fail to detect all nearby devices hence the data produced by Bluetooth is unreliable (with a lot of false negatives). Second, proximity does not always mean social interaction (for example, a person might be traveling via public transport and their smartphone might discover many other active devices of people who are traveling in same vehicle). Third, a person might carry more than one Bluetooth enabled device, hence one cannot say that a Bluetooth device corresponds to one individual [43]. A mobile platform named comm2sense [44] exploits Wi-Fi-Hotspot and Wi-Fi receivers to estimate the distance between smartphones with a resolution of less than 1m, but suffers from the limitation that the Wi-Fi capability of individual phones might vary and it also requires modification of firmware to allow adjustment of transmitting power of the Wi-Fi transmitter. The camera on the smartphone can also be exploited to detect if the person is in front of you and the microphone of the smartphone can be used to detect the actual interaction between a pair or group of individuals. All of these peer based localization methods suffer from computational limitations. They rely on an external server to compute using the raw inputs from the smartphone to draw a conclusion on actual social interaction. Today's smartphones are computationally more powerful than the devices used in earlier efforts, hence their computational power can be exploited to remove the reliance on a central node. Another aspect to be considered here is social proximity, that is proximity between individuals who may be located remotely, but they might be close to each other in a social sense. For example, if a group of persons are sitting together, then the localization technology described above will help in locating them, but physical proximity does not necessarily mean any of these people are interacting with another person at this location [45]. Inevitably, users exploit the multitasking capabilities of smartphones and they try to connect to both persons located remotely and to those who are present physically. As a result, co-location alone is not sufficient for predicting social proximity between two individuals [46].

2.3 Social Proximity

Social Proximity refers to the social distance among individuals in a group. It answers the question "Who is in your social network of friends or work colleagues?". Facebook friendship can indicate social proximity between pairs of individuals. Wiese et al. point out that association between factors such as collocation frequency, communication frequency, closeness, and social group, with preferences for sharing specific kinds of information is important to understand [47]. Manually organizing one's friends into the groups is considered to be a great burden [48]. There have been proposals of inferring the social context of person automatically using his social profile in Online Social Networks

(OSNs) [49]. Social Net [50] uses short-range RF communications to track the frequency and duration of user encounters to measure social distance. Users can be asked to individually group their friends based on hobbies, work place, and place where they first became friends to measure the social proximity of each of these individuals. Researchers have also examined data related to human interaction based on e-mail [51, 52] and call logs [53]. Studies have also been conducted to compare the self-reported behavior and the logs of cell phone data to infer the social network structure [54]. Inevitably, all of the studies rely on stored data and gives results based on different computations on these stored data. This suffers from not being able to detect in new friendships in real-time. What does a friendship “looks like” in a social graph? How can these relationships be represented?. One simple way of measuring closeness/friendship is directly asking the user “How close do you feel to this person?” on a 1-5 Likert scale [47], similar to work by McCarty[55]. OSNs are used to build recommendation systems for personalized Televison content [56]. Researchers claim that use of OSN helps improve internal communication of an organization[57]. Merging information from an OSN with the user’s phone book was attempted to allow users to have continuous updates about their online friends [58].

2.4 Activity

The study of human behavior has been an interesting area of research for decades. One of the goals has been to automatically derive contexts from these activities. This has lead to research on context aware computing. The answer to the question “What are you doing?” is a key element to build services that can support the user to answer the question “What you would like to do?”. Sensor fusion helps when inferring activities of the user (walking, sitting, driving) and an application programming interface (API) can be used to retrieve context information. Freemotion [59] is an example of such an API. Continuous sensing [60] could help in delivering better services as the system can be more responsive to changes in the user’s activities, interests, and social context. However, we must take into consideration the fact that affinity and activity are different. Can shared activity imply shared affinity? Affinity is concerned with what a user likes, for example, watching TV, drinking coffee, singing, etc. While an activity is what a user does at some point of time, for example, working on a project, going on date, kayaking, etc. Communication is a strong indicator of willingness to share information, especially when colocation is less frequent. So a quantitative analysis of a user’s communication pattern (how frequently a user communicates and with whom they communicate) of different users can help derive an approximation of their shared affinities.

Wiese, et al. stated "[...]there are people that are important to us who we communicate with much more often than we see (e.g. family who do not live nearby); similarly, that there are people who are less important to us that we see often but do not exchange as much communication with (coworkers who you see often, but with whom you otherwise do not communicate). This interaction was also significant, revealing that communication is a stronger indicator of willingness to share when collocation is less frequent." - [47]. They further explain the possible common scenarios to infer a shared activity among a group of individuals. Also, sharing activity may depend on in-common information. We are more likely to share our activity to neighbors/acquaintances if we have common friends. In-common information refers to the intersection of data between two data sets. In this example, the data set is the user's social network and in common information hints at common friends.

Another possibility to determine shared affinities is to make use of existing social networks to pull information from by mining historical data from these networks and combining it with data available in the user's smartphone. The social profile of a person can be built based upon existing data from the smartphone and user provided data, such as what he likes to share with others within his social network [61]. Another possibility is to build a repository of logs and to use an ontology together with machine learning to infer the shared affinities of the user. Data from the user's smartphone calendar events can be exploited to determine shared activities of the user. Interpersonal relationship has been identified by several researcher as an important factor when sharing preferences among people [62, 63, 13]. According to Tsai et al. getting feedback based on the user's history makes the user feel more comfortable, thus the user will be more willing to share his or her information [13]. Mood and the user's current activity can also have an impact on what preferences a user shares with anyone, as studied in [63] for sharing location context.

2.5 Whom would I like to know?

Users who meet often or meet infrequently over long periods are anticipated to be potential friends or candidates for introduction to each other. In social-net [50], if a person is a mutual friend of two friends who frequently meet, then the mutual friend will receive a message suggesting he/she introduce these individuals to one another. This process leverages human's social skills to decide whether two people should be introduced.

2.6 Summary

This chapter reviewed some of the earlier work in related area. The chapter started with an example of modeling human network and social proximity based based on inputs from various sources. Then various topics which contributes to such system are also reviewed. In summary, this chapter covered the various aspects of proximity based systems and its related work, which helps to understand proximity based systems.

Chapter 3

Methodology

This chapter introduces the scientific procedures used in addressing the problem statement. *Ex post facto research* was attempted to discover the reason for the unpopularity of applications [8, 12] in past. Blogs [64, 65] were read to understand that people interact often through traditional voice calls and messages rather than location based social proximity networks [12, 8, 6, 7]. Conceptual research methods were also adopted in this thesis project. Because of social interaction of humans through their smartphones we can build a social interaction graph for every user. Using a quantitative method based upon asking each user to classify their usage patterns or defining their own social network does not work well to produce a user centered system, as the usage pattern varies for each individual and hence, there is not any uniform scale for all individuals. The user's smartphone usage helps in building a data set which is subsequently used to infer the users' social proximity network. Users' interactions are observed and characterized by an application running in the user's smartphone.

A contact rating algorithm is proposed and evaluated using reality mining dataset [3]. Conclusions are made after soliciting feedback from the user about the rating of their contacts in their social proximity using the prototype version of the application. Users were asked to verify whether they would intentionally send these images to the other users. This lead to experimental research. The prototype photo sharing application is built on top of SPI to demonstrate the usage of SPI and the prototype is evaluated based on the different social graphs that were built, depending on the meta data about the picture.

To conduct an experiment to identify persons in close social proximity and to share images with them, the project proposed the Click2Share (C2S) system. This system is primarily a photo sharing application based on social proximity. C2S consists of a client and a server (referred as C2S client and C2S server).

An Android based smartphone client frequently interacts with the C2S server by sharing contextual information needed for the SPI to generate a graph based on preferences and the social proximity of the respective user. In this prototype SPI is the main focus i.e.,(core component) of the system.

Figure 3.1 gives an overview of the use case considered in this system. Consider a scenario where a user wants to share the images of his recent fishing activity with people within his close social proximity. First, based on social interactions, a graph of users (set of actors as shown in Figure 3.1) is built and then based on the context of each image, identified users (actors) are further filtered to identify the set of users (actors) who might be interested in receiving the images of the fishing activity of the user. Also a user can be assured that he will receive images only from known persons. Detailed use cases are given in Appendix A. Different indicators are considered for social proximity and a user's affinity is explained in the Chapter 5. The C2S system involves a number of functions, each of these will be described in the section below.

3.1 User Registration

A user registers with the service by providing their phone number and their e-mail address. The goal is to uniquely identify and characterize each user. By registering, the user agrees to allow the C2S system to access his/her smartphone data in return of a personalized service.

3.2 User Profile

The user's profile can be built by manually soliciting information from the user or automatically extracting from data available to the Android OS. This information is used to identify the other users who might be interested (i.e., to determine which images are relevant) in order to select which images should be shared. Linking OSN accounts and synchronization of contacts book can also help to improve the quality of the SPI.

3.3 Photo Sharing

User can decide to share a specific picture from their gallery by uploading this picture to the C2S server via their C2S client. The C2S server chooses the recipients of the image based on the user's social interaction graph and the image's meta data. The image's meta data includes the date, location, and a description.

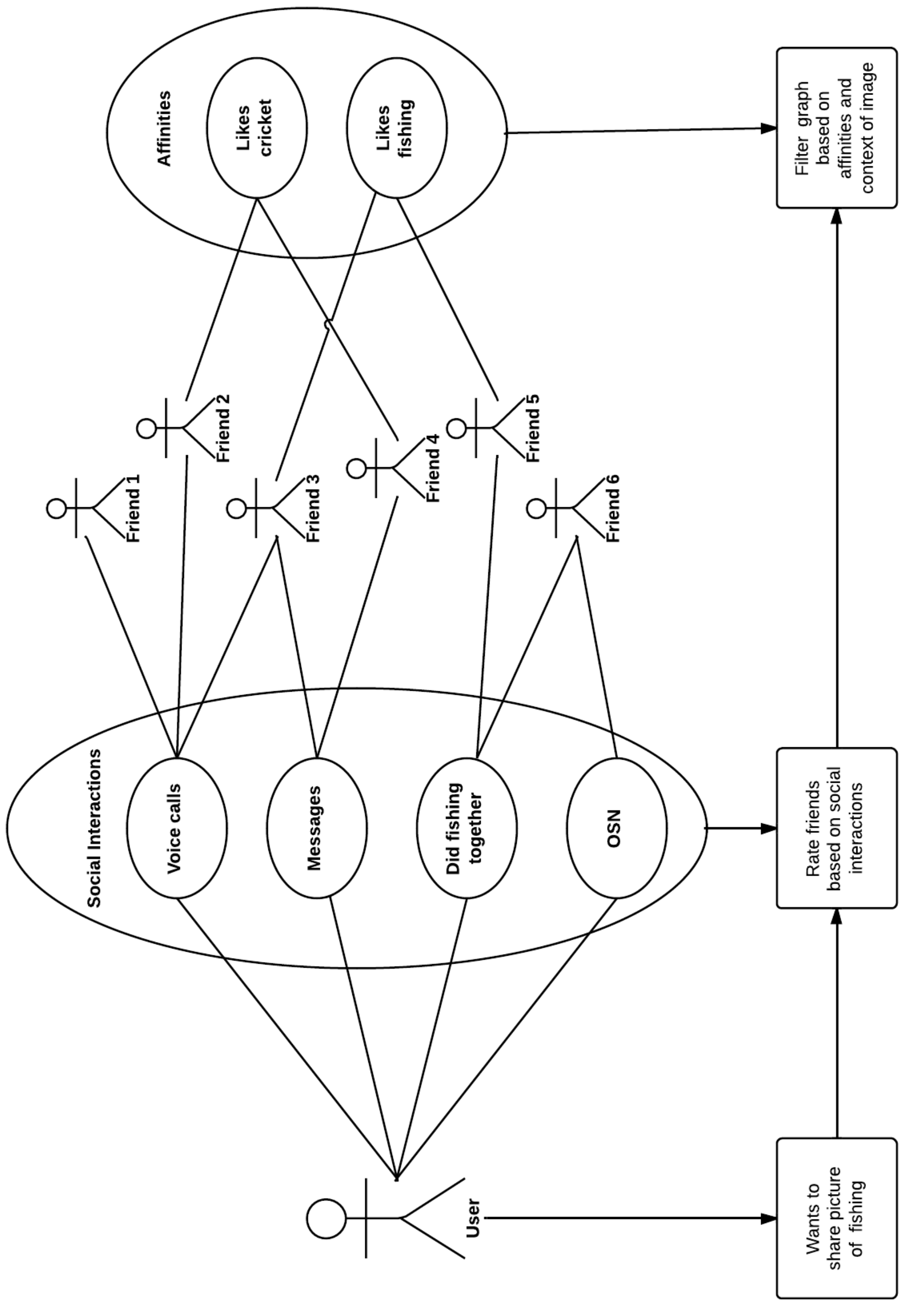

Figure 3.1: Use case diagram for image sharing application

The C2S client can also be configured so that, every picture a user takes is automatically uploaded to C2S server and it is the responsibility of C2S system to decide whom should be recipients of each picture and thus sharing happens automatically. Image processing is outside the scope of the project, but image processing could potentially give additional context information. Such context information can be help the photo sharing application to better determine which images should be sent to which other users. Example: identifying the persons, objects in the image, and the place where the image was taken would be a boon for applications such as C2S.

3.4 Social Proximity

SPI calculation is the core of the C2S system. Social proximity between any two persons depends on how frequently they interact together with the additional factors presented below (in decreasing order of importance). The prototype takes into account the following factors when estimating the SPI between any two users:

1. Number of voice calls,
2. Number of text messages,
3. Frequency of co-location, are they present together at any events, and
4. Number of common preferences.

Additional details about the SPI computation are presented in the next chapter, where these indicators are characterized after building a user profile.

Chapter 4

Review of key platform elements

This chapter presents a review of smartphones and NoSQL databases. Both of these have been used in development of a prototype. It begins with a description of the Android Operating System (OS) and smartphones that run this OS. This is followed by a section that describes the sensors that smartphones have and how one can access these sensors. The section containing sensors description also explains how one can add new physical or virtual sensors to a smartphone running Android.

4.1 Smartphones

A smartphone is a mobile phone that is able to perform many of the functions of a general purpose computer. It allows users to do more than simply make & receive phone calls and send text messages. A smartphone enables the user to access the internet via a web browser and also provides support for applications such as a calculator, music player, file browser, e-mail client, etc. As with other general purpose computers, a smartphone has an operating system running on it. In 2013, Android smartphones comprised 79% of the global smartphone market [66].

4.1.1 Android

Android is an OS initially developed by Android Inc. This company was bought by Google Inc. in 2005. Android is based on a Linux kernel and has been designed to run on small electronic devices, such as cellphones, Internet tablets, net books, car dashboards, etc. Android is released under an open source Apache license, which has lead to its widespread popularity. Nearly 71% of mobile developers are developing programs for the Android platform [67]. This paves the way for researchers and scientists to exploit the current and future potentials of a wide variety of different types of devices that are used by very large numbers of people. The Android platform is ideal for creating innovative applications through the use of sensors, since the Android OS provides direct access to a smartphone's sensor data. These built-in sensors measure motion, orientation, and various environmental conditions. The next section will introduce some of the sensors that are available in today's smartphones.

4.1.2 Smartphone Sensors

A modern smartphone contains many sensors which are used by applications independently or collaboratively to deliver a better user experience. Sensors can be classified into physical and virtual sensors. Hardware based physical sensors are built into a handset or tablet device. They derive their data by directly measuring specific environmental properties, such as acceleration, geomagnetic field strength, or angular change. Physical sensors include accelerometers, gyroscopes, pressure sensors, proximity sensors, cameras, microphones, GPS receivers, and many other types of physical sensors. Some of these sensors are capable of providing raw data with high precision and accuracy. They can be used to monitor three-dimensional device movement or position. They can also be deployed to detect changes in the ambient environment near a device. They help in activity recognition, for example to know if the user is currently on foot, in a car, on a bicycle, or stationary [68].

Software based sensors are not physical sensors, although they mimic hardware based sensors. Software-based sensors can derive their data from one or more of the hardware-based sensors and are sometimes called virtual sensors or synthetic sensors. Virtual sensors uses one or more physical sensors to arrive at a conclusion. For example to determine location with high accuracy, virtual sensors can utilize the Wi-Fi interface to listen to one or more Wi-Fi access points, output from a GPS receiver, and a wide area mobile network (such as GSM, UMTS, LTE, etc.). Virtual sensors can also be used to combine accelerometer data with location data to improve the accuracy of this location data and to reduce power consumption (for example by avoiding the need to compute a new position when the user is not moving). These sensors play an important role in proximity based applications. Google's fused location provider API is another example of software based sensors which is used in this project [69]. The fused location provider intelligently manages the underlying physical sensors and gives the best location according to the requirements of an application.

4.2 NoSQL Databases

NoSQL databases are used in the implementatio of the prototype. NoSQL encompasses a wide variety of different database technologies and was developed to cater to the needs of storing and access the growing amount of data about users, objects, and products. It was developed to address the performance and processing needs that arise when data is being accessed frequently (but rarely - if ever - modified). NoSQL databases are primarily categorized into Document Databases, Graph stores, key-value stores and Wide-Column stores. The following subsections discuss two popular NoSQL databases: MongoDB (a document database) and Neo4J (a graph database).

4.2.1 MongoDB

MongoDB [70] is an open-source NoSQL document database. MongoDB represents data as collections of JSON documents. MongoDB does not support joins, but does support multi-dimensional data types, such as arrays and embedded documents with different data types. MongoDB community edition is used in this project. Unlike relational databases, MongoDB's support for embedded documents is advantageous when storing a C2S user's profile as this data ranges from textual to spatial data.

4.2.2 Neo4J

Neo4J [71] is a graph database management system. A graph database management system (henceforth, a graph database) is an online database management system with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model. Compared to relational databases and NoSQL stores, a graph database offers increased performance when dealing with connected data. Relational databases lack relationships other than table joins. In relational databases, performance of a join-intensive query deteriorates as datasets increase in size. With a graph database, performance tends to remain relatively constant because queries are localized to the portion of the graph.

Both the above NoSQL databases are scalable and can be easily deployed in a cloud making them ideal candidates for an user centric system such as C2S.

4.3 Summary

This chapter reviewed some state of the art smartphone sensors and NoSQL databases. The review started with the fundamental knowledge of smartphones, a specific OS and sensors that are commonly included in today's smartphones. A comparison of traditional relational database management systems and NoSQL database systems was presented. In summary, this chapter covered the fundamental background of those technologies that have been used to realize a proof of concept prototype of a C2S application.

Chapter 5

Indicators of Social Proximity

This chapter discusses the design of our calculation of SPI. The user profile is the source of information which helps to infer the SPI between any two (or more) smartphone users. The SPI value indicates the proximity (closeness) of one individual to other. In order to provide an interesting and successful service through a mobile application to the user, measures of closeness are insufficient, as user preferences play a significant role in user's selection and usage of smartphone applications. It is important to identify what each user likes. In this application, it will be important to know what kind of images each user would like to receive. The computation of SPI is composed of two building blocks (shown in Figure 5.1): social interaction indicators and affinity indicators. Affinity indicators help to identify user interests, while social interaction indicators help to identify people with whom a user has close social proximity. The combination of both will enable C2S to provide a personalized social proximity based photo sharing application. First, a user profile is built leading to the development of the SPI (which is the major contribution of this thesis project).

5.1 User Profile

Identifying the user and his or her preferences is important for any user centric system. The information in the user profile can be classified into general details, social interaction indicators, and affinity indicators. Each of these classes of information are described in the following subsections.

5.1.1 General Details

The user's phone number is used to uniquely identifying each user. Some mobile service providers embed the phone numbers in their sim card and we extract it

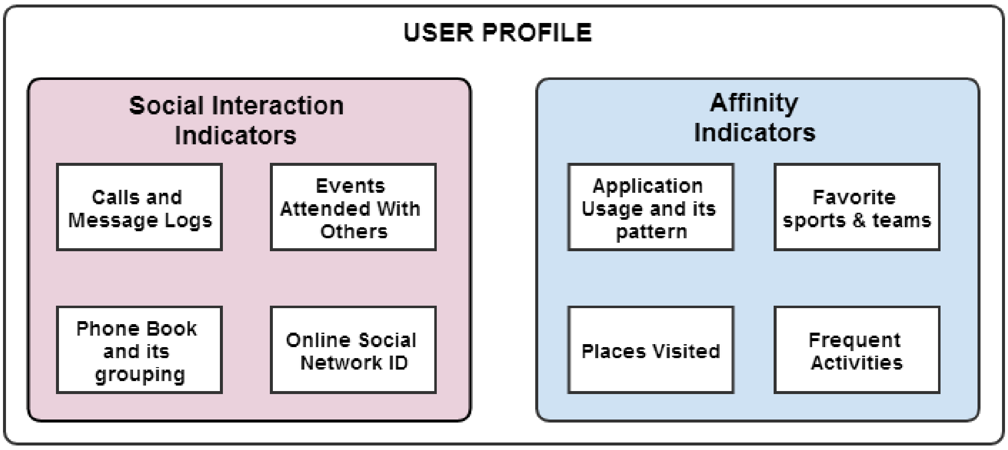

Figure 5.1: Overview of C2S Application

from there. Otherwise, the user is asked to enter their phone number during registration for the service. The user's e-mail address can be retrieved from the Android account to which his/her smartphone is linked. This helps to deal with the case of a user owning more than one Android device, as we assume that the user keeps all of these devices synchronized (the user's Android account is used for this synchronization). Age will help characterize users' affinity and may prove to be advantageous for various applications, such as recommender systems and photo sharing [72]. The user's phone number not only identifies the user, but can be used to infer social proximity if this phone number is stored in another user's phone book. All of these general details are summarized in Table 5.1.

Table 5.1: User Profile-General Details

Identifier	**Source of Input**	**Description**
E-mail ID	Android OS/ User Input	
Phone Number	Android OS/ User Input	Unique ID
Nick Name	User Input	Helpful when name mismatch occurs in user's phone book
First Name	User Input	
Last Name	User Input	

5.1.2 Social Interaction Indicators

Possible social interaction indicators are described in the following paragraphs.

5.1.2.1 Call Logs

The user's call logs gives a fair estimate of people with whom a user is in close social proximity. The duration of calls and time of the day when each call is made indicates the closeness between two users (as calls outside of the user's normal working hours are less likely to be made by work colleagues). Also characterization of the detailed call logs gives greater insight into the user's social network [73].

5.1.2.2 Message Logs

People often interact with one another remotely using messaging applications before they decide to spend time together at a physical location. So the logs of messaging applications gives a more realistic idea about which other people, a user is in close social proximity [74].

5.1.2.3 Events attended with others

The user's calendar can be used to get the details of events and the event's attendees list. Given a list of attendees, we can compute whether a user has already been with an other person, thus increasing their social proximity [75].

5.1.2.4 Group Information from the user's phone book

Today's digital phone books in a smartphone enable users to group entries. This grouping information indicates the level of social proximity with others [58].

5.1.2.5 OSN ID

The user's OSN's ID can be used to capture details of the user's online friends. The OSN profile of the user also gives some of the user's affinities [76].

5.1.2.6 Co-relation of places visited

If two users who often communicate with each other via text messages or voice calls meet and spend time together, this is strong indication of social proximity between these two users. Spending time together can also be characterized by being together at a physical location [77].

5.1.2.7 Phone Book

The user's phone book provides an overview of contact details of people with whom this user may wish to communicate. A smartphone's phone book gives a variety of information about a user's contacts (such as name, phone number, e-mail address, etc.). This information helps in building the user's profile when computing potential relationships over the set of C2S users. For example, the e-mail address of a contact can be linked with calendar events, which usually contains one or more e-mail addresses and a location. With the help of the user's phone book, other non C2S users can be identified and images can be shared with them [78]. This enables C2S to deliver services even to non C2S users if their e-mail address is known or they can be provided services via text messages if their Twitter or other messaging service address is known.

5.1.2.8 Meta Data of the picture

GPS details (location) and titles of images (names, subject of photo, symbolic place name, etc.) are important parts of information which will be used to perform the photo sharing. Titles of image might hint about the context of the image. This information can be validated against the user's contact's preferences.

5.1.3 Affinity Indicators

Affinity indicators refer to what a user likes. This can also offer hints about a user's special interest in specific activities. Activity recognition agents can be linked to the indicators (other researchers at Technicolor R&D are working on ontology based activity recognition). While a user might like certain activity, they might prefer not to do it. Activities can range from using a smartphone application frequently to playing some sport on a regular basis. A user might like watching cricket, but may not necessarily be interested in playing it. The subtopics described in the following paragraphs give an overview of some of the various possible sources which can indicate a user's affinity, along with methods to capture this information.

5.1.3.1 Application Usage

Sets of specific installed applications and application usage patterns can indicate the affinity of a person with respect to the domain of the respective application(s). For example, if a person frequently uses a cricket score application to check cricket match scores, then this indicates that person has a positive affinity towards cricket. The same reasoning is applicable to any other type of application which is specific to some activity or interest.

5.1.3.2 Activities

A user's calendar gives hints about the users favorite activities, as these are activities that the user schedules time for. For example, if a user's calendar show that every week, the user goes fishing, then this shows the affinity of the user towards fishing. This source of information helps to accurately infer the user's preferences. Also calendar information can give hints about like minded users by looking at the attendees field (i.e., other users who attended the same event). Location information from the user's calendar adds another dimension to the user's affinity. If a group of users go fishing at a specific lake, this information helps in profiling the user's major affinities.

5.1.3.3 Places Visited

A user's frequent visits to a given place gives additional details about this user's preferred locations. These locations might suggest that a user has an affinity for the pictures from these location or events taking place at these locations, suggesting that a user might be interested in pictures that others have taken of these locations or same events [79]. For example watching cricket match at a specific place on a specific day might indicate that user has an affinity towards cricket and likes a specific team.

5.2 User Profile Characterization

This section explains the characterization of those indicators presented in the previous section. Characterization explains what data is collected and presents different dimensions of the collected data that may hint at social proximity.

5.2.1 Social Interactions

The following paragraphs discuss possible modes of building a user profile based on the user's interactions with others via a smartphone.

5.2.1.1 Call Logs

Each entry for each phone number appearing in a call log (as shown in Table 5.2) will be processed and tie strength will be calculated and updated in the C2S server. Patterns of which user who initiates the call indicates whether the relationship is symmetrical or whether the closeness differs from this individual's point of view. Categorization of the duration of calls is based on time of day and day of the week, as these indicate the closeness with regard to social proximity. Inevitably, the duration of calls within the same proximity levels differ from user to user and hence the number of calls per day is another factor to be considered. For example, the fact that a user makes voice calls to communicate, even though the call duration might be shorter than calls to others may indicate that these persons are socially close. Hence, our social proximity indicator has to adapt to such users and scenarios by taking into consideration the logs of whole days and the number of calls made between all tuples of user.

Table 5.2: Characterization of Call Logs

Name of contact (if available)	Alice	Bob
Phone number	XXXXXXXX	XXXXXXXX
Call received /Made	R	M
Group name	friend	family
Average Duration(minutes)	5	2
Date	YYYY-MM-DD	YYYY-MM-DD
Time of day (Morning, Noon, Evening, Night) (07-12, 13-16, 17-21, 21-06Hrs)	Noon	Night
Weekday or weekend	weekday	weekend
Number of calls	2	15

5.2.1.2 Message Logs

The number of messages exchanged between any two users per day also indicate the social proximity between these two users. Table 5.3 shows a summary extracted from message logs.

Table 5.3: Characterization of Message Logs

Name of contact (if available)	Alice	Bob
Phone number	XXXXXXXX	XXXXXXXX
Message received /Sent	R	S
Group name	friend	family
Number of messages	5	2
Date	YYYY-MM-DD	YYYY-MM-DD
Time of day (Morning, Noon, Evening, Night) (07-12, 13-16, 17-21, 21-06Hrs)	Noon	Night
Weekday or weekend	weekday	weekend

Social proximity is characterized not only by voice call logs. There can be co-relations between text messages and calls, thus indicating close social proximity. There are situations where users prefer to express themselves through text messages, rather than making voice calls, thus indicating increased closeness in social proximity. Hence the number of messages sent and received is one indicator of social proximity.

5.2.1.3 OSN Accounts

One of the important goal of this system is to provide services to non OSN users. A user can link his/her OSN account to this system if he/she chooses to do so. OSNs can be exploited to get additional relevant details to help construct the user's profile. For example, a Facebook profile provides various fields such as favorite sport, favorite team, etc.

5.2.1.4 Metadata of the Image

Identifying objects in images and processing these images is outside the scope of this project. Hence, we rely on the metadata shown in Table 5.4 of each picture in our prototype photo sharing application.

Table 5.4: Context extraction from an Image's metadata

Context	**Description**
GPS co-ordinates	The location where the picture was taken. If GPS details are not available, then location details will be inferred using other sensors.
Date and Time	Date and time when the picture is taken
Persons in picture	To be decided manually
Title of the picture	If any keywords are present, they can hint at the context of the image

5.2.2 Affinity Indicators

Affinity indicators give hints about a user's preferences. We describe and characterize these affinity indicators in the following paragraphs.

5.2.2.1 Calendar Events

The user's smartphone's calendar hints at social interactions and also affinity towards specific types of events. While not every attendee at an event may be close, they all might share the same affinity (as they are all involved in some common activity). Table 5.5 gives some of the information which might be extracted from the smartphone's calendar.

Table 5.5: Characterization of calendar details

Event Name	**Keywords indicating affinity**
Location	Geo coding & Reverse geo-coding
Guests	E-mail address
Description	Keywords
Repetition	The greater the number of repetition, the stronger the affinity
Privacy	Private events are more important than public events as the former will indicate closer social proximity with others attending the private event

This calendar information can help to find affinity regarding specific sports. For example, if the user watches a cricket match in a stadium, this suggests that this user likes cricket.

5.2.2.2 Application Usage

Application usage time is not monitored. If the C2S application were to record time and usage of every application, then the C2S would need to periodically run as a background service, thus draining the battery power. Hence, were proposed to check recently used applications, using foreground applications to create analytics to be used by C2S system. Another challenge here is to identify the domain of applications. For example, a dictionary of popular applications can be built with manually selected domains. This way we can identify that "cricinfo" is related to sports, specifically cricket. The list of recently used apps is a good indicator of a user's favorite applications and it implies that the user's interest lies in that particular domain.

5.2.2.3 Location Proximity

Logging the user's location is advantageous when processing call logs as the location when each call/message occurs and content of message logs provides additional information. For example, two users might be in sufficiently close location proximity such that no need arises for any voice or message communication. Location logs can help infer social proximity through voice calls logs and message logs. A Bluetooth sensor can be used to detect if users are within close physical proximity of each other. This information along with message and call logs will help infer social proximity. Geo–coding is helpful when a user visits specific places of interest. Repeated visits are an indicator of affinity. If similar places are being visited by other users in social proximity, then a recommendation can be made or images can be automatically shared.

Chapter 6

Design of a Social Proximity Index

This chapter discusses the estimation of social proximity between any two (or more) users. We show how the data gathered (as explained in the previous chapter) is used to compute the tie strength between a user and his/her contacts. First, tie strength is calculated using call logs and message logs, then this value is normalized. We discuss how individual tie strength leads to a SPI value. Initially we designed an algorithm which normalized the call logs and message logs and infers the SPI. However, this normalization based SPI did not take regularity of interaction into consideration and hence another design is proposed which infers SPI based on a football ranking algorithm.

6.1 Design 1:Normalization based SPI

The following paragraphs explain the SPI calculation based on voice calls and text messages of any smartphone user.

6.1.1 Proximity calculation based on call logs

From the explanations in the previous chapter about call logs and message logs it is evident that there is no single reference scale by which these logs can be characterized and compared to infer social proximity. Table 6.1 shows the first step of proximity calculation based upon call logs and Figure 6.1 shows the social graph of an user based on the results of proximity calculations. Individual preferences differ and the SPI should adapt to one's preferences and determine social proximity accordingly. It is anticipated that when a user spends more time (characterized by number of messages & voice calls, being together at the same event, being together at a place, etc.) with their socially close friends and acquaintances that the period of time spent together is a strong indicator of their social proximity, as tailored for individual preferences. Table 6.2 displays the average call duration for a user based on time slots.

Table 6.1: Proximity calculation based on call logs

Phone number	**Time when call received or made (HHMM*) every 24Hours**	**Duration (seconds)**	**Average call duration of the respective time slot †**
xxx345	0900	35	136/3 = 45.33
xxx123	1000	45	
xxx234	1700	56	
xxx345	2000	75	150/2 = 87.5
xxx123	2030	100	

Table 6.2: Average call duration of time slots according to call logs in Table 6.1

Time Slot (HHMM)	**Average call duration (seconds)**
0800 - 1800	45.33
1801 - 0759	87.5

Table 6.3 indicates the social proximity between caller and callee based on Equation 6.1.

$$\text{Tie strength} = \frac{\text{call duration} \times \text{ weight of the respective timeslot}}{\text{average call duration of respective timeslot}} \tag{6.1}$$

Table 6.3: Tie strength calculation between the caller and callee per call

Phone number	**Time when call received/Made** (HHMMddmmyy) 24Hours	**Duration (seconds)**	**Tie strength**
xxx345	900	35	(35*0.5)/45.33 = 0.38
xxx123	1000	45	(45*0.5)/45.33 = 0.49
xxx234	1700	56	(56*0.5)/45.33 = 0.61
xxx345	2000	75	(75*0.75)/87.5 = 0.64
xxx123	2030	100	(100*0.75)/87.5 = 0.85

*Time is represented in HHMM format where HH stands for hours and MM stands for minutes

†We have assumed that 0800–1800 represent Office hours to which we give a weight of 0.5 per average call duration; while 1801–0759 represents personal hours to which we give a weight of 0.75 per average call duration.

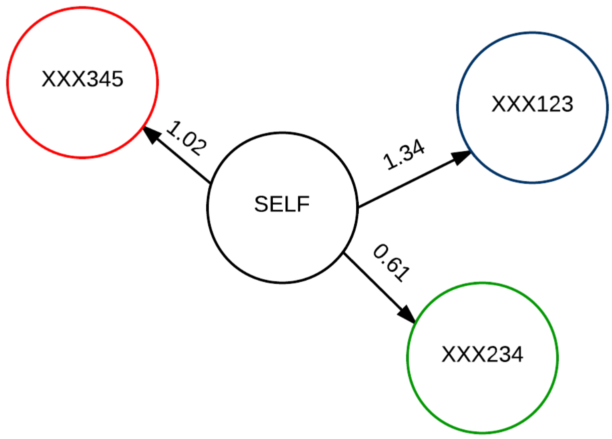

Figure 6.1: Social graph of the user showing tie strength

6.1.2 Proximity calculation based on messaging logs

Every user has his/her own traits of sending messages to individuals in their social circle. Thus I may send 20 messages per day to close friends, but only 10 messages per day to acquaintances. Moreover this pattern of texting differs from individual to individual. Hence a relative measurement scale is proposed to be calculated using Equation 6.2 shown in Table 6.4.

$$\text{Tie strength} = \frac{\text{Total number of messages to one individual}}{\text{Average of the number of total messages sent to each contact}} \tag{6.2}$$

Table 6.4: Example of messaging logs

Phone number	Number of messages in 24 hours	tie strength
xxx345	10	(10)/6.33 = 1.57
xxx123	4	(4)/6.33 = 0.63
xxx234	5	(5)/6.33 = 0.78

6.1.3 Normalization of calls and message logs

It is challenging and difficult to normalize call and message logs as individual usage patterns vary. To simplify the model, we consider the total number of calls and messages. SPI is computed using the formula as shown in Equation 6.2 with the terms shown in Equations 6.4 and 6.5.

$$SPI = W_1 \times I_1 + W_2 \times I_2 \tag{6.3}$$

I_1 − Tie Strength based on Call Logs
I_2 − Tie Strength based on Message Logs
W_x − Respective weights

$$I_1 = \frac{\text{Number of calls to one person}}{\text{Total number of calls made}} \tag{6.4}$$

$$I_2 = \frac{\text{Number of messages sent to one person}}{\text{Total number of messages sent}} \tag{6.5}$$

6.2 Design 2: Football ranking based SPI

Explicit interactions of users in their social network gives the user the impression that, they have full control over their social network. However explicitly manually entering this information is time consuming. Also, users would need to organize and structure their social network by classifying, rating, and categorizing their relationships. A study has shown that only 16% of people categorize their contacts in groups on their mobile phones [80]. At the time of writing this thesis, social networking sites such as Facebook, Google+, and others offer support for grouping by allowing user to explicitly classify and organize their network structure. However, research has shown that relationships are dynamic [81, 82], thus one periodically needs to update the grouping created for privacy purposes because of changes in relationships [83]. Hence a photo sharing application cannot be implemented by assuming the availability of grouping of a user's contacts.

In this thesis, a ranking algorithm is implemented which automatically rates the people with whom a user has interacted through his/her smartphone. Interaction data such as call logs and message logs from the user(s) smartphone is used to arrive at a ranking for each person whom a user has interacted with in the past. A football ranking method has inspired this contact rating algorithm, thus complementing the work done in [84]. Each friend of the user is treated as a team, and a score is allotted to them based on interactions (referred to as an interaction value as shown in Equation 6.6). One or more sources of interaction data available in smartphones enable us to infer an interaction value.

$$i_{t(a,b)} = \alpha \times c_t + \beta \times m_t + \gamma \times p_t + \epsilon \times v_t \tag{6.6}$$

This interaction value between user *a* and user *b* for the time period *t* is the sum of all possible interaction values derived from all interaction sources. In Equation 6.6, c_t refers to the interaction value derived from voice calls over time period t. while m_t, p_t, and e_t refers to the interaction values derived from interactions via text messages, physical face to face interaction, and video calling respectively over a period of time *t*. Additional types of interactions can be added depending on the capabilities of the user's smartphone and their social context. In this equation, α, β, γ and ϵ are weights of interaction values. Intuitively, an e-mail interaction appears to be more distant than a voice call. A voice call has been shown to be less effective in personal relationships than meeting in person [85]. Considering these differences we propose to assign specific weights to certain types of interactions based on earlier studies [85, 86].

6.2.0.1 Interaction Evaluation

In this thesis project, ranking of user's friends is based on interactions through voice calls and text messages as per Equation 6.6 (rewritten as Equation 6.7). We tested this approach on individual's smartphone, and evaluated the results by collecting feedback from a number of smartphone users. Interaction data is weighted based on the importance of different modes of interaction. The type of interaction determines the importance of each mode of interaction. We assume that voice calls are more important and have a greater personal impact, thus assigned a weight of 1 ($\alpha = 1$). Interactions through text messages are assigned a weight of 0.5 ($\beta = 0.5$). For a user who uses voice calls and text messages equally, our evaluation shows that the allotted weights give a reasonable ranking (see Section 8.3). These values of interaction weights can be changed depending on the social context.

$$i_{t(a,b)} = \alpha \times c_t + \beta \times m_t \tag{6.7}$$

6.2.0.2 Rating of a Contact

Robin Dunbar claims that the strength of friendships correlate with the number of interactions among friends [87]. In the C2S system, friends are ranked based on interaction score calculated using Equation 6.7. Any individual whom a user interacts with via their smartphone is referred as one of his/her contacts. This subsection discusses the method of rating a user's contacts based on the interaction scores computed using Equation 6.7. Contacts with a interaction value for a period t compete against each other. Contacts with a higher interaction score win against the other contacts with a lower interaction score within the same period. Thus, each contact competes against all of the user's other contacts to secure a high enough ranking to be placed in the user's top friends list. The sport ranking methods, in which teams win or loose against each other, can be utilized to rank friends [84]. The most common sports ranking methods utilize the team's winning percentage to calculate the ranking of the teams (contacts in this context). Winning percentage is the ratio of the number of matches won to the total games played (as shown in Equation 6.8). This method can be used to find the rank of a contact c of user u by using Equation 6.8.

$$r_c = \frac{n_{w,c}}{n_{w,c} + n_{l,c}} \tag{6.8}$$

where $n_{w,c}$ and $n_{l,c}$ are the number of matches won and lost by contact c of user u. This method relies on the number of times the contact had a higher interaction value than others and the total number of comparisons made. This method is biased towards recent interactions with a contact because a newly made contact will have participated in fewer comparisons. Therefore if a user makes a new friend and interacted with him/her frequently the newly established contact will have a 100% winning ratio, as he/she will have participated only in a few recent comparisons. Hence, the need arises for complicated ranking methods such as Colley [88] and Massey [89] ranking algorithms.

Analysis by Chartier et al. [90] shows that Colley and Massey algorithms are *less sensitive to changes*. This is desirable property for any ranking algorithm in the context of social network. For example, if a person moves to another town or goes to work in a new company, then the social network of the person may be extended to include new friends if the person spends more time with these newly made friends. If a sensitive algorithm is used, then a newly made friend may have a high tie strength compared to older good friends, but this rarely happens in real social networks [91].

Both the *Colley and Massey algorithms take current ratings into consideration* while calculating the rating of a contact. The Colley method is based on the results of comparisons of either a win or loss. In contract, the Messey method uses the actual game score and home field advantage which are not relevant to social networks. Social network allows users to interact with each other even if they are not physically present in the same location. Therefore, the Colley method is utilized to calculate the rating of the user's contacts. Colley modified Equation 6.8 to include the initial ranking of all teams (contacts) to 0.5 as follows:

$$r_c = \frac{1 + n_{w,c}}{2 + (n_{w,c} + n_{l,c})} \tag{6.9}$$

However, this rating does not depend on the rating of other contacts. In sports, the ranking of a winning or losing team also depends on the rank of the opposite team. Therefore following Colley we define the rating of a contact of a user as follows [88]:

$$(2 + (n_{w,c} + n_{l,c}))r_c - \sum_{k \in F_u} r_k = 1 + (n_{w,c} - n_{l,c})/2 \tag{6.10}$$

where F_u is the set of all contacts with whom a user communicated in the period being considered.

With this definition, the rating of a contact depends on the rating of other contacts that reflects the relationships in social networks. Equation 6.10 is a system of N linear equations with N variables. This can be solved using matrix computations. We can rewrite Equation 6.10 as:

$$C\,\vec{r} = \vec{b} \tag{6.11}$$

where $\vec{r}_{(n\times 1)}$ is a column vector of all ratings r_c of contacts c of user u.
$\vec{b}$ is a column vector of the right-hand-side of Equation 6.10 and is defined as follows:

$$b_c = 1 + (n_{w,c} - n_{l,c})/2 \tag{6.12}$$

The matrix $C_{n\times n}$ is known as Colley coefficient matrix and is defined as:

$$C_{ij} = \begin{cases} 2 + (n_{w,c} + n_{l,c}) & i = j \\ -n_{ij} & i \neq j \end{cases} \tag{6.13}$$

where n_{ij} is number of times an interaction value of a contact c_i is compared with c_j. The rating of a contact c of the user is defined as:

$$r_c = \frac{1 + \frac{(n_{w,c} - n_{l,c})}{2} + \sum_{k \in F_u} r_k}{2 + (n_{w,c} + n_{l,c})} \tag{6.14}$$

In this calculation, the rating of a contact c of the user relies on the rating of other contacts k of user u. Hence interacting with someone more than a person's best friend (i.e., their top rated contact) has a high impact on a contact's own rating. The initial rating of contacts is 0.5 (i.e., a median of 0 and 1). Depending on the comparison of interaction values, a win increases and a loss decreases the rating value. This makes system less sensitive to changes. Therefore communication between a user and friend needs to retain a high rating for several rating periods until it has significant impact on the ratings of others.

To summarize Colley's method does not give a high rating to a contact who only has a high interaction value for a short period of time. These ratings can be sorted and contacts of a user can be ranked. These ratings are updated periodically. The level of intimacy can be decided according to the calculated ratings. We propose the following requirements for the ranking algorithm:

1. It should be unbiased with regard to interactions with newly made contact,
2. Uses minimum of assumptions, and
3. Considers the rating of other contacts/friends.

6.3 Characterizing User Preferences

Learning and characterizing user preferences is fundamental in any user centered application. User preferences can be learnt by gathering clues from the user's calendar, smartphone application usage, and by monitoring the user's activity in OSN(s). Alternatively, a digital life agent can be deployed on the smartphone that can infer user's activity along with other cues (without manual user intervention).

6.3.1 User Calendar

The user's calendar is read, and an index is constructed about frequently occurring events. This index consists of the name of the activity, domain, and the frequency of occurrence along with the details of participants. The domain hints at a user's possible preferences and the list of attendees hints at the social proximity of the user with regard to them. Lucene Index [92] is constructed by parsing the title and description of the calendar and clustering them in the domain. For example, as per the calendar, if a user frequently plays cricket, it is likely that the user likes cricket.

6.3.2 Application Usage

The applications present in the user's smartphone hints at the user's affinities. Knowledge of the type of application that a user uses can help the system decide upon this user's preferences. According to Seneviratne, and Mohapatra, the kind of application that a user uses may indicate what kind of person he/she is and what he/she likes [93]. The C2S system track applications that a user frequently uses in his/her smartphone. However, the C2S system does *not* consider the list of all installed applications in user's smartphone. Given the today's smartphone capabilities, a user may have downloaded many different applications. However, seldom does a user use all of these installed applications. Hence tracking only those applications that a user frequently uses helps identifying this user's preferences. Application usage frequency per domain is constructed as shown in Table 6.5. A dictionary of applications classifying each application into one or more domain was manually created for the list of applications that were found to be frequently used by the participants in our experiment.

Table 6.5: Frequency of application Usage

	Domains		
Applications	Cricket	Messaging	Images
Cricbuzz	4	-	-
Yahoo Cricket	5	-	-
WorldT20	1	-	-
Whatsapp	-	3	-
Wechat	-	5	-
Camera	-	-	2
Instagaram	-	-	4

Chapter 7

Implementation

This chapter gives details of the implementation of the prototype C2S application. Figure 7.1 gives an overview of architecture of this C2S application. This chapter also provides justification for each of the design decisions that have been made.

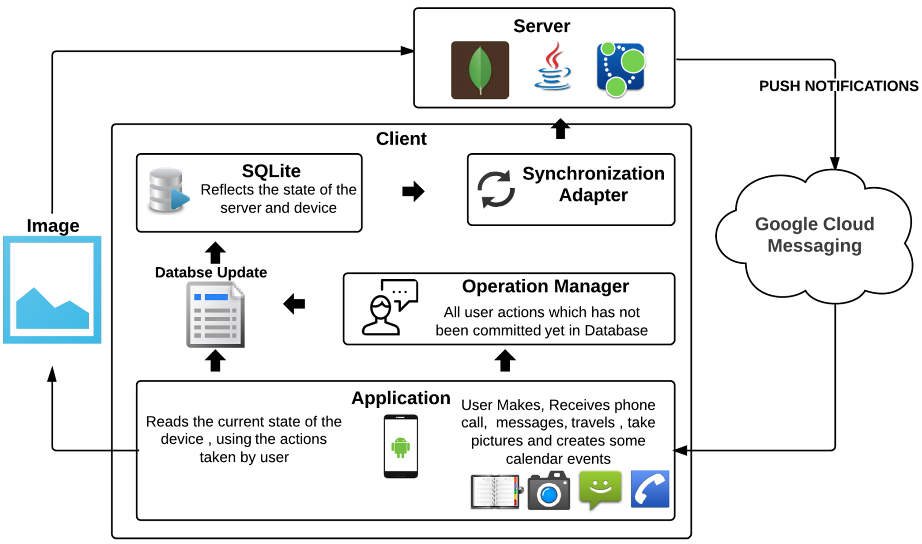

Figure 7.1: Overview of C2S Application

Questions to be answered include:

1. How long should data be stored in the server? Should all computations be offloaded to server?

2. What is the time line of the data that is to be captured? 1 week or 1 month?

If data is to be stored for 1 month or 6 months, then how does one evaluate the importance of a new contact?

3. How frequently should synchronization occur? Every hour, every 6 hours, or once at the end of each day?

4. How is the data at the server kept consistent with the client?

7.1 C2S Client

C2S client is composed of an application manager supported by operation manager, synchronization adapter, and SQLite (Content provider). The following paragraphs give details of the implementation of a prototype C2S client.

7.1.1 Application

A prototype C2S client was implemented for Android platforms (see Section 4.1 for further details about this platform and why it was selected). This client application monitors the user's device state. Device state refers to the type of application that the user uses and the frequency of each application's usage, and the number of calls this user makes, and to what places does this user travels. Location, application usage, and entries in the user's calendar help to determine the user's preferences. Calls and messages logs help to rank the user's contacts, leading to the construction of this user's social interaction graph. The type of application a user frequently uses indicates affinity to the particular context of each application. The context that we will focus on in this thesis is the category sports. Similarly, various contexts of each application can be included. The frequency of communication is used to estimate the closeness between individuals and also their willingness to share any information [47].

7.1.2 Operation Manager

The operation manager module functions as a bookkeeper that records all the user's activities into a database. First the user's application usage history is recorded. A snapshot of all open applications in foreground is taken and recorded in a local SQLite (see 7.1.4 for further details about its implementation and usage) based storage. This snapshot is taken when the user turns off the screen after any activity. For example, if a user likes cricket, then that it is anticipated that the cricket related application remains active in the background so that the user can get updates. Because the snapshot is only made when the user turns off the screen the client application is more battery efficient (i.e., it consumes less battery power than if it ran periodically in the background). If periodic snapshots are taken, more battery is consumed and we might miss the user's activity. We will not know what kind of application the user uses and when unless we use a short polling period - thus further increasing battery power consumption. Our choice of when to perform a snapshot avoids running our application during the night and saves battery cycles during long periods of inactivity.

The operation manager also updates the device's location data in a SQLite database whenever a change in location is detected. The location information is stored as Latitude, Longitude coordinates by fusing measurements from different physical sensors. In this thesis project, a software based sensor namely "fused location provider" is used to infer the location [69]. This sensor is configured for battery saving mode in order to minimize battery power consumption , while maintaining medium accuracy. Battery saving mode uses Wi-Fi access point's Service Set Identifiers (SSID) and mobile base station identifiers to derive the device's location. The user can change the mode from battery saving mode to high accuracy mode. In high accuracy mode, the GPS sensor along with other available sensors are used to calculate the device's location with higher accuracy. The fused location provider (also known as location API) fuses data from the different sensors to determine the device's location detail with the required high accuracy. The C2S application does not rely on precise location data. Having a location with reference to a point of interest or places of interest is sufficient to enable the detection of user affinity and to compute physical proximity. Location details are synchronized with the server via a content provider using a synchronization adapter. Location data is removed from the local SQLite DB after the server acknowledges the reception of data.

7.1.3 Synchronization Adapter

This subsection discusses when the data is transferred to the server. Android's Synchronization adapter [94] is used and a synchronization service has been implemented. Android's synchronization adapter queues all data transfer requests. The actual data transfer is initiated when there is network connectivity. These data transfer requests are bundled together with other applications' data transfer requests, thus saving battery. Whenever a change or update from a content provider is observed, a data transfer request is created and handled by Android's synchronization adapter. Using this synchronization adapter prevents C2S client from activating a dedicated channel for data transfer, thereby reducing unnecessary energy consumption wasted in high power states [95] after completing data transfer activity. One more significant advantage of using this synchronization adapter is avoiding correlated data transfer peaks at the C2S server. Such peaks could overload the server with data arriving from all of the C2S clients at the same time. Since the clients will interact with C2S server at different times, the load on the server is expected to be uniform or at least it will avoid large peaks. Each data transfer request is composed of an array of JSON Objects (such as the example shown in Listing 7.1).

Listing 7.1: JSON Array containing location coordinates of the user

```
{"locationlogs": {
  "id": "xxxx12345",
   "location_update": [
     {"Latitude": "48.1224855", "Longitude": "-1.6285135",
           "TimeStamp":"2014-04-12T08:50Z"},
     {"Latitude": "48.1224855", "Longitude": "-1.621457",
           "TimeStamp":"2014-04-12T08:50Z"}
  ]
}}
```

7.1.4 SQLite

Android's content provider provides ways for applications to store data locally in their smartphones. It uses the SQLite [96] database engine to handle all CRUD queries. An observer is implemented for the content provider which initiates a data transfer request via the synchronization adapter. The content provider is implemented using four tables, as described in Figure 7.2.

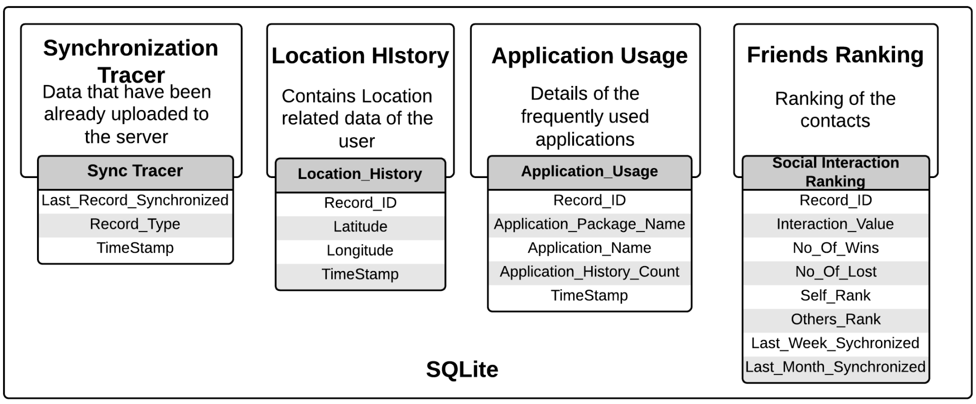

Figure 7.2: Content Provider Implementation

The synchronization tracer table reflects the status of the server by storing the identifier(ID) of the last record that the C2S server acknowledged along with its type. The type of record refers to a location record, a application history, or contacts rating. Another advantage of the content provider is that other applications can access this data given the appropriate permissions, thus in the future if an application wants to use a user's friends rating, this information could be retrieved from the content provider.

7.1.5 Contacts Rating

The C2S client uses Colley's football team ranking method for rating contacts. An interaction value is assigned to each of the user's contacts. The interaction value of the user's contacts is compared with all of their other contacts, wins and loses are counted, and a rating is assigned based on Equation 6.14 (A description of this method was given in Section 6.2). In this implementation, due to the limited duration of this thesis project, the interaction value is computed using only SMS and voice call based interactions. Equation 6.7 is used with value of $\alpha = 1$ and $\beta = 0.5$. Call logs and message logs are retrieved from the user's smartphone, and an interaction value is computed as per Algorithm 1. The equivalent implementation of Equation 6.7 is given in Equation 7.1.

$$i_{t(c)} = \alpha \times Number.of.Voicecalls + \beta \times Number.of.Textmessages \quad (7.1)$$

where $i_{t(c)}$ is the interaction value of a contact c for period of t. For example, if the period is one week, then the contacts c of user u compete against each other every week, and their ratings are calculated once each week. The rating computed is sent to the C2S server using the synchronization manager. The rating algorithm runs locally on the user's smartphone.

7.2 C2S Server

The C2S server is built on the Java Platform, enterprise edition (J2EE) and exposed through a REST API using JAX-RS*. The C2S server receives client requests and process them and then notifies the client about the status of the request. The following subsections will detail the implementation steps and frameworks used.

*JAX-RS: Java API for RESTful web services which provide support in creating services according to REST architectural pattern.

input : C - Set of Contacts with initial rating of 0.5
L - Set of Log data from Smartphone
λ - Timeline
α - Weight for interaction via Voice calls
β - Weight for interaction via Text Messages
SD - Start Date
ED - End Date
output: C - Set of Friends with their ranking

```
while SD < ED do
    /* First Phase */
    Calculate interaction value int of all contacts c in C
    while (L.Date > SD) AND (L.Date ≤ SD + λ) do
        if (L.type == calls) then
            c_int = c_int + α
        end
        if (L.type == TextMessages) then
            c_int = c_int + β
        end
    end
    /* Second Phase */
    Calculate rating r of all contacts c in C
    while ∀c_i ∈ C do
        while ∀c_j ∈ C do
            if c_i.interaction < c_j.interaction then
                c_i.lost = c_i.lost + 1
                c_j.won = c_j.won + 1
            end
            if c_i.interaction > c_j.interaction then
                c_i.win = c_i.win + 1
                c_j.lost = c_j.lost + 1
            end
            if c_i.interaction == c_j.interaction then
                c_i.draw = c_i.draw + 1
                c_j.draw = c_j.draw + 1
            end
        end
        r_i = (1 + (c_i.win − c_i.lose)/2 + Σ_{k∈C_u} r_k) / (2 + (c_i.win + c_i.lost + c_i.draw))
    end
    SD = SD + λ
end
Sort ∀c ∈ C based on r_c
return C
```

$$r_i = \frac{1+\frac{(c_i.win-c_i.lose)}{2}+\sum_{k\in C_u} r_k}{2+(c_i.win+c_i.lost+c_i.draw)}$$

Algorithm 1: Rating Calculation

7.2.1 Application Logic

Every request contains a JSON string as its payload. This JSON string is parsed and persistently stored in a MongoDB. Every request contains some user data along with the user's unique ID. Each request may contain one or more set of tuples containing this user's data. Each set of tuples is identified by a record ID. For example, a tuple can contain the user's location coordinates. A request may contain one or more sets of this user's location coordinates. Each set of tuple is identified by a continuous sequence of numeric IDs. Hence if the server receives one or more requests or one request with many tuples, it sends an acknowledgement of last record, thus the client learns the status of the server. This helps to avoid duplicate requests issued by the client, thus reducing the bandwidth consumption. According to Carroll and Heiser this has a direct impact on minimizing the battery consumption of the client [97].

7.2.2 Persistent Layer

MongoDB provides the persistent storage for the C2S server. It stores all the data received from the C2S clients. In this implementation, six tables namely, user-profile, user-location, user-application, user-calendar, application-dictionary, and user-interactions are used to store the data. The user-profile table contains each user's unique ID, their e-mail addresses, and other account details from their smartphone. The C2S client application builds the user's profile by accessing the user's account details with the relevant permissions. The client sends this data to the C2S server, and this data is stored in the user-profile table. User account details include the user's e-mail address along with IDs of any available OSN accounts. The user's OSN accounts are to be helpful in inferring the user's preferences [76] as they include the information such as the user's favorite sport and teams, places visited, events attended and many other potentially useful bits of information. In this implementation, due to the limited duration of this thesis project the user's OSN accounts were not used.

The user-location table contains location details of places where this user has been. As noted earlier these location details are not collected periodically, but only are collected when there is a change in location. The user-application table logs the frequently used applications based on a snapshot taken every time the phone's screen is turned on. In MongoDB, one can do aggregation over data to perform analytics. For example, frequently used application can be aggregated on a weekly basis, daily basis, or monthly basis. However, there is a challenge of when to do this aggregation. Performing aggregation when receiving a request may delay the response to a request, especially when the data set grows. Hence in this prototype implementation, pre-aggregation is used. The schema shown in Listing 7.2 was designed to meet this requirement. With this schema, statistics always reflect the updated user's preferences.

Listing 7.2: Schema of Application history table

```
{
  "id": "xxxx12345",
     "application id1":{
         "Usage Count": "26", "Week No": "36"},
     "application id2":{
         "Usage Count": "16", "Week No": "36"}
     "Usage statistics:{
         ["Month Number": "5",
                "Application Id": "wats app",
                "total usage count per month": "26",
                "average application usage per month":
                    "6.25"],
         ["Month Number": "5",
                "Application Id": "cric buzz",
                "total usage count": "16",
                "average application usage per month":
                    "4.83"]
          }
}
```

The Application-Dictionary maps each application to a domain. For example, the *cricinfo* application belongs to the domain cricket. With this mapping information, we can know that if a user frequently uses 'cricinfo' application, then this user has affinity towards cricket. For this prototype, a dictionary of applications was built by manually categorizing the applications. However in a practical implementation, it would be possible to mine all of these application categories from application stores as done in [93].The user-calendar table is used to store the calendar details of a user. However, in the prototype only the title, descriptions and list of guests are extracted and used.

7.2.3 Constructing Social graph of the User

Neo4J is used to build the social graph of the user using the data stored in the MongoDB. The calendar and location data are deleted from the MongoDB after being included in the user's social graph. This ensures that a user's private data is stored for as little time as possible. The graph in Neo4J helps to find relevant contacts of the user who are likely to be interested in receiving a given picture. The following paragraphs give more details about the construction of the user's social graph and how this graph is utilized in this project. Nodes are categorized based on labels. A label is a name that organizes nodes into groups. The graph in Figure 7.3 shows two nodes labeled as user. Each user nodes is connected through typed, directed relationships. Each user node is connected to another user's node with relationship type "*KNOWS*" with the rating computed using the rating Algorithm. This implies that user1 *knows* user2 and has an interaction rating of 0.6. The

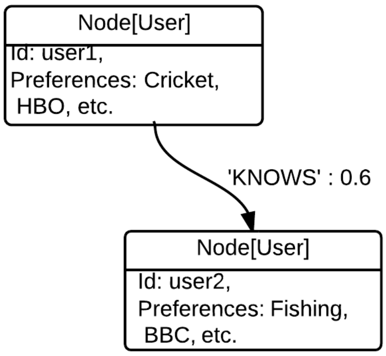

Figure 7.3: Graph consisting of a user and his/her contact

rating is included as an attribute of the property of the relationship 'KNOWS'. The graph evolves as more data about the user's application usage and calendar are used to augment this graph. Figure 7.4 shows a node for user1 with an edge leading to a calendar event via the user's own e-mail address. Similarly, another node for user2 has an edge leading to the affinity cricket based upon application usage.

7.3 PictureSharing

The first step in evaluating the prototype was to install the C2S client on each user's smartphone. Every client installation ensures that the user registers with the Google cloud messaging service. These registration details are sent to the C2S server. This messaging service is used by C2S server to send notifications to clients. Once the client is installed, the user can choose an image from any of installed applications (such as flickr [98], instagram [99], Phone's gallery, or any third party image viewing application). Once the user decides to share an image via the C2S client, the image is sent to C2S server. The metadata of this image

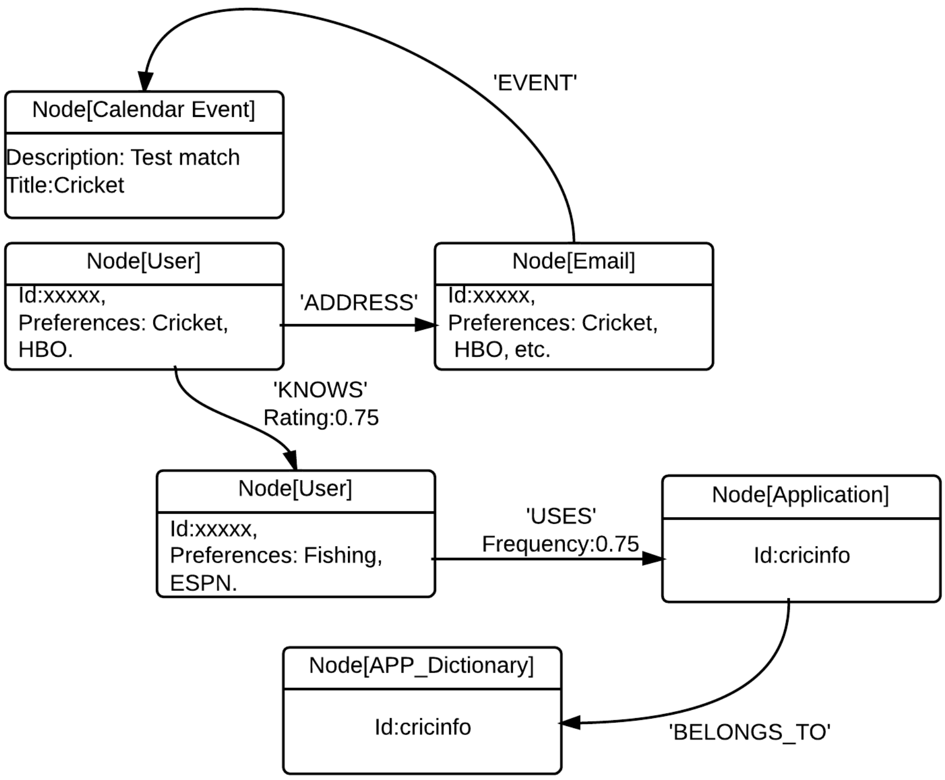

Figure 7.4: Graph illustrating the users profile

is parsed using an open source library named meta-data extractor [100]. In this implementation C2S relies on the metadata of the picture, specifically it uses the title, description, and GPS co-ordinates. In a practical implementation, one could do image processing to infer further information about the objects in the picture. One such image processing API is the clarifai API [101].
The keywords from the extracted metadata are used as search parameters to find those contacts of the user who would be interested in receiving this picture. Listing 7.3 gives an example of a search query in Neo4J. Listing 7.3 shows an example query for a user who wishes to share an image of a cricket match to his/her contacts who also like cricket. In line two, searches for contacts of this user's who have calendar events with the keyword 'cricket'. An apache Lucene* index is used for indexing nodes in Neo4J. Line four searches for applications

*Apache Lucene [92] is a high performance, and full-featured text search engine library related to cricket. Line six matches the user nodes who uses applications in the Cricket domain. Line seven identifies the users who had any event in the past related to cricket. Line nine returns user IDs who are in contact with the user with the rating of 0.5 and above with an affinity towards cricket. It ensures that images are sent only to the user's contacts who have an affinity towards cricket.

```
1 START
2 Event=node:node_auto_index("title:cricket* OR
3 description:cricket*"),
4 Domain=node:node_auto_index("id:cricket*")
5 MATCH
6 (User1)-[:USES]->(Application)-[:TYPEOF]->(Domain),
7 (Event)-[:ATTENDED]-(Email)-[:EMAIL_ADDRESS]-(User)
8 RETURN
9 Event,Application, User,User1, Domain,Email
```

Listing 7.3: Neo4J Query illustrating node search for users having affinity to cricket

After identifying the user's contacts who might be interested in receiving the image, the MongoDB is queried for the respective Google cloud messaging's registration identifiers and a notification containing a Uniform Resource Locator (URL) of the image is sent to the client. The C2S client downloads the image and notifies the user. Once all the notified clients have downloaded the image, then the image is deleted from the server.

7.4 Dataset Generation in Neo4J for prototype Implementation

This subsection gives overview of the data set generated and used for demonstrating the prototype implementation of C2S. Figure 7.5 shows the social graph containing 6 users represented as six nodes. Each nodes are connected via vertex with property "INTERACTSWITH". This property has the value of rating which is computed using the algorithm. Figure 7.6 presents the the modeling of user's social graph including the details of the calendar events linked via their e-mail address. Nodes in yellow color represents user's email ids and orange color nodes are the calendar events linked to the user via their email ids with vertex property as "ATTENDED". We also modeled application usage as nodes in user's social graph. So each application used is represented as nodes linked to the user via a vertex with property 'USES'. Each application belongs to a category which hints about the user's affinity.

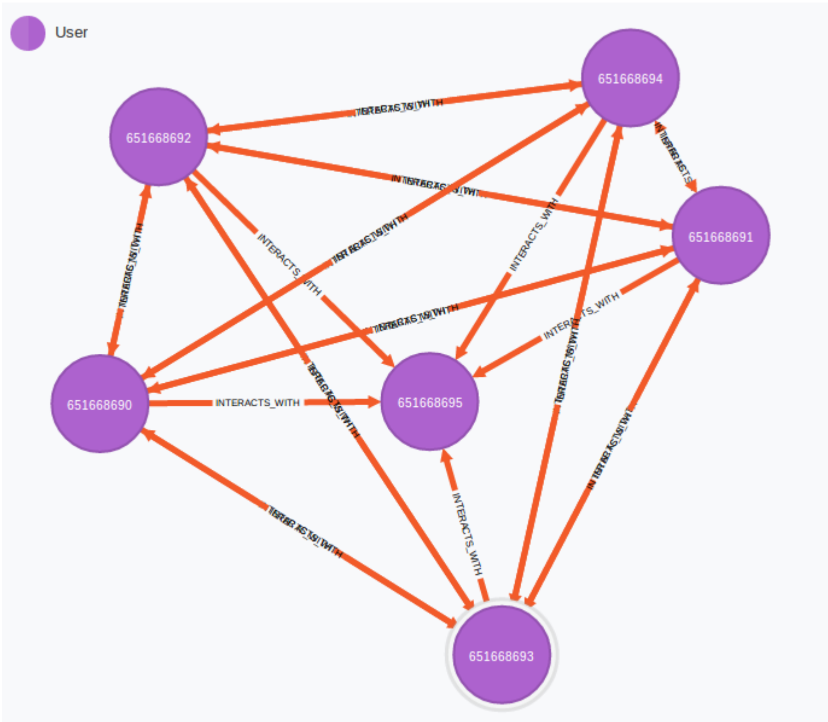

Figure 7.5: Social graph of users constructed using Neo4J

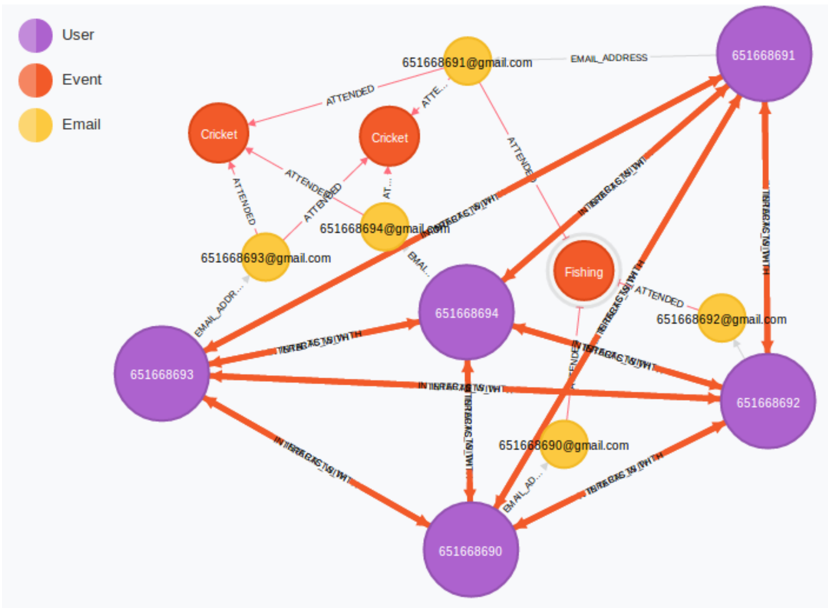

Figure 7.6: Social graph of users with calendar details

In Figure 7.7 we see that nodes of application are linked to domain with property of vertex as ‘TYPEOF’. For example, cricbuzz is an application which gives update about cricket game to user. User uses cricbuzz which belongs to domain cricket. Hence we say that user likes cricket. Figure 7.8 gives the

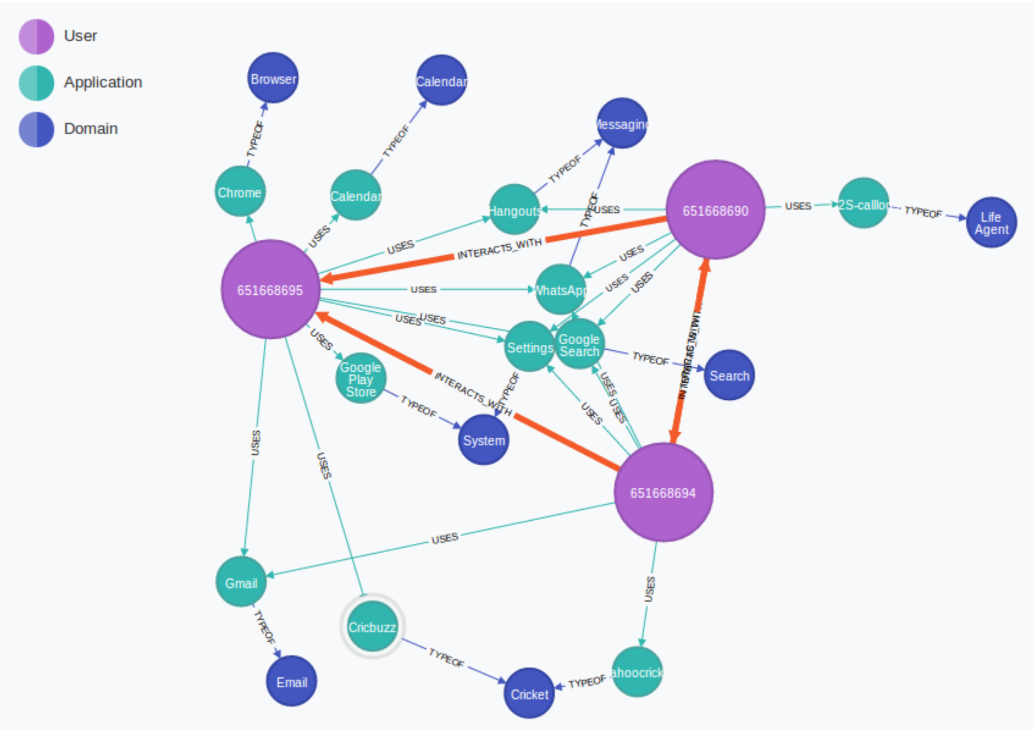

Figure 7.7: Graph of application and its domain

overview of entire hypothetical data set for demonstrating the prototype of C2S application.

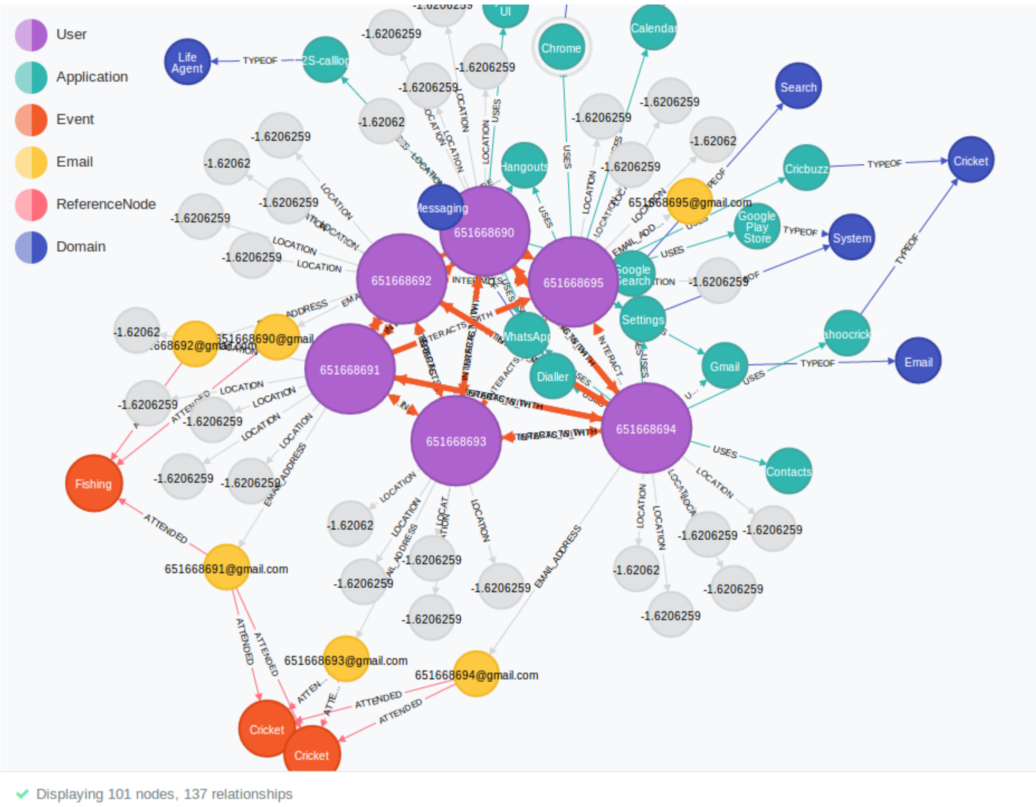

Figure 7.8: Graph representation of entire user data

Chapter 8

Evaluation

This chapter presents details of experiments and simulations carried out for evaluating the Colley football ranking based SPI implementation. The C2S application is evaluated in two phases. In the first phase, an SPI implementation is tested on the Reality mining data set [3]. We evaluate how much data is required and what is a reasonable timeline period to use for rating computations. In the second phase, we conducted experiments by inviting volunteers with a smartphone to download and install the prototype C2S client and collected feedback based on a questionnaire.

8.1 Tests on a public data set

Rating of people whom a user interacts with can be computed using Algorithm 1 with various input schemes. We discuss four possible different schemes and evaluate them with respect to which finds the closest match to real life social proximity of the user. We refer to people whom the user interacts via text messages or voice calls as *contacts*. We use the term *time period* to refer to the minimum duration of log data used for computation of ratings of user's contacts. The word *window* refers to the duration of log data for which interaction value is computed and using that interaction value, a rating is computed for same window. A rating is computed for each window within the time period. The window is formed by a user's calls and message log data.

Scheme 1: Every week for six months

This scheme considers time period of six months and window of one week. Interaction values are computed on a weekly basis and contacts are rated every week. A rating is computed for each week within the time period of six months.

Scheme 2: Every month for six months

This scheme infers the contact's rating by using the time period of past six months with the computation of interaction values for each window of 30 days.

Scheme 3: Every week for three months

This computation scheme has time period of three months. The interaction values and its respective rating are calculated over each window of 7 days within time period.

Scheme 4: Every month for three months

This scheme uses data of the past three months with window of a month to compute ratings of a user's contacts.

8.2 Sensitivity of parameters

The dataset contains anonymized call logs and message logs of 94 persons participated in the Reality mining study conducted by Eagle, Pentland, and Lazer at Massachusetts Institute of Technology Media laboratory [3]. Among the 94 participants, 82 participants made some calls and sent messages during the study period, and their details are recorded. The data set also contains the self reported data about social proximity where participants reported about their close circle of friends. In this evaluation, users call logs and message logs are processed and rated using Algorithm 1. Missed calls and calls to voice mail are not taken into consideration. Figure 8.1 shows the ratings of 110 contacts of a user who made 1930 voice calls and sent 38 messages in six months time. The ratings are based on the IDs of the contacts (i.e. The given contact has same ID in each of the scheme).

The ratings of this user's contacts are calculated using four different schemes as explained in previous sections. A rating for each contact is calculated every week based on the interaction he/she had with the user as per scheme 1. The green curve in Figure 8.1 represents the ratings of this user's contacts at the end of six months period. Secondly, the contacts are rated every month based on the interaction he/she had using Scheme 2. The red curve in the figure shows the ratings of this user's contacts at the end of six months period. The black curve in the figure represents the ratings calculated on a weekly basis for past three months periods as per scheme 3. Similarly, the blue curve represents the ratings of contacts computed on a monthly basis for past three months. The red curves indicates few high ratings (with just a few peaks), but the peaks of blue and black curves indicates poor ratings for a few of this user's contacts who were rated highly by other schemes. This leads to conclusion that rating Scheme 1 - that considers six months of data with weekly comparisons will yield more consistent ratings. We will evaluate how realistic these ratings are in following sections and how different parameters affect the rating of user's close friends which we are concerned about.

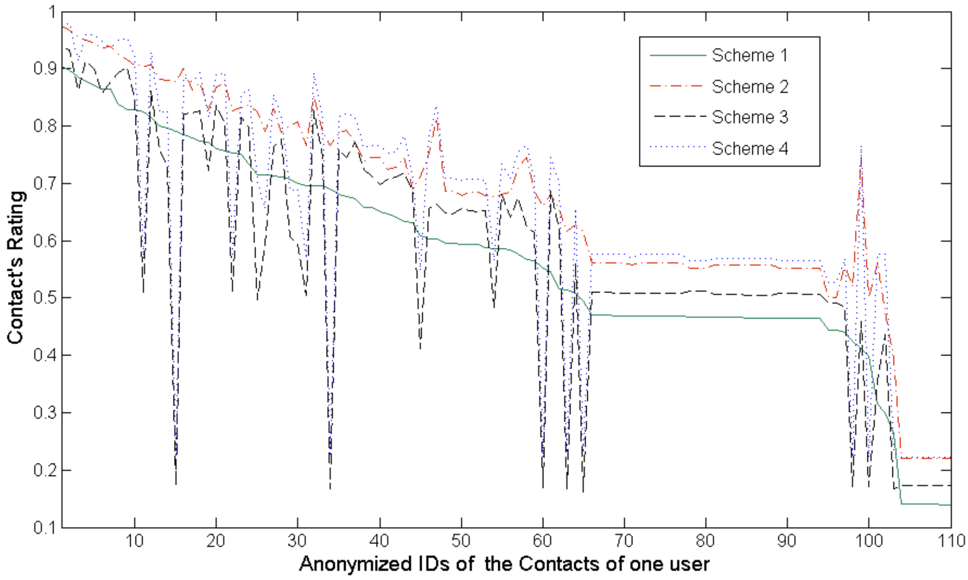

Figure 8.1: Comparison of different rating schemes for one user's contacts

In the Reality mining dataset, user's self reported their social proximity (circle of best friends) with contacts and we evaluate which scheme best fits the user's actual self reported social proximity. Figure 8.2 shows a comparison of the ratings of a user's best friends using the 4 different schemes (as explained in Section 8.1). The ratings computed using Scheme 3 and Scheme 4 gives lower ratings (lower than 0.5) to contacts (with IDs 2 and 5) whom the user is in with close social proximity, compared with the other two schemes. Also, the evaluations of data of other users in the dataset shows that ratings for best friends have a score above 0.72.

In the Figure 8.3, we show the ratings of contacts of one user, but other user's best friends also have similar ratings. Individual bar graphs in Figure 8.3 demonstrates the ranking of a user's best friends using different rating schemes. The green bar graph hints the rating of a user's best friend using Scheme 1, and the red bar graph refers to the rating of a user's best friends using Scheme 2. Similarly, the black bar graph refers to ratings computed using Scheme 3 and the blue bar chart for ratings computed using Scheme 4. Figure 8.3 illustrates the differences between different rating schemes. Computation using Scheme 3 and Scheme 4 on average gives poorer rating to this user's best friends.

We studied and evaluated all four different schemes and the results shown in Figures 8.1, 8.2, 8.3 indicates that Scheme 1 gives the consistent rating compared to other schemes. Scheme 2 has few peaks whereas Scheme 3 and Scheme 4 completely underrated user's close friends. Therefore, we use the Scheme 1 to compute ratings of a user's contacts, and we evaluate further how the weights assigned to calls and messages are sensitive to the ratings of a user's contacts.

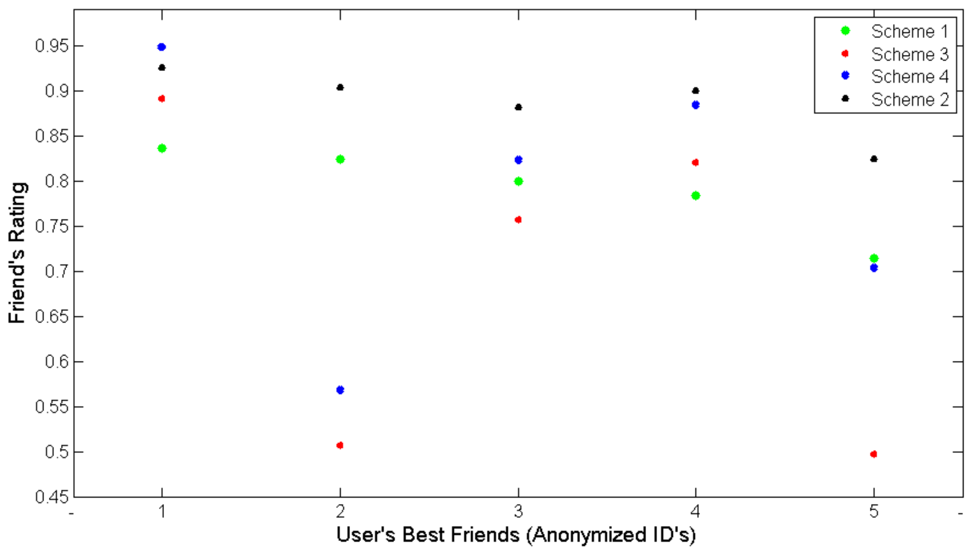

Figure 8.2: Comparison of a user's best friend's rating using different computation schemes

Ratings using different computation schemes rely on the interaction values that depend on the α and β values of Equation 6.7 (i.e., depending on the usage pattern (number of text messages and voice calls users uses to interact via his smartphone)). Figure 8.4 presents the usage pattern of 84 different users. From the figure, it is evident that the usage pattern differs with voice call to messages ratio ranging from 1:1 to 1:3. There are some users whose usage pattern indicates that they use a negligible amount of text messages compared to voice calls.

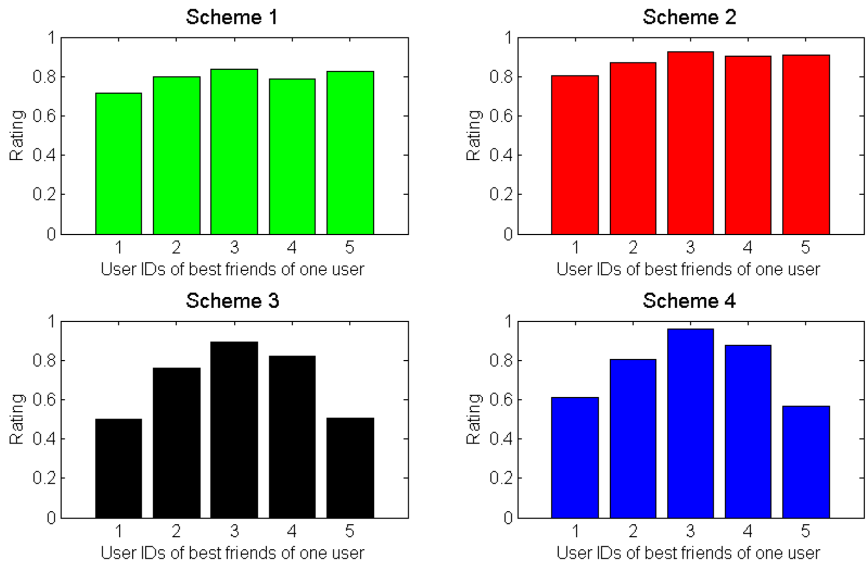

Figure 8.3: Comparison of different rating schemes for one user's best friends

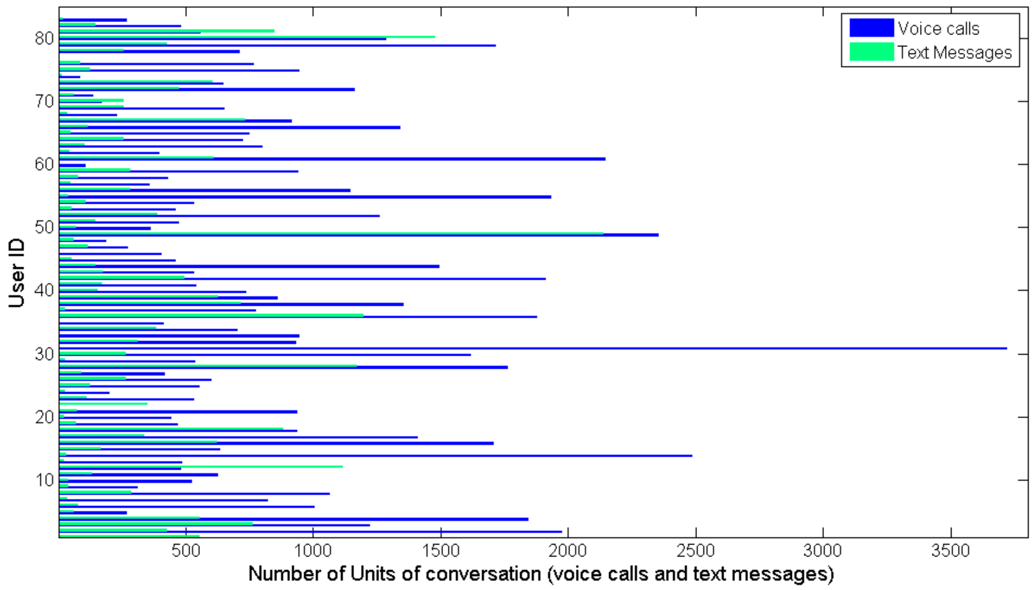

Figure 8.4: Summary of usage pattern of 84 users

Considering different voice calls to text messages ratios, rating of two different user's (with different voice calls to text message usage ratio) contacts are evaluated. Figure 8.5 shows the rating of contacts of a user who made 1284 voice calls and sent 1475 text messages in a period of six months.The ratio of voice calls to messages for this user is close to 1:1. We calculate ratings of this user's contacts using Scheme 1 with different alpha and beta parameters.

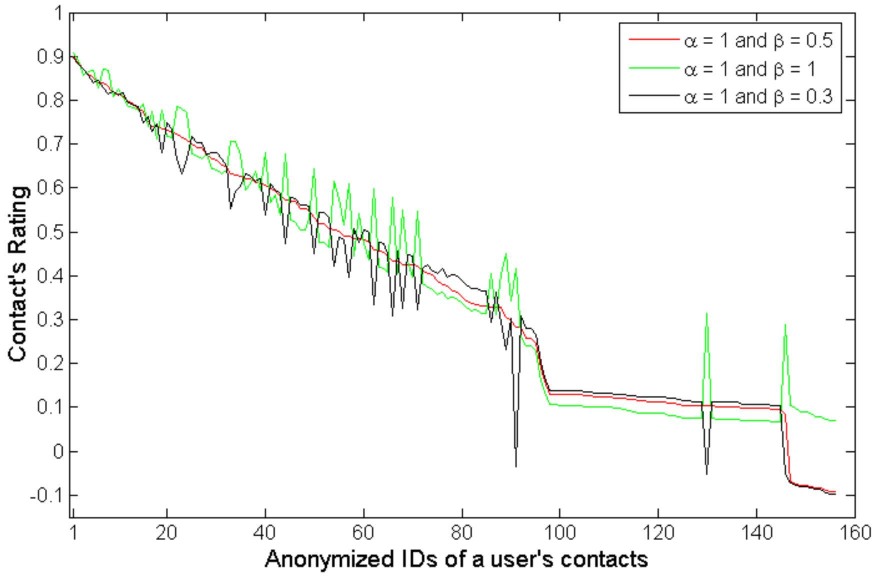

Figure 8.5: Effect of α and β parameters on ratings for user A (1:1 voice calls/text messages ratio)

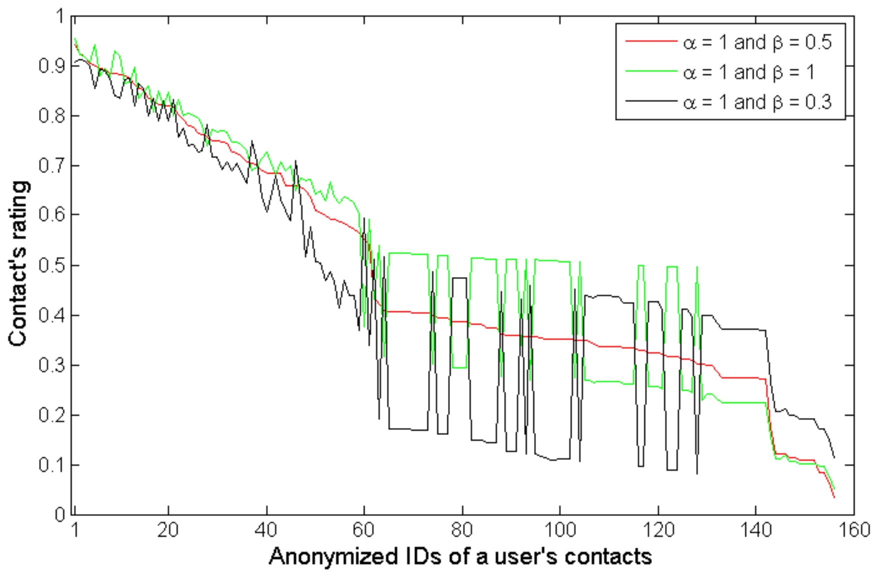

Figure 8.6: Effect of α and β parameters on ratings for user B (1:3 voice calls/text messages ratio)

Figure 8.6 shows the rating of contacts of a user who made 479 voice calls and sent 1113 text messages in a period of six months. The ratio of voice calls to messages for this user is closer to 1:3. We calculate ratings of this user's contacts using scheme 1 with different alpha and beta parameters. Figures 8.5 and 8.6 shows that ratings above 0.6 are less sensitive to changes in α and β parameters. So rating of contacts in a user's close social proximity are less affected by changes in these two parameters.

8.3 User Survey

We conducted another evaluation by inviting people working at Technicolor R&D, Rennes, France to participate as volunteers. E-mails with a brief description about the investigation and a user agreement were sent and volunteers were invited to participate in the experiment by installing the prototype C2S client. These volunteers agreed to fill out a survey questionnaire (see Annexure B). This study was conducted to evaluate the relevance of the proposed rating scheme in comparison with the actual social proximity of users. The study included 16 participants. These participants (10 of 16) preferred to send personal selfies and personal pictures to contacts with ratings 0.83 and above (see Table C.7). 50% of the participants preferred to send pictures of family events to contacts with a rating above 0.8 (see Table C.6). 56% of participants (9 of 16) were inclined to send pictures of sports events/activities to contacts having relevant affinities with the rating of 0.6 and above (see Table C.5). Since this rating method relies on access to a user's log data present in their smartphone, participants were asked how frequently they delete their log data. The majority of the participants (11 of 16) reported that they never delete log data unless it hurts the performance of their smartphone. Also to learn how accurate the rating system is, we computing ratings using all four different schemes and asked users which rating scheme that they feel is more appropriate with actual social proximity to their contacts. We requested them to find any outlier assigned a rating of above 0.5. One of the commonly identified reason for outliers was a recent interaction with a contact for delivery of services, for arranging a meeting, or for some favor. Participants in the study also were questioned about their interaction patterns. Ratio of calls to messages among different participants ranges from 1:1 to 1:100 (as shown in Table C.4). Participants also hinted that they are less interested in contacts which are not available in their Phonebook. Figure 8.7 shows screen shot of the C2S client installed on participant's phones for evaluation of rating of user's contacts. Four buttons are provided to rate the user's contacts using different schemes. Participants were asked to compute the rating using various schemes by clicking the respective buttons, and results are shown as in Figure 8.8.

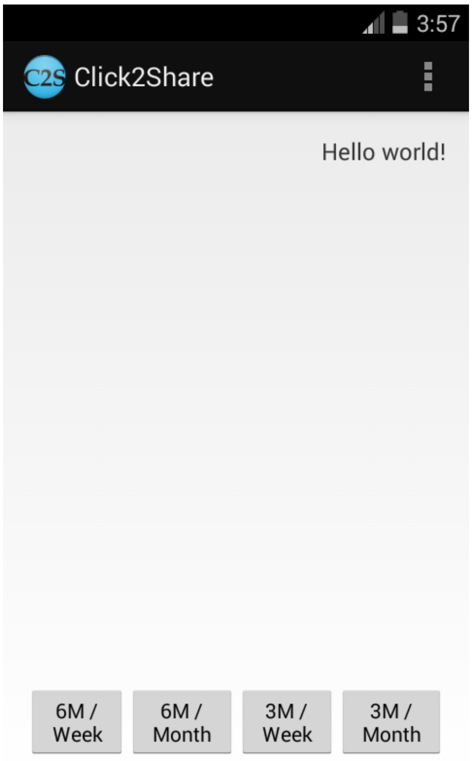

Figure 8.7: Initial C2S screen

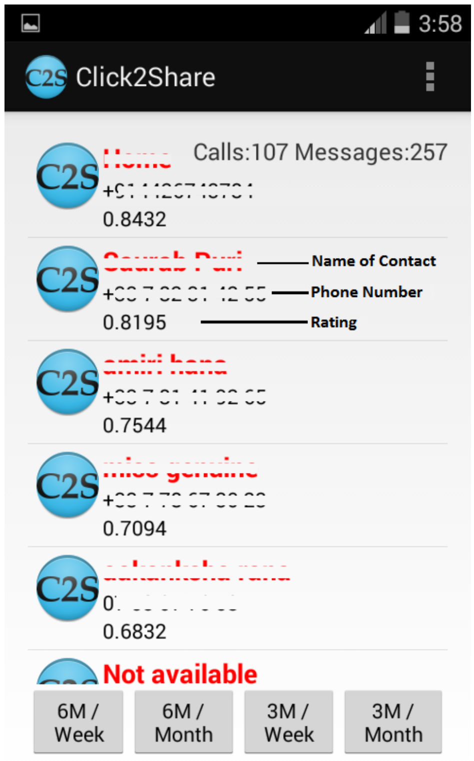

Figure 8.8: Output after selecting a rating scheme (in this case for Scheme 2)

The preference of schemes categorized by age group is shown in Table 8.1. Survey shows 85% of participants (6 of 7) belonging to 45-54 age group prefers Scheme 1 whereas, 66% of participants (4 of 6) belonging to 15-24 age group prefer Scheme 4. Hence we say that age group has an impact on acceptance to different schemes of rating. The proposed system will not yield consistent results if no data or data lesser than period of three months is available.

Ratings are computed based on phone numbers, and linked to the user's phone book and display it on the screen with the contact's name and a picture (if available) to ease the evaluation process for the participants. In a real life implementation, this computation is expected to happen as a background process and the data would be transmitted to the C2S server via the REST API. However, when evaluating picture sharing it was difficult to convince people and their contacts in their social network to use the C2S system.

Table 8.1: Preference of Rating Schemes by participants of Survey

Age Group	Preference of Schemes	Number of Participants
15-24	Scheme 1	1
	Scheme 2	0
	Scheme 3	1
	Scheme 4	4
35-44	Scheme 1	0
	Scheme 2	0
	Scheme 3	0
	Scheme 4	2
45-54	Scheme 1	6
	Scheme 2	0
	Scheme 3	1
	Scheme 4	0

To address these problems in carrying out tests, a hypothetical data set of six persons was created, and 5 smart phones available in the lab had C2S clients installed. Photo sharing is enabled by transferring image to C2S server, after reading meta data of the picture and using queries as shown in Section 7.3 recipients were identified, and delivery was verified by receiving notification at client and availability of picture in the the Gallery application of the user's smartphone. The evaluation by user survey also shows that 75% of the participants had applications installed on the phone related to affinity and 80% of the participants were willing to send pictures to people above rating of 0.5 (see Annex C). On the basis of this rating threshold and affinity, we confirm that we can realize a social proximity based photo sharing service.

Chapter 9

Conclusion

This thesis project demonstrated the feasibility of social proximity based applications. The prototype of the application developed in this thesis allows users to share pictures via their smartphone. User's social proximity to his/her contacts could be derived from interaction data already present in the user's smartphone. The evaluation shows that the prototype C2S client achieves 80% accuracy when determining social proximity when using Scheme 1. Furthermore, it potentially allows other applications within the user's smartphone to use this proximity information (subject to the user's permission to share this information with a specific application). Also, transmission of the user's private data (call logs and message logs) is avoided by calculating social proximity *within* each user's smartphone. The user has full control of which data sources are accessed, including which sensors are used. Each user decides when he/she wants to perform a data transfer. This allows user to prioritize the data transfer among their other applications, thus improving usability. The fact that individuals are likely to carry their smartphone most of the time, and the fact that these smartphones have become a hub for many other devices (such as health monitoring systems and smart watches) makes the choice of the user's smartphone as the platform for applications such as SPI a good choice. Potentially the social proximity indicator could be used to automatically alert relevant persons in close social proximity (rating 0.8 and above) in case of emergencies.

We showed social graph construction technique based on a user's social interactions. Furthermore, the social proximity data is accessed outside of OSNs. We presented a smartphone based method of learning a user's preferences by characterizing various affinity indicators. A social proximity rating algorithm was validated with minimum assumptions.

The design decisions made in this project led to an adaptable and scalable solution. Using appropriate tools for Android development significantly reduces battery power consumption. Using Neo4J [102], MongoDB, and REST API

[103] enables scalability, and the complete system could be moved to the cloud [104] when desired. C2S ensures that the users receive pictures related to their affinity and only from people in their social proximity. Thus images are not shared by, or received from, unknown people. The literature review showed the sensitivity of users to exposure of their private data which led to the failure of several applications in the past. As described in Section 8.2 the rating of persons who are in a user's close social proximity are relatively insensitive to changes in parameters of the ranking algorithm when using Scheme 1.

Chapter 10

Future work

Possible future work includes:

- In this thesis project, we evaluated the use of call logs and message logs based on the number of units of conversation (i.e., one voice call (irrespective of duration) is treated as one unit of interaction. Similarly, one text message (irrespective of length of text) is treated as one unit of conversation). SPI could be extended by incorporating the duration of voice calls and length of text messages when calculating the interaction scores. It would be useful to evaluate if this increases the accuracy of the system. Also the origin of voice calls/ text messages can give some information about a user's social proximity with respect to the other person. During the study, we received feedback that regularity of interaction on a daily basis should also be considered when calculating the interaction score.

- Third party applications (such as Whatsapp, Viber, Skype, Wechat, etc.), other modes of interaction (such as e-mails, MMS, etc.) could be integrated with SPI when calculating the interaction score of contacts, thus leading to more consistent ratings of a user's contacts.

- SPI can also be extended to perform contacts classification. The origin of the call/message, the time when a call or message is made/received, could indicate if the contact involved in the interaction belongs to a friend or work place colleague. Based on this information an ontology can be applied to classify each of the contacts. Once contacts are classified, the system could propose to the user a group of contacts based on the meta data of an image. This could improve the application's usability.

- While sharing an image, the recipient can also be sent the context of the image to explain why he/she received this image, leading to a better user experience. For example, if an user shares the image of cricket game to his friends, and if C2S identifies that the user and his contact previously played cricket together, then the image along with a text message stating that "X sent you this image as you played cricket together" could enrich the user's experience. Also, the user could be asked to give feedback (whether the user likes this image or not) when they received an image. A machine learning algorithm could be incorporated which would learn from the collected feedback. This algorithm could adapt the SPIs to be more appropriate for these users.

- Interaction via voice calls and messages is demonstrated in the prototype implementation developed for this thesis project. Further, interactions based on co-location (physical proximity) could be explored and included in SPI computation.

- We utilized fixed weights of voice calls/text messages and obtained 80% accuracy. It would be interesting to evaluate the model when assigning different weights to each mode of interaction according to each user's specific usage pattern (i.e. if a person uses more messages than voice calls, then the weight for text messages might take preference over voice calls).

- User preferences could also be learned by integrating a user's OSN accounts if access to them is available. Also, reverse geo-coding can be used to characterize a user's frequently visited places perhaps hinting at the user's affinity. For example, if a user regularly goes to watch cricket games with specific teams on the field, this could hint that this user has a specific affinity towards this team and the sport. A richer profile of the user's preferences could enable the SPI rating to be used for recommendation engines or other media sharing purposes.

Chapter 11

Required reflections

The study of different user's modes of interaction and various associated patterns via their smartphones provides evidence for developers/researchers to better estimate and understand the user's social network and related usability requirements. This thesis project aims to improve the user's social network by providing a service that reflects this user's own social network. Additionally, this approach provides services to the users even those who do not use OSN, which is a desirable **social** effect of this master's thesis project.

With regards to **ethical** aspects, specifically in terms of privacy, we did a maximum amount of computation on the client, thus minimizing the transfer of the user's private data to the C2S server. In a real life implementation, the data stored in server would be encrypted and stored/processed as per the respective country's regulations. Privacy statements by the vendor should be designed to protect each user's personal data. However, there is always a trade-off when giving up some information in return for an improved user experience [105, 106]. This is a decision which each user makes by himself/herself. With respect to the proposed system he/she can opt in/ opt out any time. Considering the limitations in the work, the proposed architecture laid a good foundation for the proposed future work towards a more trustworthy solution.

With regards to **economic** aspects, the users can download the current application free of cost. A variety of business models [107] (such as one-time purchase, free trial, freemium, subscription, and in-app purchase (of high value software features or related products and services).) could be used by the vendors who provide support for such services. However, before this application has a major economic impact there is a need to provide a well defined API via which third party applications could interact with this application both to provide additional data and to utilize the SPI information (subject to the user's permitting this information to be shared). Given such an API there are quite a variety of applications that could use this information to improve their user's experience perhaps leading to better coupling of advertisements, increased ticket sales at events, greater participation in group events, etc.

Bibliography

[1] International Telecommunication Union (ITU) statistics. http://www.itu.int/net/pressoffice/press_releases/2013/05.aspx, February 2013. Accessed: 25-June-2014.

[2] Huiqi Zhang and Ram Dantu. Discovery of social groups using call detail records. In *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, volume 5333 of *Lecture Notes in Computer Science*, pages 489–498. Springer, Heidelberg, Berlin, 2008.

[3] Nathan Eagle, Alex (Sandy) Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, september 2009.

[4] Aleksandar Matic, Venet Osmani, and Oscar Mayora-Ibarra. Analysis of social interactions through mobile phones. *Mobile Networks and Applications*, 17(6):808–819, December 2012.

[5] Vindu Goel. Where Are You? Show 'Em With Google Latitude. http://bits.blogs.nytimes.com/2009/02/04/where-are-you-show-em-with-google-latitude/, February 2009. Accessed: 24-June-2014.

[6] Loopt. http://en.wikipedia.org/wiki/Loopt. Accessed: 09-Feb-2014.

[7] Dodgeball. http://en.wikipedia.org/wiki/Dodgeball_(service). Accessed: 09-Feb-2014.

[8] Color labs. http://colorlabsproject.com/, January 2014. Accessed:24-June-2014.

[9] Foursquare. https://foursquare.com/. Accessed: 15-Jan-2014.

[10] Wechat - the new way to connect. http://www.wechat.com/en/. Accessed: 15-Jan-2014.

[11] Guang Yang, Zhigang Liu, Karim Seada, Hawk-Yin Pang, August Joki, Jun Yang, Daniela Karin Rosner, Manish Anand, and Péter Pál Boda. Social proximity networks on cruise ships. In *proceedings of the 2008 Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 105–114, 2008.

[12] Highlight. http://highlig.ht/. Accessed: 15-Jan-2014.

[13] Janice Y. Tsai, Patrick Kelley, Paul Drielsma, Lorrie Faith Cranor, Jason Hong, and Norman Sadeh. Who's viewed you?: The impact of feedback in a mobile location-sharing application. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2003–2012, New York, NY, USA, 2009. ACM.

[14] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, page 337–350, New York, NY, USA, 2008. ACM.

[15] Eran Toch, Justin Cranshaw, Paul Hankes-Drielsma, Jay Springfield, Patrick Gage Kelley, Lorrie Cranor, Jason Hong, and Norman Sadeh. Locaccino: A privacy-centric location sharing application. In *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Adjunct*, Ubicomp '10 Adjunct, pages 381–382. ACM, New York, NY, USA, 2010.

[16] Sanem Kabadayi, Adam Pridgen, and Christine Julien. Virtual sensors: Abstracting data from physical sensors. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, WOWMOM '06, pages 587–592, Washington, DC, USA, 2006. IEEE Computer Society.

[17] Tanzeem Choudhury and Alex Pentland. The Sociometer: A Wearable Device for Understanding Human Networks. In *Computer Supported Cooperative Work - Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments*, November 2002.

[18] Anthony LaMarca and Eyal de Lara. *Location Systems: An Introduction to the Technology Behind Location Awareness*. Synthesis Lectures on Mobile and Pervasive Computing, pages 1–72. Morgan & Claypool Publishers, 2008.

[19] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *Proceedings of INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 775–784. IEEE, 2000.

[20] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 70–84. ACM, New York, NY, USA, 2004.

[21] Ekahau: Wi-Fi based real-time tracking and site survey solutions. `http://www.ekahau.com/real-time-location-system/solutions`. Accessed: 13-Jan-2014.

[22] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill Schilit. Place lab: Device positioning using radio beacons in the wild. In *Proceedings of the Third International Conference on Pervasive Computing*, PERVASIVE'05, pages 116–133. Springer, Heidelberg, Berlin, 2005.

[23] Kari Laasonen, Mika Raento, and Hannu Toivonen. Adaptive on-device location recognition. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 287–304. Springer, Heidelberg, Berlin, 2004.

[24] H. Laitinen, J. Lahteenmaki, and T. Nordstrom. Database correlation method for gsm location. In *53rd Vehicular Technology Conference*, volume 4, pages 2504–2508. IEEE, May 2001.

[25] Gaetano Borriello, Alan Liu, Tony Offer, Christopher Palistrant, and Richard Sharp. Walrus: Wireless acoustic location with room-level resolution using ultrasound. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 191–203. ACM, New York, NY, USA, 2005.

[26] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 32–43. ACM, New York, NY, USA, 2000.

[27] K. Chawla, G. Robins, and Liuyi Zhang. Object localization using rfid. In *International Symposium on Wireless Pervasive Computing (ISWPC)*, pages 301–306. IEEE, May 2010.

[28] Ubisense real-time location intelligence - ubisense. http://www.ubisense.net/en/. Accessed: 13-Jan-2014.

[29] Raffaele Bruno and Franca Delmastro. Design and analysis of a bluetooth-based indoor localization system. In *Personal Wireless Communications*, volume 2775 of *Lecture Notes in Computer Science*, pages 711–725. Springer, Heidelberg, Berlin, 2003.

[30] Fredrik Christiansson. A distributed, mobile positioning system for wireless, handheld devices. Master's thesis, KTH Royal Institute of Technology, Teleinformatics, Available from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-93545, June 2000.

[31] Anqi Luo and Lei Ge. Indoor location detection using WLAN. Master's thesis, KTH Royal Institute of Technology, Communication Systems, CoS, TRITA-ICT-EX-2009:209, Available from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-91200, January 2010.

[32] Mark T. Smith and Gerald Q. Maguire Jr. Receiver and correlator used to determine position of wireless device. US Patent 6975618, filed 26 June 2001, patent issued 13 December 2005.

[33] Pantopic. http://www.pantopic.com/what.htm. Accessed: 20-Feb-2014.

[34] M. Atzmueller, M. Becker, S. Doerfel, A. Hotho, M. Kibanov, B. Macek, F. Mitzlaff, J. Mueller, C. Scholz, and G. Stumme. Ubicon: Observing physical and social activities. In *proceedings of 2012 IEEE International Conference on Green Computing and Communications (GreenCom)*, pages 317–324, Nov 2012.

[35] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *Proceedings of Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2 of *INFOCOM '04*, pages 1012–1022 vol.2, March 2004.

[36] Nattapong Swangmuang and Prashant Krishnamurthy. An effective location fingerprint model for wireless indoor localization. volume 4 Issue

6, pages 836–850. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, December 2008.

[37] Ling Pei, Robert Guinness, Ruizhi Chen, Jingbin Liu, Heidi Kuusniemi, Yuwei Chen, Liang Chen, and Jyrki Kaistinen. Human behavior cognition using smartphone. *Sensors*, 13(2):1402–1424, January 2013.

[38] D. Yu S. Hwang. GPS localization improvement of smartphones using built-in sensors. *International Journal of Smart Home*, 06(03):1–8, July 2012.

[39] Lars Erik Holmquist, Jennica Falk, and Joakim Wigström. Supporting group collaboration with interpersonal awareness devices. In *Personal Technologies*, volume 3, pages 13–21. Springer, 1999.

[40] John Krumm and Ken Hinckley. The nearme wireless proximity server. In *Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 283–300. Springer, Heidelberg, Berlin, 2004.

[41] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: A high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 1–14. ACM, New York, NY, USA, 2007.

[42] Bluehoo. http://bluehoo.softonic.com/symbian. Accessed: 13-Jan-2014.

[43] Trinh Minh Tri Do and Daniel Gatica-Perez. Human interaction discovery in smartphone proximity networks. In *Personal and Ubiquitous Computing*, volume 17(3), pages 413–431. Springer, March 2013.

[44] Venet Osmani, Iacopo Carreras, Aleksandar Matic, and Piret Saar. An analysis of distance estimation to detect proximity in social interactions. In *Journal of Ambient Intelligence and Humanized Computing*, volume 5, pages 297–306. Springer, Berlin Heidelberg, June 2014.

[45] Danny Wyatt, Tanzeem Choudhury, Jeff Bilmes, and James A. Kitts. Inferring colocation and conversation networks from privacy-sensitive audio with implications for computational social science. *ACM Trans. Intell. Syst. Technol.*, 2(1):7:1–7:41, January 2011.

[46] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. Bridging the gap between physical location and online social

networks. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, pages 119–128. ACM, New York, NY, USA, 2010.

[47] Jason Wiese, Patrick Gage Kelley, Lorrie Faith Cranor, Laura Dabbish, Jason I. Hong, and John Zimmerman. Are you close with me? are you nearby?: Investigating social groups, closeness, and willingness to share. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, pages 197–206. ACM, New York, NY, USA, 2011.

[48] Simon Jones and Eamonn O'Neill. Feasibility of structural network clustering for group-based privacy control in social networks. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, pages 9:1–9:13. ACM, New York, NY, USA, 2010.

[49] George Danezis. Inferring privacy policies for social networking services. In *Proceedings of the 2Nd ACM Workshop on Security and Artificial Intelligence*, AISec '09, pages 5–10. ACM, New York, NY, USA, 2009.

[50] Michael Terry, Elizabeth D. Mynatt, Kathy Ryall, and Darren Leigh. Social net: Using patterns of physical proximity over time to infer shared interests. In *Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 816–817. ACM, New York, NY, USA, 2002.

[51] G Kossinets and D Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[52] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt. Scale-free topology of e-mail networks. *Physical Review E*, 66, 2002.

[53] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 171–180. ACM, New York, NY, USA, 2000.

[54] Nathan Eagle, Alex(Sandy) Pentland, and David Lazer. Mobile phone data for inferring social network structure. In Huan Liu, John J. Salerno, and Michael J. Young, editors, *Social Computing, Behavioral Modeling, and Prediction*, pages 79–88. Springer, 2008.

[55] Christopher McCarty. Structure in personal networks. *Journal of Social Structure*, 3, 2002.

[56] Michel Shahan. Personalized TV with Recommendations: Integrating social networks. Master's thesis, KTH Royal Institute of Technology, Communication Systems, CoS, COS/CCS 2008-25, Available from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-91670, October 2008.

[57] Jesus Paredes Sanz. Exploiting social networking technologies in order to enhance internal communication within and across a large enterprise: Improving a corporate social network. Master's thesis, KTH Royal Institute of Technology, Communication Systems, CoS, TRITA-ICT-EX-2013:58, Available from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-120598, April 2013.

[58] Erik Hedin. Social networks and mobile devices : The next generation mobile phonebook with social networking widgets. Master's thesis, KTH Royal Institute of Technology, Communication Systems, CoS, COS/CCS 2008-18, Available from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-91688, September 2008.

[59] Freemotion library. http://www.sensorplatforms.com/freemotion-library/. Accessed: 14-Jan-2014.

[60] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *Comm. Mag.*, 48(9):140–150, September 2010.

[61] H. Khan and U. Aziz. Proximity-based social networking. Patent US 20110238755 A1 filed 24 March 2011, patent issued 29 September 2011.

[62] Scott Lederer, Jennifer Mankoff, and Anind K. Dey. Who wants to know what when? privacy preference determinants in ubiquitous computing. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 724–725. ACM, New York, NY, USA, 2003.

[63] Sunny Consolvo, Ian E. Smith, Tara Matthews, Anthony LaMarca, Jason Tabert, and Pauline Powledge. Location disclosure to social relations: Why, when, & what people want to share. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 81–90, New York, NY, USA, 2005. ACM.

[64] Natastha Lomas. Stop trying to make proximity-based social networking happen. http://techcrunch.com/2013/07/23/location-vs-communication/, July 2013. Accessed:13-June-2014.

[65] Melissa Jun Rowley. Proximity-based social change. http://newsroom.cisco.com/feature/1204130/Proximity-Based-Social-Change, June 2013. Accessed:13-June-2014.

[66] Nmawston. Android Captured 79% Share of Global Smartphone shipments in 2013. http://blogs.strategyanalytics.com/WSS/post/2014/01/29/Android-Captured-79-Share-of-Global-Smartphone-Shipments-in-2013.aspx, January 2014. Accessed: 28-Feb-2014.

[67] Shane Schick. 71% of developers pick android as their top platform. http://www.fiercedeveloper.com/story/71-developers-pick-android-their-top-platform/2013-07-29, July 2013. Accessed: 28-Feb-2014.

[68] João Prudêncio, Ana Aguiar, and Daniel Lucani. Physical activity recognition from smartphone embedded sensors. In *Pattern Recognition and Image Analysis*, volume 7887 of *Lecture Notes in Computer Science*, pages 863–872. Springer, Heidelberg, Berlin, 2013.

[69] Location API. http://developer.android.com/google/play-services/location.html. Accessed: 13-June-2014.

[70] MongoDB : An open-source NoSQL database. http://www.mongodb.org/. Accessed: 13-June-2014.

[71] Neo4J : A graph database. http://www.neo4j.org/. Accessed: 13-June-2014.

[72] Stacy E Thayer and Sukanya Ray. Online communication preferences across age, gender, and duration of internet use. volume 9, issue 4, pages 432–440. National Library of Medicine, US, 2006.

[73] Santi Phithakkitnukoon and Ram Dantu. Inferring social groups using call logs. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, volume 5333 of *Lecture Notes in Computer Science*, pages 200–210. Springer, Heidelberg, Berlin, 2008.

[74] Jun-Ki Min, Jason Wiese, Jason I. Hong, and John Zimmerman. Mining smartphone data to classify life-facets of social relationships. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 285–294. ACM, New York, NY, USA, 2013.

[75] Alex Wright. Get smart. *Communications of the ACM - Rural engineering development*, 52(1):15–16, January 2009.

[76] Cliff A.C. Lampe, Nicole Ellison, and Charles Steinfield. A familiar face(book): Profile elements as signals in an online social network. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 435–444. ACM, New York, NY, USA, 2007.

[77] Brett Adams, Dinh Phung, and Svetha Venkatesh. Sensing and using social context. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 5(2):11:1–11:27, November 2008.

[78] Risto Sarvas, Mikko Viikari, Juha Pesonen, and Hanno Nevanlinna. Mobshare: Controlled and immediate sharing of mobile images. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 724–731. ACM, New York, NY, USA, 2004.

[79] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 199–208. ACM, New York, NY, USA, 2012.

[80] Reto Grob, Michael Kuhn, Roger Wattenhofer, and Martin Wirz. Cluestr: Mobile social networking for enhanced group communication. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work*, GROUP '09, pages 81–90. ACM, New York, NY, USA, 2009.

[81] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 44–54. ACM, New York, NY, USA, 2006.

[82] J-P Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336, 2007.

[83] Patrick Gage Kelley, Robin Brewer, Yael Mayer, LorrieFaith Cranor, and Norman Sadeh. An investigation into facebook friend grouping. In *Human-Computer Interaction - INTERACT 2011*, volume 6948 of *Lecture Notes in Computer Science*, pages 216–233. Springer, Heidelberg, Berlin, 2011.

[84] M.I. Akbas, R.N. Avula, M.A. Bassiouni, and D. Turgut. Social network generation and friend ranking based on mobile phone data. In *Communications (ICC), 2013 IEEE International Conference on*, pages 1444–1448, June 2013.

[85] Bradley M. Okdie, Rosanna E. Guadagno, Frank J. Bernieri, Andrew L. Geers, and Amber R. Mclarney-Vesotski. Getting to know you: Face-to-face versus online interactions. In *Computers in Human Behavior, in Third International Cognitive Load Theory Conference*, volume 27, pages 153 – 159. Elsevier Limited, January 2011.

[86] Eliane M. Boucher, Jeffrey T. Hancock, and Philip J. Dunham. Interpersonal sensitivity in computer-mediated and face-to-face conversations. *Media Psychology*, 11(2):235–258, 2008.

[87] Robin Dunbar. *How Many Friends Does One Person Need?: Dunbar's Number and Other Evolutionary Quirks*. Faber and Faber, February 2010.

[88] Wesley N. Colley. Colley's Bias Free College Football Ranking Method: The Colley Matrix Explained. http://www.colleyrankings.com/matrate.pdf. Retrieved: 20-May-2014.

[89] Kenneth Massey. Statistical models applied to the rating of sports teams. Master's thesis, Bluefield College, 1997.

[90] Timothy P. Chartier, Erich Kreutzer, Amy N. Langville, and Kathryn E. Pedings. Sensitivity and stability of ranking vectors. *SIAM Journal on Scientific Computing*, 33(3):1077–1102, May 2011.

[91] David Easley and Jon Kleinberg. Networks, crowds, and markets. *Cambridge University Press*, 6(1):6–1, 2010.

[92] Erik Hatcher, Otis Gospodnetic, and Michael McCandless. *Lucene in action*. Manning Publications, 2 edition, July 2010.

[93] Suranga Seneviratne, Aruna Seneviratne, and Prasant Mohapatra. Your apps are what you are: Personification through installed smartphone apps. NICTA, Australia, November 2013.

[94] Transferring data using sync adapters. http://developer.android.com/training/sync-adapters/index.html. Accessed: 13-June-2014.

[95] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Internet Measurement Conference*, IMC '09, pages 280–293. ACM, New York, NY, USA, 2009.

[96] Android SQLite database. http://developer.android.com/reference/android/database/sqlite/package-summary.html. Accessed: 13-June-2014.

[97] Aaron Carroll and Gernot Heiser. An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX Conference*, USENIXATC'10, pages 21–21. USENIX Association, Berkeley, CA, USA, 2010.

[98] Flickr. https://www.flickr.com/. Accessed: 15-June-2014.

[99] Instagram. http://instagram.com/. Accessed: 15-June-2014.

[100] Metadata-extractor - A Java library for reading metadata from image files. http://www.clarifai.com/. Accessed: 15-June-2014.

[101] Clarifai - advanced image recognition. http://www.clarifai.com/. Accessed: 15-June-2014.

[102] David Montag. Understanding Neo4J scalability. Technical report, Neotechnology, 01 2013. http://info.neotechnology.com/rs/neotechnology/images/Understanding%20Neo4j%20Scalability(2).pdf Retrieved: 24-June-2014.

[103] Megree chooses Neo4j for its scalability, REST API, high availability - Use Case. http://www.neotechnology.com/megree-usecase/. Accessed: 22-June-2014.

[104] Miles Ward. NoSQL Database in the Cloud: MongoDB on AWS. Technical report, 03 2013. https://media.amazonwebservices.com/AWS_NoSQL_MongoDB.pdf Retrieved: 24-June-2014.

[105] When privacy and enhanced user experience collide online. http://socialmediatoday.com/kenmueller/1153081/when-privacy-and-enhanced-user-experience-collide-online. Accessed: 23-June-2014.

[106] Hyowon Lee, Nazlena Mohamad Ali, and Cathal Gurrin. Reflection on reflection: Daily review of lifelog photos and the usability of wearable digital camera. In *HCI International 2013 - Posters' Extended Abstracts*, volume 374 of *Communications in Computer and Information Science*, pages 368–372. Springer, Heidelberg, Berlin, 2013.

[107] On selecting an optimal mobile application business model. Technical report, Arrayent, 01 2012. http://www.arrayent.com/pdfs/TheOptimalMobileAppBusinessModels.Arrayent.2012-01-26.pdf Retrieved: 24-June-2014.

Appendices

Appendix A

Use case scenarios

The scenario relies on an event representing a activity carried out by user along with set of other people. An example of such activity can be fishing. Any user/actor clicks some pictures using his smartphone. While clicking picture, smartphone sensors will be used to collect location information where the activity has taken place. The proposed application will enable user who clicked the picture to share with his friends who were present during the event. We will name the users as Alice, Bob and Bill. The given scenarios present different use cases.

A.1 Scenario 1

Alice wants to go for fishing with Bob and Bill and creates an event in calendar.

A.1.1 Case 1

Alice creates an event in his calendar and sends out invite to Bob and Bill to join him for fishing on weekend. Each of them goes for the fishing and clicks pictures using their pre-owned smartphones and decides to share among each other.
Assumption: Alice creates an event via calendar application in his smartphone and sends invite through emails. The event contains the list of attendees list and location.
Expected outcome: After each of them who participated in activity decides to share the pictures using C2S, all of the three users are expected to receive the images.
Dependency: Calendar invitation acceptance by user.

A.1.2 Case 2

Alice creates an event in his calendar and sends out invite to Bob and Bill to join him for fishing on weekend. Only two of them goes for the fishing and clicks pictures using their pre-owned smartphones and decides to also share with the third person who could not make it for that day.
Assumption: Alice creates an event using via claendar application in his smartphone and sends invite through e-mails. The event contains the list of attendees list and location. Two persons accepted the invite and third refused.
Expected outcome: After each of those two (who carried out the activity as per the calendar) decides to share the pictures using C2S, all of three are expected to receive the images. The person who was not present at the activity should also be able to receive images automatically.
Dependency: Calendar invitation acceptance/denial by user.

A.2 Scenario 2

Alice wants to go for fishing with Bob and Bill and creates an event in calendar. But there are other friends of Alice, not present in calendar event who goes for fishing but not necessarily on same day.

A.2.1 Case 1

Alice creates an event in his calendar and sends out invite to Bob and Bill to join him for fishing on weekend. Each of them goes for the fishing and clicks pictures using their pre-owned smartphones and decides to share among each other and with their friends, who also likes fishing but goes on some other day.
Assumption: Other friends of Alice, who likes fishing had already expressed their affinity through some medium. May be its through C2S application or any other Online Social Networks.
Expected outcome: Automatic sharing of pictures between Alice and his friends who were not together doing fishing on same day should take place. Pictures from Bob and Bill can be shared with Alice but not with Alice's friends.
Dependency: A graph with friends should be maintained and then filtered out based on interests/likings/activities. Dependency on google calendar also exist.

A.3 Scenario 3

Alice goes for fishing alone and happens to see some of his friends doing same at the venue

A.3.1 Case 1

Alice goes for fishing alone and clicks some pictures along with his friends whom he met coincidentally and he wants to share pictures with his friends who were present at the event and vice versa.
Assumption: Some indicator of friendship exists. Indicator can be android contact list or OSN friend list. Geolocation is being fed to the C2S server. Contextual information is present in the pictures.

Expected outcome: Automatic sharing of pictures between Alice and his friends who were present at the event.
Remarks: In former case 1 of scenario 2, it is social proximity enabling decision and in this scenario , it is location with social proximity which takes decision.

A.3.2 Case 2

Alice goes for fishing alone and clicks some pictures. Coincidentally, his friends of friends are also present and he wants to share his pictures with them and vice versa.
Assumption: Some indicator of friends of friends exists. Indicator can be android contact list or OSN friend list. Geolocation is being fed to the C2S server. Contextual information is present in the pictures.
Expected outcome: Automatic sharing of pictures between Alice and his friends of friends who were present at the event.
Remarks: In former case 1 of scenario 2, it is social proximity enabling decision and in this scenario , its location with social proximity which takes decision.
Challenges: Contact list synchronization to detect friends of friends.

Appendix B

Experiment and Survey Questionnaire

B.1 Introduction

As part of Mobile interacting devices division of media computing lab at Technicolor R&D, we are doing research on the use of various smartphone data sources as an indicator of social proximity. We have developed an application which exploits the communication logs present in your smartphone to calculate an index of frequently contacted persons. We invite volunteers to install this application and give feedback. We plan to use this social proximity index for the photo sharing application.

B.2 User Agreement for participation in Experiment

By participating in this experiment, you agree to install an android application and provide feedback based on the questions asked. This application does not require internet connection and hence we assure you that your data is not shared with any one. You agree to give access to your contacts and SMS logs to the android application for the duration of the test . so that we can test our contact rating algorithm.

B.3 Survey Questions

B.3.1 About you

1. What is your age group ?

15-24	
25-34	
35-44	
45-54	
55-64	
65+	

2. How often do you delete smartphone call logs and message logs?

Never	
Yearly	
Half-Yearly	
Monthly	
Weekly	

3. Which mode of interaction do you prefer with people when using your smartphone?

Voice Calls	
SMS	
Third party messaging applications (e.g. WhatsApp, Viber, etc.)	
Third party Voice over Internet Protocol (VoIP) applications (e.g. Skype)	

4. Do you use smartphone applications related to your interests (e.g. sports scores, fitness, ...) ?
    - If Yes: can you estimate the number of such installed apps ?

5. Do you use a smartphone calendar to plan the activities of your interests?
    - If Yes: do you regularly fill in the address for the event?
    - If Yes: do you regularly fill in the contacts for the event?

6. Do you use more voice calls or texting ?
    - Can you provide a rough ratio of voice calls to text messaging?

B.3.2 About the C2S rating

The test application has four buttons enabling you to run four versions of our algorithm calculating an index of frequency of interaction for your contacts.

7. Which of the 4 algorithms gives the best ranking of the people you interact most with ?

6M/Week	
6M/Month	
3M/Week	
3M/Month	

8. Do you think that such a rating is a good indicator of social closeness ?
9. Can you identify any obvious anomalies in the ratings (e.g. contacts too high or low in the list) ?
    - If Yes: is there an obvious explanation (e.g. a very close friend with whom you only interact via Skype, ...) ?

B.3.3 About photo sharing

10. To whom would you like to send your pictures of any sport activity?
11. To whom would you like to send your pictures of any party/any family event which you attended?
12. To whom would you like to share your selfie captured at event above?

Appendix C

Survey Results

This chapter gives results obtained by survey (see Appendix B) conducted. The survey included 16 participants.

C.1 Participant Details

The Table C.1 presents the age groups of the survey participants.

Table C.1: Number of participants categorized by age groups

Age group	**Number of Participants**
15-24	6
25-34	0
35-44	3
45-54	7

Participants were asked about how often they delete their voice calls and messages related log data. The results are tabulated in Table C.2.

Table C.2: Frequency of voice calls and message logs deletion

Frequency of log deletion	**Number of Participants**
Never	11
Yearly	2
Half-yearly	1
Montly	1
Weekly	1

Participants were asked about their most preferred mode of interaction via smartphone. Responses are tabulated in Table C.3.

Table C.3: Participant's most preferred mode of Interaction

Mode of Interaction	Number of Participants
Voice calls	-
Text Messages (SMS)	8
Third Party messaging applications (e.g. Whatsapp, Viber etc.)	2
Third party VOIP applications (e.g. Skype)	-
Voice Calls and SMS	5
Voice Calls and Third party Messaging applications	1

When asked about application usage related to affinity, 14 of 16 participants indicated that they use applications in their smartphone related to their affinities and they also gave feedback that SPI is a good indicator of social closeness. Additionally, 13 of 16 participants indicated that they use calendar to plan activities of their interest, but only 5 of those 13 include their contact details and address of the event in the calendar. When participants were questioned about anomalies in the ratings of their contacts, 50% (8 Participants) indicated that their one or more contacts (mostly family members) are low rated. They pointed out that interaction with their family members happen via third party VOIP applications or through e-mail.

Table C.4 shows the ratio of voice calls to text messaging of the survey participants.

The survey questionnaire also collected feedback about the ratings of contacts to whom a user would like to send pictures as categorized below.

- Picture of any sport activity (Refer Table C.5)
- Picture of any party or family event (Refer Table C.6)
- Personal picture or Selfie (Refer Table C.7)

Table C.4: Ratio of voice calls to text messages of participants

Ratio of Voice calls to text messages	Number of participants
1:1	2
1:2	2
1:3	1
1:5	2
1:10	2
1:100	3
2:1	1
3:2	1
5:1	1
10:1	1

Table C.5: Rating threshold for sharing pictures of any sport activity

Rating	Number of participants
Above 0.84	1
Above 0.6	9
Above 0.4	4
Above 0.2	1
All contacts	1

Table C.6: Rating threshold for sharing pictures of family event

Rating	Number of participants
Above 0.9	2
Above 0.8	8
Above 0.87	1
Above 0.79	1
Above 0.7	1
Would prefer manually to choose contacts	3

Table C.7: Rating threshold for sharing personal images or Selfie

Rating	Number of participants
Above 0.8	1
Above 0.87	1
Above 0.83	10
Above 0.7	2
Do not prefer selfies	2

TRITA-ICT-EX-2014:66