Networks and Distributed Systems:

Trust Management in Online Social Networks

by

Cian Malone, B.Sc.

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

September 2008

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Cian Malone

August 28, 2008

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Cian Malone

August 28, 2008

Acknowledgments

First of all, I would like to thank my supervisor, Declan O'Sullivan, for providing guidance, support, and valuable suggestions throughout the year and ensuring that I successfully completed my research by helping me to work at a steady pace.

I would also like to thank the members of this year's NDS course, especially those who provided feedback regarding their experiences with trust, privacy, and Facebook. Thanks also go to all those whose survey responses helped to shape my research.

Finally, none of this would have been possible without the support of my family. I want to thank my Mum and Dad, Diane and Paddy, and my brother Lorcan for everything they have done for me.

Cian Malone

University of Dublin, Trinity College

September 2008

Networks and Distributed Systems:

Trust Management in Online Social Networks

Cian Malone, M.Sc.

University of Dublin, Trinity College, 2008

Supervisor: Declan O'Sullivan

With the recent growth of online social networks such as Facebook, privacy concerns have become more and more serious. Users are often not made aware of the visibility and permanence of the data they place online, and are often discouraged from taking a more proactive approach to protecting their privacy due to complex privacy controls. This thesis designed an application to evaluate the integration of a multi-faceted, specialisable, personalisable trust model with Facebook. We found that users feel that trust is a necessary aspect when making privacy decisions on the network, and that a multi-faceted trust model allows diverse users to express their individual opinions about the important qualities of trust. In the future, better integration of trust and online social networks is important in order to maintain user privacy.

Contents

List of Figures

Chapter 1

Introduction

Nowadays hundreds of millions of people actively use Online Social Networks (OSNs) such as Facebook [1], MySpace [2], and Bebo [3] to create and maintain social relationships. As the size of these networks has increased, concerns about members' privacy have become more pressing. Although some sites offer sophisticated, fine-grained privacy control, studies have shown that a disparity exists between OSN users' stated attitudes towards privacy and their actual behaviours – namely, they state a high desire for privacy when in reality they rarely change or even look at the privacy settings of the network [7, 8]. The reasons for this dichotomy include a misunderstanding of the actual visibility of users' content, the fact that privacy controls are often considered overly-complex by users, and a perception that more rigorous privacy settings would cause offense to friends or reduce the value of the network.

Trust is a useful shortcut for distilling and analysing multiple facets of individuals' attitudes and behaviours in order to better make decisions about how to interact with those individuals.

In commerce, for instance, trust is a valuable tool when determining who to make deals with. Characteristics such as reputation, financial competence, and reliability are important factors of trust in a business situation. In a distributed computing system, individuals are AI agents, and they must determine which agents or services to interact

with in order to complete a process successfully, likely taking into account the accuracy of data supplied by other agents in order to avoid malicious or incorrect results, as well as reliability and other factors vital to progress. With regard to interpersonal relationships, people trust those they believe to be honest and reliable.

When applied to OSNs, trust could be a valuable factor in implementing intuitive fine-grained privacy controls to manage the visibility and searchability of users' data without requiring a large commitment on the part of network members to spend time learning and using them.

1.1 Motivation

myTrust [4], a multifaceted and personalisable trust management service developed in the Knowledge and Data Engineering Group (KDEG) [5] at Trinity College Dublin, allows participants to specify which factors of trust are the most important to them in order to capture their own subjective view of trust. Unlike many current trust models that rely on a single trust rating, myTrust breaks trust into eight constituent factors: belief, competence, confidence, credibility, faith, honesty, reliability, and reputation. The author, Quinn, posits that such a multifaceted approach is necessary due to the diverse meanings individuals attribute to the term trust. The model is specialisable, allowing for use across a wide array of contexts, depending on the trust factors that are most suited to a particular situation. In particular, Quinn presents a Web service selection mechanism, a policy-based access control mechanism, and a trusted Instant Messaging client to demonstrate the applicability of the model to various contexts (distributed agent-based systems, organisational policy systems, and interpersonal communications, respectively).

Quinn notes in his outline of further work that the trust model could be adapted to provide trust-based features to an Online Social Network. miniOSN [6] is a prototype OSN that incorporates the trust model to provide access control to blog posts and photographs that the network's users upload. Fu's project determined that OSN users feel a need for

trust semantics when making privacy decisions. However, the small scale and specialised nature of miniOSN meant that further research into the area was necessary.

In May 2007, Facebook, the world's second most popular OSN, released a development API to allow third parties to build applications that exploit the site's vast social graph. As a result, it became possible to build and evaluate an application that provided the features of miniOSN to an existing social network containing tens of millions of users in order to determine the real-world applicability of the myTrust model to online social networking.

1.2 Research Questions

This research attempted to answer several questions:

- Do Facebook users feel a need for trust-based privacy control?
- Does the myTrust model capture the trust semantics that people need on Facebook?
- Which factors of trust do Facebook users feel are the most important to the context of friend relationships on the site?
- What features must a trust-based privacy control application provide?
- Once those features have been implemented, do Facebook users feel that they would be useful and usable additions to the network?

1.3 Objectives

In order to successfully answer the research questions specified above, the following objectives were identified:

- Research the current state of the art in trust-based systems.

- Replicate the functionality of miniOSN as a Facebook application.
- Survey users about the usefulness of the application, as well as their opinions about which trust factors they most value.
- Based on those results, identify the Facebook-specific features that users want.
- Implement those features and survey users on the usefulness and usability of the final application.

1.4 Approach

To complete the objectives and answer the research questions posed, we first researched general trust-based systems, with a specific focus on myTrust and miniOSN. Next, we designed and developed a Facebook application in PHP to replicate the functionality of the miniOSN project. Following the completion of the application's development, we evaluated the application with a small-scale user study and a wider-scale survey to determine users' opinions about the application, and about trust on Facebook in general. We used the results of this evaluation to identify features to improve the existing application. A second iteration of the application was designed and developed, and we evaluated its usability and utility with another small user study and a survey.

1.5 Contribution

We evaluated the necessity of trust-based privacy control on Facebook, and provide the design and implementation details of an application that adds this feature to Facebook. This thesis includes the results of evaluating the application, as well as analyses of Facebook users' attitudes towards trust with regard to relationships on the site. Finally, we outline possible directions for future work in the area.

1.6 Thesis Outline

The remainder of the thesis is organised as follows:

- Chapter Two presents the current state of the art of Online Social Networks and trust-based systems. In particular, it focuses on existing OSNs that implement trust features to some extent and explores in detail the myTrust and miniOSN projects that preceded this work.
- Chapter Three outlines the design and implementation of a Facebook application that provides the trust mechanisms of miniOSN.
- Chapter Four presents the evaluation of the application outlined in Chapter Three.
- Chapter Five details the design and implementation of a second iteration of the application, with particular detail on the features identified as a result of the first iteration's evaluation.
- Chapter Six shows the results of the evaluation of the second iteration of the application.
- Finally, chapter Seven presents the results of the research, as well as avenues of further research in the area of trust management and Online Social Networks.

Chapter 2

State of the Art

2.1 Introduction

This chapter presents the current state of the art in Online Social Networks and trust systems. Section 2.2 presents an overview of OSNs as well as a discussion of Facebook and its approach to protecting user privacy. Section 2.3 presents an overview of trust and its diverse meanings. Sections 2.4 and 2.5 detail the current state of the art of trust systems and OSNs with integrated trust mechanisms. Finally, section 2.6 presents a summary of the chapter.

2.2 Online Social Networks

2.2.1 OSN Overview

Online Social Networks provide a mechanism for social interaction across the Internet, typically through the use of profiles containing users' personal information. By allowing people to search and find profiles with common interests, geographical locations, expert knowledge, or existing offline friendships, as well as providing methods of interacting with the owners of those profiles, OSNs facilitate the formation and maintenance of online social relationships. Depending on the nature of the network, these relationships may be

useful for entertainment purposes on a site like YouTube [11], finding employment using a network such as LinkedIn [12], or meeting new people and keeping up with old friends on MySpace or Facebook.

2.2.2 Facebook

Since its inception in 2004, Facebook has grown exponentially, currently boasting over 90 million active users. Facebook segregates users geographically by assigning each to a network corresponding to his or her University or location. As a result, Facebook is relatively unique among OSNs in that Facebook relationships tend to originate offline before being brought online for easier surveillance and maintenance [13]. Numerous studies have examined the impact of this feature on the attitudes and behaviours of Facebook users. Acquisti and Gross argue that this has led to an imagined sense of community and security when in fact the actual security and privacy protection measures are far more lax than users think [7, 8]. Perhaps as a result of this, studies such as [14] have revealed that while Facebook users often express that privacy is of high importance to them, their behaviour on the site contradicts their stated attitudes.

The size of Facebook's network, combined with the unique physical grounding of Facebook relationships, makes it an interesting point of study for trust-based privacy measures.

2.2.3 Facebook & Privacy

Facebook has a somewhat rocky history when it comes to protecting users' privacy. Although it currently boasts perhaps the most comprehensive, fine-grained privacy controls of any OSN, the site has been criticised for breach of user privacy several times in the past.

- The Newsfeed and Minifeed collect users' actions on Facebook and alert friends to status updates, photo uploads, and more. Although all of the information was

available before the feeds were added to the site, their introduction was met with outrage by a large contingent of Facebook users, claiming that the features encouraged stalking. Over time, however, the majority of users grew to view the feeds as a valuable feature of the site.

- Beacon, a social advertising scheme that pulls content from users' behaviour on third-party sites, was met with outrage when first unveiled. Originally an opt-out scheme, Beacon allowed actions such as purchases and reviews on external sites to be published to users' Newsfeeds. After backlash, Facebook revamped Beacon, making it opt-in.

- The most recent privacy breach was the result of a bug, rather than Facebook company policy. Currently, Facebook is in the process of rolling out a new user interface, and has a public beta available at http://www.new.facebook.com. In July 2008, Sophos Labs, an IT security firm, reported that users' birthdates, even if marked private, were currently displayed to the public on the beta interface [15]. Facebook quickly fixed the problem, but critics pointed to the bug as further evidence of Facebook's apathetic approach to privacy.

In spite of its past privacy breaches, Facebook offers sophisticated privacy controls that allow users to specify privacy settings for the various pieces of information on their profiles, making them visible only to friends, to friends of friends, or to all members of their networks, or by blacklisting individual people. Additionally, for school networks, users can restrict profile access to any subset of undergraduates, graduates, staff, and alumni. Users can adjust their searchability on the site to allow their friends, members of their networks, or the public to find them on the site. Also, users can control which of their actions on the site are eligible to be published to their friends' news feeds, which external sites can submit data to Facebook (i.e. Beacon settings), and whether or not their behaviours can be used to advertise to their friends.

2.2.4 Facebook Platform

In May 2007, Facebook released the Facebook Platform [16], a set of APIs to provide external developers with access to Facebook's social graph and communication channels. The platform includes a REST-based API, a SQL-style interface called FQL, and a markup language called FBML. It also provides a sandboxed area of the site, known as the Canvas, in which applications run, as well as the ability to add static content to users' profile pages and access viral uptake channels such as the Facebook News Feed, friend invitations, and email notifications.

Since its release, the Facebook Platform has grown to power more than 24,000 applications, with 95% of Facebook members using at least on third-party application.

2.3 Trust

2.3.1 Definition

Trust has diverse definitions depending on context. [17] provides an overview of the various meanings of the term trust when applied to the contexts of philosophy, sociology, psychology, management, marketing, ergonomics, HCI, and e-Commerce. Until relatively recently, the concept of trust when related to computer science research focused on providing trusted computing – for instance, encryption algorithms, robust agent evaluation and negotiation mechanisms, and access control systems. With the growth of OSNs, computer scientists have started to explore the social side of trust. For the purposes of this thesis, trust is defined as a metric based on the eight characteristics of the myTrust model (belief, competence, confidence, credibility, faith, honesty, reliability and reputation), that OSN members use to facilitate making privacy decisions with regard to other members of the network.

2.3.2 Properties of Trust

Transitivity

Depending on the context, trust may or may not be transitive. In cases where trust is deemed to transfer across links, it usually degrades as it flows from the source. For instance, if person B highly trusts person C, and person A highly trusts person B, it does not necessarily follow that A highly trusts C. Instead, the transitivity of A's trust depends on A's evaluation of B's ability to provide reliable recommendations. It is important to note that B's trustworthiness and the reliability of B's recommendations may not be the same.

Additionally, the transitivity of trust may work over multiple degrees of separation – for instance, if C trusted person D, then it may be possible to determine the amount that A should trust D. In many implementations, though, if trust can commute over multiple hops, it degrades as it reaches further from the source to prevent homogeneity of trust ratings.

Asymmetry

Trust is typically asymmetric, due to its personal nature. It is completely plausible, for instance, that for a given pair of people, A and B, A may highly trust B while B does not trust A at all, or vice versa.

Context-Specificity

Trust is context-specific. The qualities that convey trust when one looks for medical advice, for instance, probably centre on a doctor's medical competence, reputation and reliability. When trying to buy a car, however, medical knowledge is not particularly important, and a person is likely to seek dealers with strong competence, reputation, and reliability in the area of car sales. And while in general those three characteristics are important when determining whether or not to strike up a friendship with another

person, it is likely that factors such as honesty and believability are more important in that context.

As a result, no one definition of trust is sufficient for all contexts in which trust is useful.

Distrust

It is worth noting that distrust is not equivalent to a lack of trust. Marsh [18] outlines that while a lack of trust can result from (for example) two actors never having interacted with one another in the past, distrust implies that one actor has made an explicit judgment not to trust the other, perhaps as a result of previous bad experiences when interacting with that individual.

Many trust metrics have no concept of distrust, although two of the works mentioned in the following section, TidalTrust and Appleseed, incorporate the notion into their metrics. The myTrust model which serves as the basis for the research conducted in this thesis does not represent distrust.

Local and Global Trust

There are two main approaches to trust algorithms – local and global. Local algorithms are user-centric, whereas global algorithms are community-centric. A local algorithm relies on the ratings a given user provides to generate a picture of the rest of the networks trustworthiness for that user. Global algorithms, on the other hand, typically analyse all users' ratings in parallel, and take a network-flow based approach to generating an explicit trust value for each user within the network. Google's PageRank algorithm [19] is one such global trust algorithm, analysing links in parallel to determine the overall reliability of a given page with respect to search parameters.

Due to the user-centric nature of Facebook, the trust calculation algorithms developed as part of this thesis are local algorithms.

2.4 Trust Systems

2.4.1 EigenTrust

EigenTrust [20] is an algorithm that uses trust to mitigate the effects of malicious members of a peer-to-peer network offering inauthentic files to legitimate users. The algorithm takes a hybrid local-global approach to trust calculation, whereby a peer's global reputation is determined by its local ratings by all other peers weighted by those peers own global reputations. The use of collusion among malicious peers to subvert reputation systems has been cited in several cases [21, 22], but by requiring peers to place trust in a vector of seed peers, EigenTrust breaks up malicious collectives, limiting the effect of collusion on the rest of the network. To further protect against the behaviour of malicious peers, the EigenTrust algorithm requires majority vote among a group of peers conducting the same calculation. The algorithm has the added side effect of providing an incentive for legitimate users to share files on the network, because other peers will be more willing to trust and share with them.

2.4.2 TidalTrust

TidalTrust [23] is a trust system based on the enabling technologies of the Semantic Web. TidalTrust extends the FOAF standard [24] to incorporate trust semantics, and uses a network-flow approach to infer trust for indirectly-connected network members. The algorithm propagates trust across the maximal trust links in the graph formed by the FOAF documents in the network, and supports binary 'trust-or-distrust' as well as continuous rating scales. Through evaluation of the algorithm, Golbeck demonstrates that shorter paths between members of a network correlate to higher accuracy of the inferred trust rating between the two. TidalTrust is the basis of the trust system in the FilmTrust community discussed below, as well as TrustMail, an email client that displays inline trust ratings alongside messages.

2.4.3 Appleseed

Cai-Nicolas Ziegler's Appleseed [25] is an eigenvector-based trust algorithm, similar to EigenTrust, developed to be the trust metric of a decentralised trust-based recommender system. Currently, recommendation systems such as Amazon.com's product recommendation system are centralised, with a single server or cluster of servers storing data and performing recommendation calculations. Appleseed attempts to provide the underlying trust metric for a distributed recommendation engine that could perform a similar function on the Semantic Web or Peer-to-Peer systems, where data is stored in a fully distributed manner. The trust algorithm itself is a *local group* algorithm based on *spreading activation models* that drive the propagation of trust through a network, with trust decay based on a node's *spreading factor* (similar to the propagation of energy through an inefficient process). The term *local group* refers to the fact that although Appleseed provides a global calculation, it only explores the minimum necessary group of nodes to generate its rankings.

Rather than providing a numerical rating for the amount of trust a given node has, Appleseed returns a ranked list of the network's members ordered by trust. Also, due to its basis on principle eigenvectors, Appleseed suffers from the issue that as a node grants trust to more and more peers, the trust is in a sense diluted – that is, when a node rates another, all rated nodes experience diminishing returns in terms of the trust they are given. As a result, nodes are penalised from being overgenerous in granting trust to others.

2.4.4 myTrust

Quinn's myTrust is motivated by the fact that existing trust mechanisms employ a single value to represent trust (e.g., eBay's 'reputation', FilmTrust's one-to-ten rating scale, Advogato's certifications, etc.), whereas in reality there are often multiple constituent factors that combine to determine that single value. Quinn postulated that a single-

faceted rating is insufficient for capturing the diverse meanings of the term trust across the countless contexts in which trust is useful. In response, Quinn developed a multi-faceted trust model that is specialisable, so that it is meaningful in any context, and personalised, so that within a single context it satisfies all users' subjective views of trust.

Although Quinn's model of trust is unnamed, for the purposes of clarity this thesis refers to the model as the myTrust model. The associated trust management service, that Quinn named myTrust, will be referred to as the myTrust service.

The model itself consists of an upper ontology, representing the various trust concepts, and a meta-model, representing the possible relationships between the elements of the upper ontology. By using the upper ontology and meta-model, specialised and personalised models can be generated. The upper ontology contains eight trust concepts that Quinn determined were important to the computer science community when discussing trust: belief, competence, confidence, credibility, faith, honesty, reliability, and reputation. Within this group of eight concepts, five are considered *concrete*, or well-defined: competence, credibility, honesty, reliability, and reputation. The remaining three – belief, confidence, and faith – are more ambiguous and open to interpretation, and in the language of Quinn's thesis are dubbed *abstract* concepts. This distinction between *concrete* and *abstract* concepts is useful when dealing with generating specialised trust models based on the upper ontology and meta-model. Within the meta-model, *concrete* concepts can be derived from other *concrete* concepts (e.g., credibility is derived from a person's strong reputation); *concrete* concepts can be informed by *abstract* concepts (e.g., one's confidence in a person is informed by that person's competence); and *abstract* concepts can be affected by other *abstract* concepts (e.g., having high faith in a person affects the amount of belief one has in that person).

By specifying sets of meta-model relationships between the upper ontology concepts and creating any domain specific factors that constitute those trust concepts, specialised models can be generated. Also, users can specify their own ideas about the relationships between the concpets to create a personalised trust model. Combined, the two models

can be used to determine a user's trust in another with respect to a specific context. Once the models have been defined, the myTrust service allows for users to create trust annotations for others, calculation of trust using either opinion-based or evidence-based algorithms, and policy support.

2.5 Trust in Online Social Networks

Several existing OSNs implement trust mechanisms, most typically in the form of trust-based recommendation engines, or to display trust annotations alongside user-created data.

2.5.1 Advogato

Advogato [26], a blogging and social networking site for free software proponents, implements a global network-flow based trust algorithm to determine the reliability of information users post [27]. Site users rate others according to a three-level certification: *apprentice*, *journeyman*, and *master*. Depending on an individual's overall rating within the network, they may be restricted to read-only access, or be granted the right to post content to the site. Like Appleseed, Advogato uses a local group calculation to determine trust.

The major focus of Advogato's trust implementation was to devise an attack-resistant trust mechanism. By determining flow from several a priori trusted seeds in the network (the site's creators), the algorithm determines graph cuts that remove untrusted nodes and their certifiers from the trusted section.

2.5.2 Epinions

Epinions [28] is an online consumer review community that provides user-contributed domain-specific product reviews. Users can rate the reviews they read on a five point

scale of *off topic*, *not helpful*, *somewhat helpful*, *helpful*, or *very helpful*. Additionally, users can choose to trust or block other members of the community. Epinions uses each member's web of trust to determine which reviews to recommend to him in the future, exploiting the social phenomenon of people preferring to receive product recommendations from those they know and trust, rather than unknown peers with similar tastes. As a user becomes more trusted by the rest of the community, his reviews and ratings of reviews carry greater weight than less-trusted members, and similarly to Advogato's certification levels, Epinions grants users more rights to site features as their trust increases.

2.5.3 FilmTrust

FilmTrust [29] is a research project that applies the TidalTrust trust mechanism to an online movie review community. In a similar manner to Epinions, users can specify how much they trust other community members, in this case, by rating individuals on a one-to-ten scale. The trust ratings are then used to determine the relevance of reviews on the site to that user.

2.5.4 miniOSN

miniOSN is unique among current OSNs that implement trust due to its multifaceted model. Advogato, Epinions, and FilmTrust all rely on a single trust value to determine the trust rating for other community members. Quinn argues that due to the subjective nature of trust, such a single value rating could represent different aspects of trust to different community members. miniOSN allows users to rate each other across the eight factors of trust used by the myTrust trust management service.

Users can create a profile, add friends, post photos and blog entries, and comment on friends' posted content. Additionally, they can specify their relationship with each friend (colleague, family member, physical friend, etc.) and rate them according to belief, competence, confidence, credibility, faith, honesty, reliability, and reputation. When a

user uploads content, he can apply access conditions, so only friends with equal or higher trust ratings across the various factors are able to see the photo or blog post.

2.6 Summary

This chapter presented an overview of OSNs, in particular Facebook, its privacy support, and the Facebook Development Platform. It also provided a broad view of trust and its properties, as well as a review of existing trust mechanisms, especially the myTrust model and service. Finally, it covered existing trust support in OSNs, with a particular focus on miniOSN, the prototype OSN incorporating the myTrust model as the basis for its trust support.

Chapter 3

Design and Implementation of Application, Round One

3.1 Introduction

This chapter outlines the design decisions and implementation details involved in the first iteration of the myTrust application for Facebook. Section 3.2 details the overall design, with sections 3.2.1 and 3.2.2 focusing on the influences of the myTrust model and the miniOSN project on the resulting design. Section 3.3 covers the implementation details, and section 3.4 provides a summary of the chapter.

3.2 Design

The majority of the feature-set of the myTrust application for Facebook comes from the work of Quinn and Fu.

3.2.1 Influences from a Multi-faceted Model of Trust that is Personalisable and Specialisable

Quinn's trust model and the features of the myTrust trust management service provide the basis for the Facebook application developed during the course of work on this thesis. Specifically, the application must allow users to create personal models of trust, annotate other users with trust ratings, and create access rules, all based on the eight trust concepts from myTrust's upper ontology. Additionally, users should be able to express their own personal ideas about the relationships between the eight concepts in order to define a personalised model of trust.

Because domain-specific knowledge is necessary in order to generate a specialised trust model for Facebook, the first round of implementation does not incorporate any specialised model. Instead, it presents users with all eight trust characteristics and allows them to specify their feelings about the eight factors by defining the weight of each concept, effectively providing a ranking of the importance of the concepts with relation to Facebook. This weighting data, combined with information gathered during the evaluation stage, when users were asked about which factors they deemed most important with respect to Facebook in particular, was gathered with the aim of possibly using this feedback to guide the generation of a specialised model, if one was necessary.

3.2.2 Influences from miniOSN

Fu's miniOSN project most directly influenced the design of the first iteration of the application. The majority of the features developed for this iteration were intended to replicate the functionality of miniOSN on a wider scale. Additionally, several of the features that Fu noted as future improvements to miniOSN were implemented in this application as a result of her findings.

miniOSN was created to apply Quinn's multifaceted model of trust to an OSN. As a result, it provided users the following features:

- Profile creation: users can create an identity on the site. This is a core OSN function.
- Relationships: users can link their profiles with other people's profiles to specify friendship, colleague, or family relationships. This is also a core OSN function.
- Content posting: users can post photos and text content to the site, and can comment on others posted content.
- Content access rules: users can specify minimum trust ratings that others must have in order to view their content.
- Trust rating: users can rate their friends, colleagues, and family members according to the eight trust factors of the model, and can edit these ratings at any time. Because of the personal nature of these ratings, they should be kept private at all times.
- Transitivity of trust: when rating a friend, the current user can specify whether or not all unrated friends of that friend should be given the same rating.

Features such as profile creation, relationship creation and maintenance, and content posting are features already offered by Facebook. The application developed for this research implements user rating, rule creation and editing in the same manner that miniOSN does.

Differences Between Quinn's and Fu's Work

The myTrust service provided several trust calculation algorithms that distilled the eight-factor trust ratings into a single average, based on the specialised and personalised trust models in question. Fu determined that doing so in the context of an OSN led to several useability problems – firstly, it reduced the multifactor ratings into a single value, which could be confusing and unintuitive to users; and secondly, it reduced the ability of people to reason about trust ratings, as it became impossible to tell the difference between two

users rated at 75% (for example), even though the users could have vastly different ratings when mapped to the eight factors of the underlying trust model.

Additionally, miniOSN did not provide a mechanism for users to define personalised models of trust. Because it was the first attempt at applying the myTrust trust model to an OSN, trust model specialisation and personalisation was beyond the project's scope. Instead, the project focused on determining the applicability of the concepts of the myTrust upper ontology to a social networking context.

Differences Between miniOSN and the myTrust Facebook Application

The first iteration of the application developed for this thesis for the most part incorporates the features of miniOSN, with a few exceptions.

First of all, in order to provide users with the ability to capture their own subjective views of trust, the application allows them to set weights on each of the various characteristics of the upper ontology. These weights are used to compute a weighted average of the eight values, which allows users to quickly compare their friends' ratings.

Secondly, whereas miniOSN avoided calculating a single-value average trust rating across the eight concepts, the myTrust application for Facebook does so, providing a weighted average based on the user's specified personal idea of trust. While users rate each other and create access rules across the eight characteristics, they can get an impression of their friends ratings at a glance with the single value. If necessary, they can view a friend's multifactor rating in detail, allowing them to reason about the differences between two friends who may have the same single-value rating with different underlying ratings.

Because Fu developed miniOSN herself, she had full control over the integration of the trust features with users' profiles and posted content. Unfortunately, due to the sandboxed nature of the Facebook Development Platform, as noted below, it is not possible for an application to actually control the visibility of a user's profile content to others. As a result, the myTrust application for Facebook was developed as a proof of concept to demonstrate the possible future integration of trust semantics into the Facebook network,

and evaluate and investigate the issues associated with integrating the two.

Fu also found that users were overwhelmed by having to keep track of ratings and rules themselves. Because there was no feedback as to which friends would be affected by a given rule, users found tailoring rules to specific friends difficult. As a result, the myTrust application for Facebook allows users to see which friends are affected by a given rule, and which rules affect a given friend.

3.2.3 Features of myTrust for Facebook

The first major feature of the Facebook application is the ability for users to specify a personalised model of trust by setting the weights of the eight trust concepts that constitute the general model. Figure 3.1 demonstrates the personalisation settings, where users can assign higher weights to the trust attributes they find more important. These weights govern the calculation of the single trust value that lets users quickly compare their friends' ratings at a glance.

The application allows users to rate their friends based on the trust model. Figure 3.2 shows the rating page for a particular friend of the current user. It also shows the user which resources the friend is denied access to based on user-defined rules, and which exact trust requirements the friend fails on, highlighted in red. Figure 3.3 illustrates the aggregate ratings page, where a user can see all of their friends' ratings at once, as well as each friend's overall rating, calculated as per the user's personalised trust model.

The friend rating page also shows users suggested trust ratings of people that have not been rated – either unrated friends or people that they are not directly related to. These ratings are provided by what is effectively a recommendation engine, based on friend relationships. The ratings provided are translated through the intermediary friend's personalised trust model and the current user's trust model. Although trust decay across links is not implemented, the translation typically enforces an implicit decay due to the fact that unless the individual is trusted 100% by the intermediary, the recommended

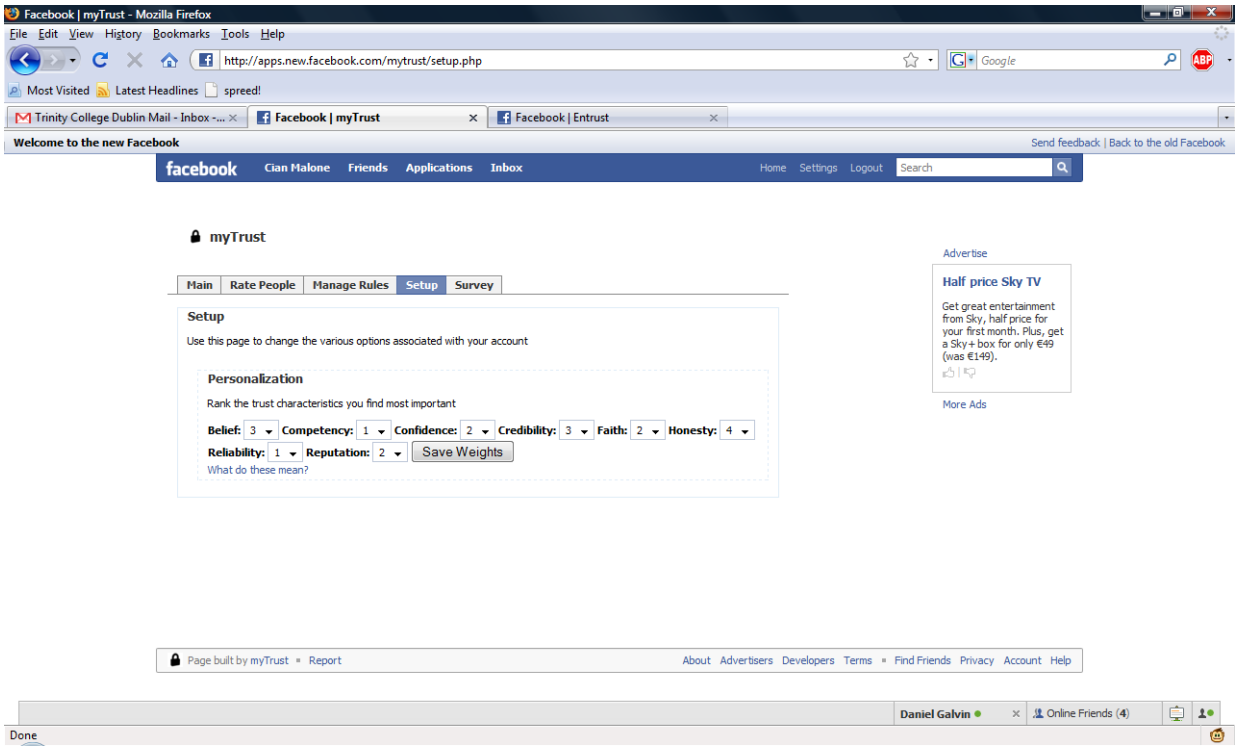

Figure 3.1: The setup page allows users to personalise their trust models by assigning weights to the various concepts.

rating is the result of multiplying two fractional ratings.

The third major feature offered by the application is the ability to create, edit, and delete rules regarding the trust ratings necessary to access user resources.Figure 3.4 shows a user's general rules page, where he can see and edit existing rules, as well as create new rules, and figure 3.5 shows the page for a specific rule, providing the user with feedback about which of his friends are affected by a given rule. The resource dropdown is populated with Facebook profile elements such as gender, home address, phone number, and more, allowing users fine-grained control over access to specific features of their profiles.

By default, if a user does not define a rule for a particular resource, it is assumed that the resource is accessible by all. Additionally, if a user does not rate a particular friend, it is assumed that the friend is fully trusted – that is, no rules govern that friend's access to the user's profile elements. This is a time-saving feature, because it allows users to apply rules and ratings only to the resources and friends with respect to which they are concerned about maintaining privacy.

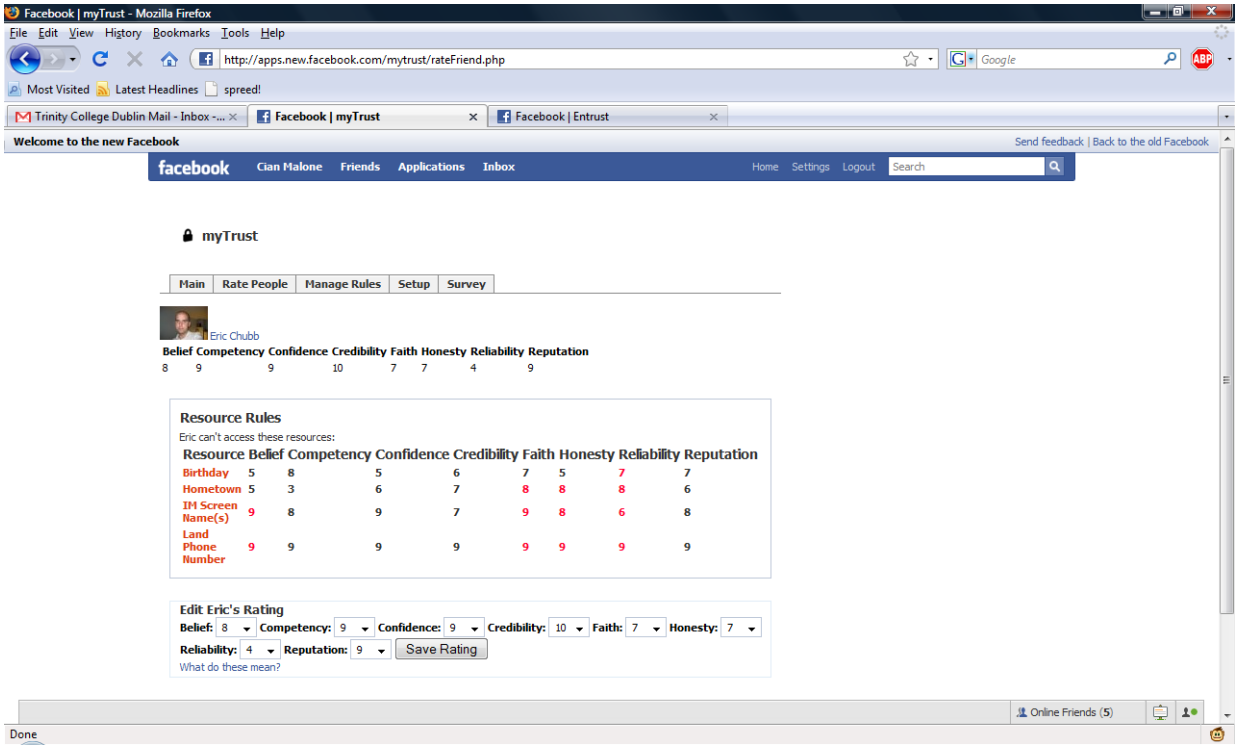

Figure 3.2: This page shows a particular friend's rating, as well as any resources the friend is denied access to. On this page, the user can edit the friend's rating.

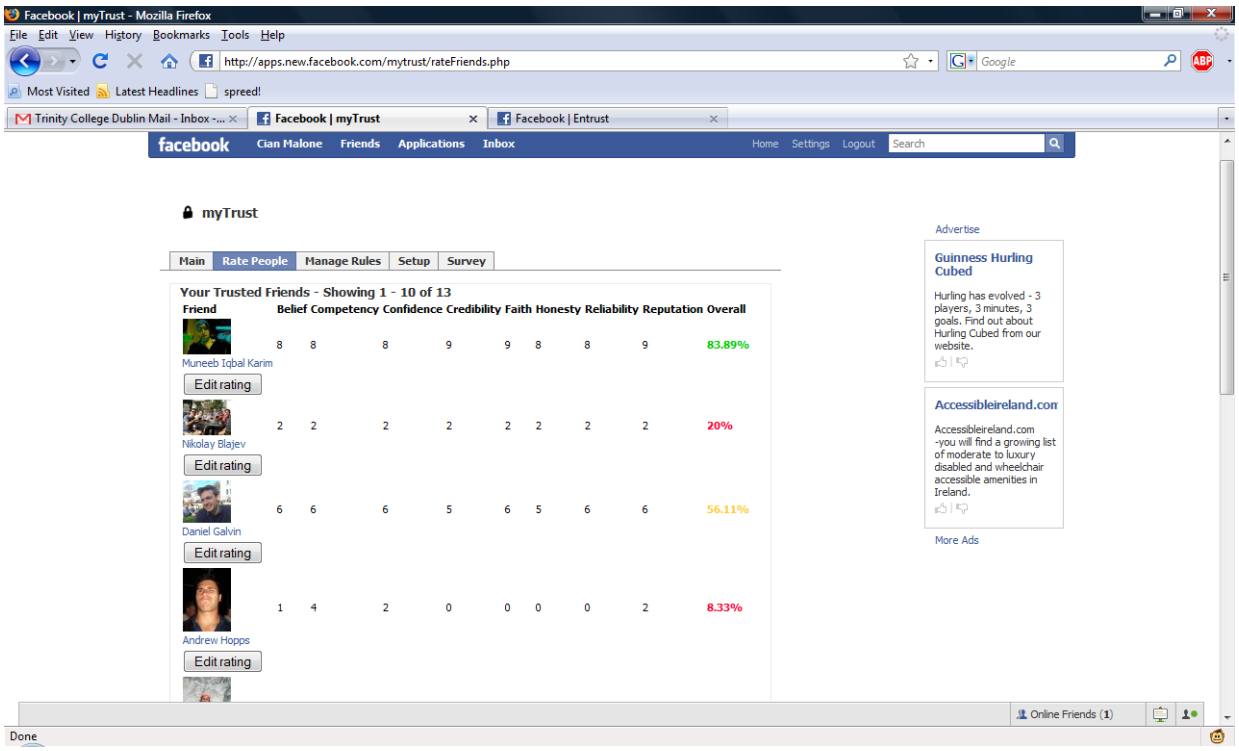

Figure 3.3: The 'Rate Friends' page gives users a more detailed picture of their friends' ratings.

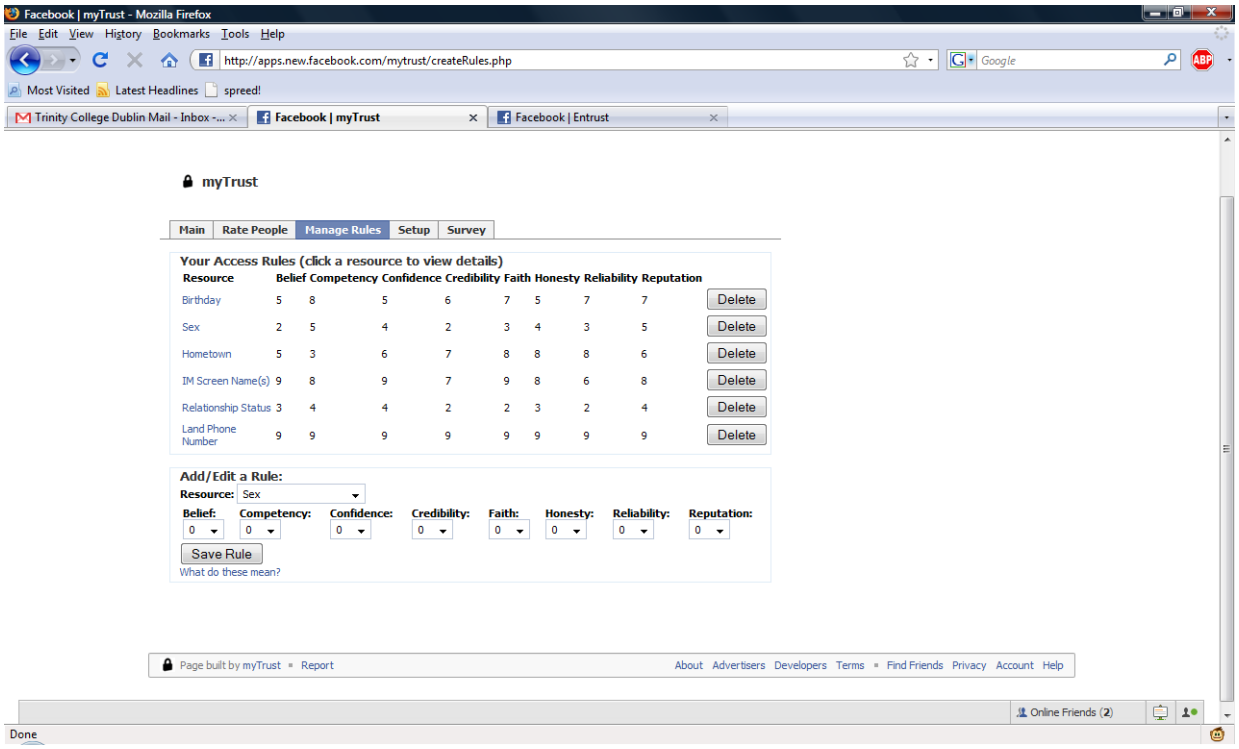

Figure 3.4: The 'Rules' page provides an overview of all the current user's rules, and allows him to create new rules or edit existing ones.

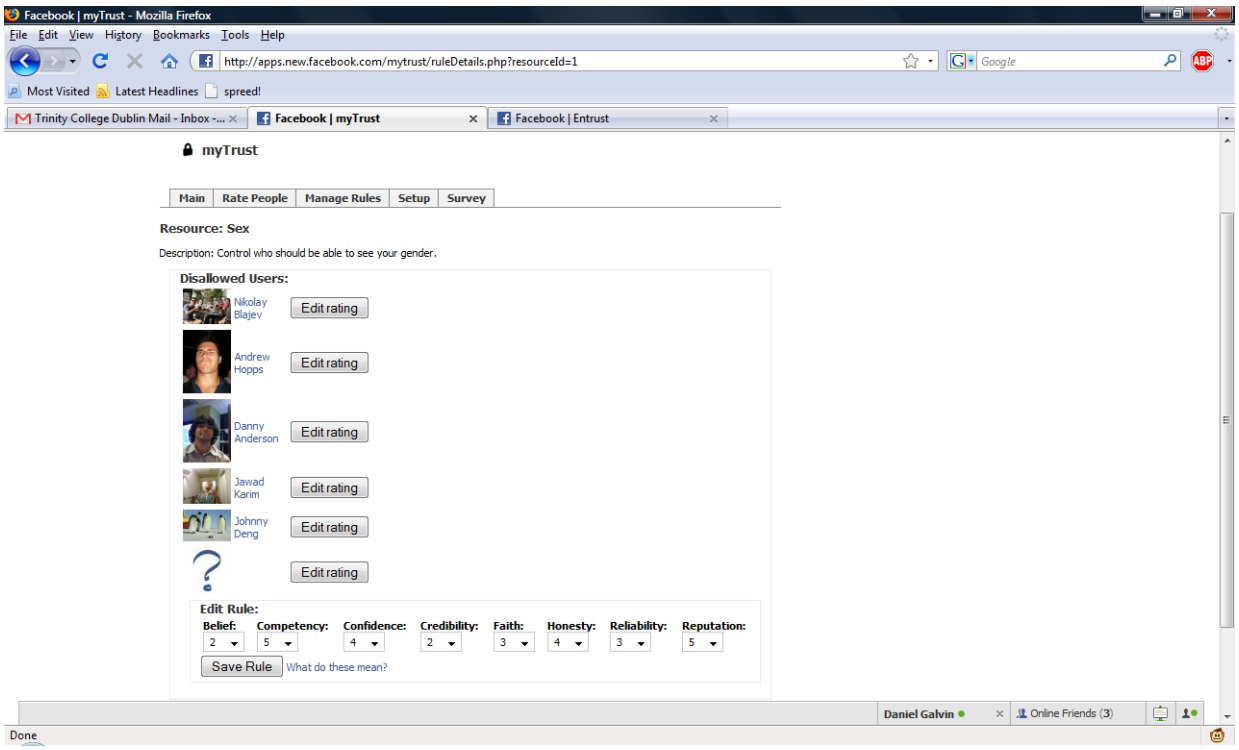

Figure 3.5: This page outlines a particular rule, and provides feedback about which friends are denied access based on the rule.

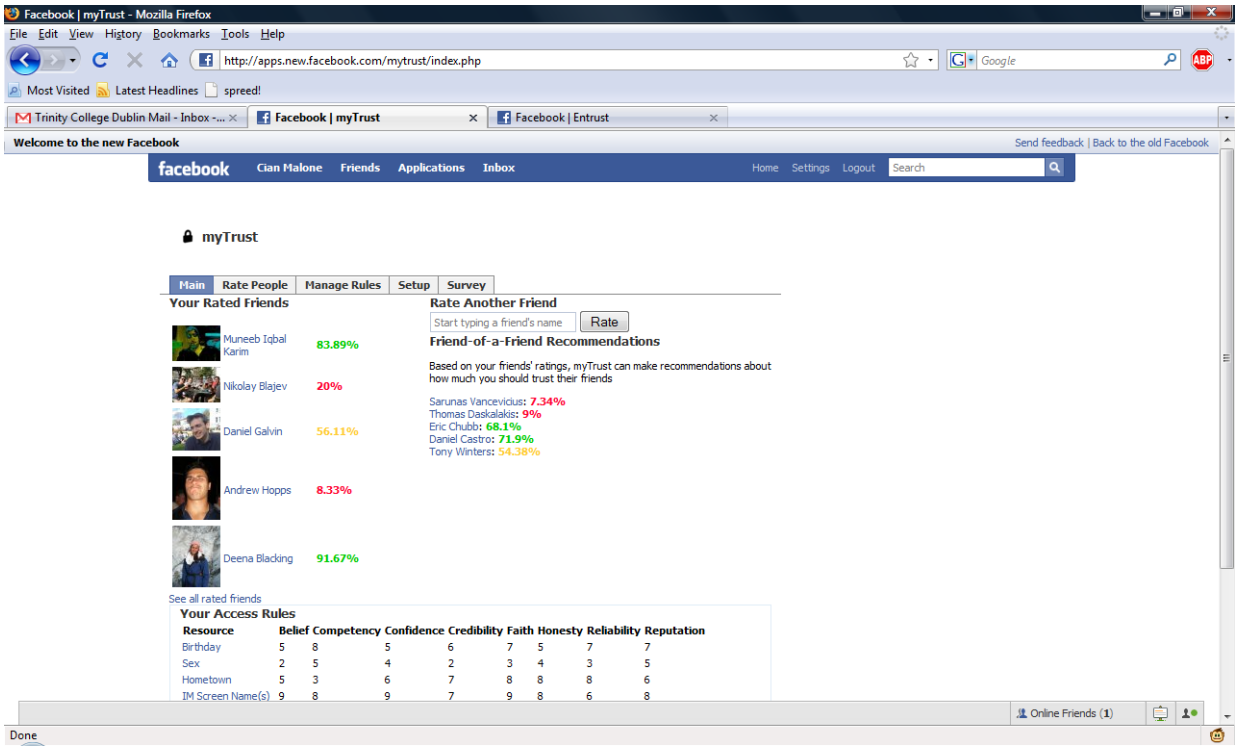

Figure 3.6: The application's home page allows users to get an overview of their rated friends, friends of friends' suggested ratings, and the rules they have created.

Finally, it was important to provide the user with a broad overview of their activity with the application. Figure 3.6 illustrates the application's home page, where at a glance, users can view a selection of their friends' ratings, friend-of-a-friend rating suggestions, and resource rules. The aim of providing this information on one page was to address the issues of information overload experienced by users of the miniOSN project. By placing friend ratings and rules on one page, users can quickly determine the relationships between the various sets of information.

3.3 Implementation

3.3.1 Facebook Platform

The Facebook Platform is a set of APIs and site features that allows third-party developers to create applications that leverage the network's social graph. The platform allows

access to friend lists, dedicated space to display application content on users' profiles and elsewhere, and viral marketing channels to increase user uptake. The following sections provide a brief overview of the components of the platform.

API

The Facebook API is a set of REST-like interfaces to access the underlying social graph. It provides methods for logging a user into the site, accessing a given user's friends list, and more.

FBML

Facebook Markup Language (FBML) is a markup language, similar to HTML, that allows developers to quickly add standard Facebook elements to a page with minimal programming effort. For instance, the <fb:name uid='x'> tag renders the name associated with the given user id, and the <fb:profile-pic uid='x'> tag renders that user's profile picture.

FBJS

FBJS is a sandboxed subset of Javascript, designed to facilitate AJAX-like functionality within a Facebook application, while maintaining the security of the platform.

FQL

Facebook Query Language (FQL) is a query language based on SQL that allows for more flexible data gathering than is possible through the Facebook API methods.

Canvas & Profile Boxes

Profile boxes are sections of users' profiles that can be used to display static content provided by an application. Because many applications rely on providing dynamic content to users, workarounds based on FBML and the 'mock-AJAX' functionality of FBJS are necessary to simulate dynamism on profiles. These restrictions, along with the inability of

applications to access anything on the profile outside their boxes, make it impossible for the myTrust application to actually control visibility of profile elements to rated friends.

The Canvas is a separate page dedicated to a particular application. The restrictions governing dynamic content on a user's profile do not apply to the Canvas, so this is where the majority of user interaction occurs. In fact, in the case of the myTrust application, the Canvas page constitutes the entire user interface of the application, and the profile is not altered in any way.

Viral Channels

Facebook provides several methods for developers to exploit the social nature of the network in order to increase the number of users of their applications. By posting events to users' Newsfeeds and Minifeeds, and allowing users to send application invitations to their friends, a good application can increase its user base exponentially.

Viral uptake was not a major concern for the myTrust application, so it does not post anything to users' feeds and does not ask users to invite friends to use the application. The personal nature of rating friends' trust meant that users may have been wary of using the application if it made any use of the viral channels. However, these channels could prove useful for future studies, because they provide an easy way to greatly increase the number of users involved in a study, if there is a good way to incentivise rapid uptake.

3.3.2 PHP

The application itself was written in PHP. The motivation for this decision was the fact that the official Facebook API client, maintained by Facebook as the API evolves, is written in PHP. Using another language would have meant either hoping that the third-party client developers were as proactive about maintaining the client as Facebook themselves, or personally altering the client in the event of an API change.

Additionally, PHP is a widely used server-side scripting language, with a huge amount

of developer resources available on the Internet. Using PHP meant that it was possible to get a functional application built quickly, which was important in order to fit two development and evaluation cycles within the thesis timeline.

3.3.3 Architecture

Figure 3.7 shows the overall architecture of the myTrust application. The myTrust application itself lies within the curved box, with the Facebook architecture (the application server, API server, and database of social connections) lying outside.

Application Server

The Facebook application server is a Web server responsible for presenting the content delivered by the third party application to the user's browser. It also converts FBJS into actual Javascript, and translates FBML into actual HTML elements before rendering the page.

PHP Pages

These pages compose the application's view. Pages were written for each of the major functions of the application (rating a friend, viewing all ratings, setting personalised concept weights, viewing and creating rules, etc.). Each page is responsible for interacting with the Facebook API and the TrustEngine class to create, retrieve, update, or delete trust annotations, settings, and rules. The pages create a mixture of HTML and FBML and deliver this content to the application server for presentation to the user.

API Server

The API server provides a set of REST-like Web services that application developers can call to authenticate users, access the underlying social graph and viral channels, evaluate FQL queries, and send email to users. The myTrust application uses the API to authenticate the current user and to find friends to rate.

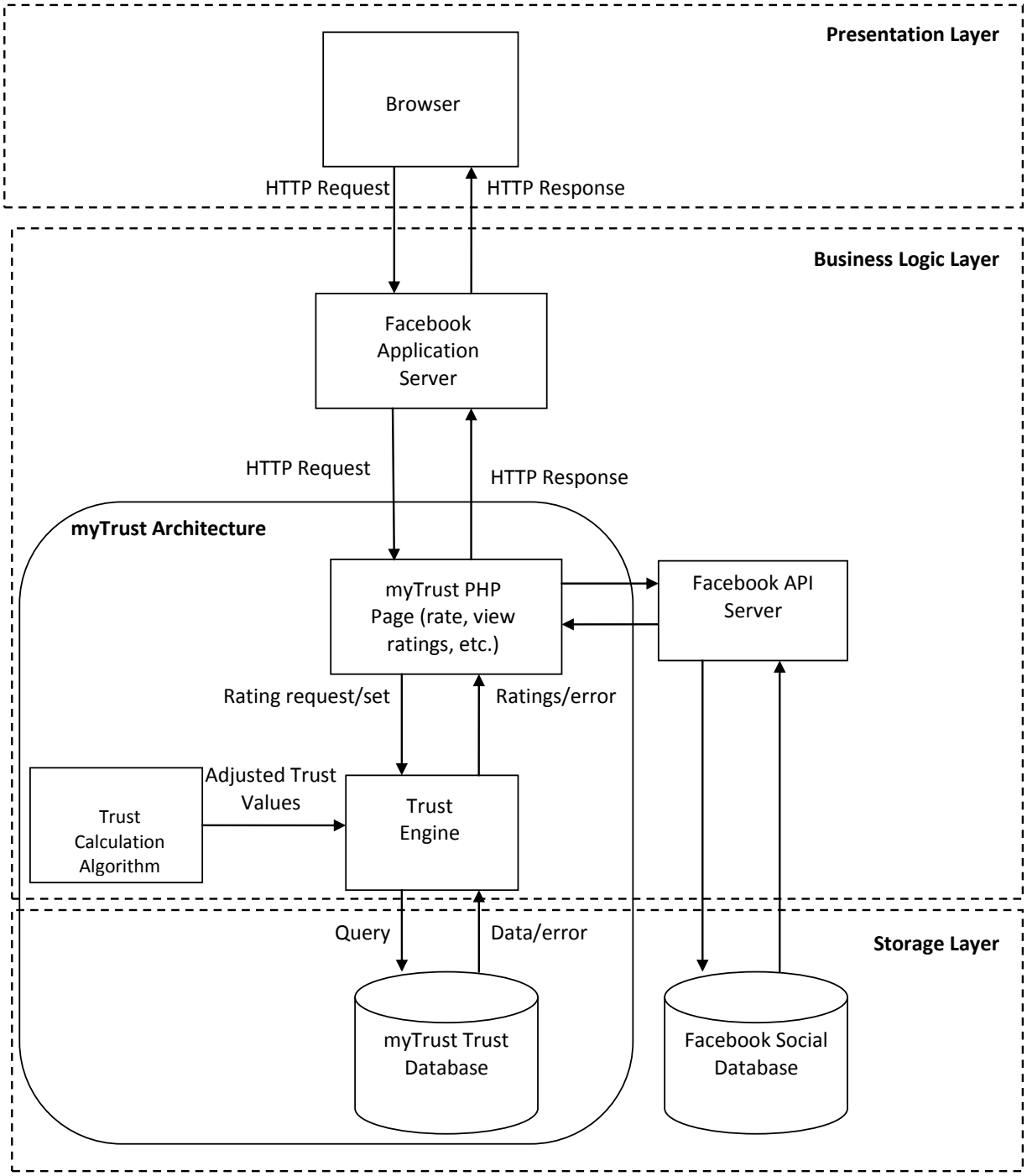

Figure 3.7: myTrust application architecture.

Trust Algorithm

Two simple algorithms for calculating trust were developed for the first iteration of the project. The first is a simple unweighted average of the ratings of the eight trust characteristics of the myTrust model. The second calculates a weighted average based on the personalised weights a user supplies for each of the eight factors in order to capture the fact that users may consider one or more of the eight more important than others in the context of Facebook activity.

TrustEngine class

The trust engine is the workhorse of the application. It provides access to the trust database, and provides methods for creating and altering trust ratings and rules, applying personalised weights to the ratings, and inferring trust across relationships.

The TrustEngine class exposes methods to the PHP pages in order to allow creating ratings, rules, and personalisation settings; retrieving these values to display to users; updating these values; and deleting existing entries. Additionally, the class exposes methods for determining which users are affected by a given rule, which rules apply to a given user, and providing inferred trust ratings for indirect relationships.

By supplying a trust algorithm to the engine, it is possible to alter the method of calculating the overall trust rating for an individual. The application uses the trust algorithm to determine the personalised trust rating of a given user based on the current user's trust model settings.

Persistence Layer

The persistence layer is composed of four MySQL tables to store ratings, rules, resources, and personalisation settings.

The first of these tables, UserTrust, stores trust annotations. This table contains an auto-incrementing primary key, the rating user's Facebook identity number, the rated

user's Facebook identity number, and the values of each of the eight trust concepts.

The rules table stores user-defined rules regarding resource access requirements. This table contains an auto-incrementing primary key, the user's Facebook identity number, a resource identifier (a foreign key into the resources table), and the values of each of the eight trust concepts that must be met in order to satisfy the rule and be granted access to the resource.

The resources table stores resource identification information. Alongside an auto-incrementing primary key, the resource name and resource description are stored. As a result, it is possible to add or remove resources as the elements Facebook allows on users' profiles change.

Finally, the personalisation settings table stores information relating to each user's personal model of trust. The table contains an auto-incrementing primary key, the user's Facebook identity number, and values for the weights of each of the eight trust concepts. These weights are retrieved when calculating personalised trust values.

3.3.4 Implementation Difficulties

In retrospect, the architecture relies too heavily on a single class, the TrustEngine, to provide the majority of application functionality. While this did not cause issues during the development of the first iteration of the application, it makes extending the application overly difficult, due to the massive codebase in a single class file and the fragility of the code within that class.

This issue was addressed with the development of the second iteration of the application, which completely refactored the code base according to object-oriented design principles.

3.4 Summary

This chapter discussed the influences of previous work on the first application developed in the course of this thesis. It outlined the various design decisions that were made, as well as implementation details, including functional and technical architecture, design concerns, and implementation difficulties associated with the first iteration of the application.

Chapter 4

Evaluation, Round One

4.1 Introduction

This chapter presents the results of evaluating the first iteration of the application. Section 4.2 discusses the small-scale user study conducted, and section 4.3 outlines the results of a survey conducted to determine Facebook users' attitudes toward trust, privacy, and the application. Section 4.4 provides an interpretation of the results of the survey. Section 4.5 gives a summary of the chapter.

4.2 User Study

A small-scale user study was conducted to evaluate overall application usability. Four computer science students were asked to try out the application. They were tasked with setting the weights of their personalised trust models, rating several friends according to the eight trust concepts, and setting several rules governing the ratings required in order to see aspects of their profiles. Finally, they were asked to complete the survey discussed in the following section.

A couple of issues arose during the study. Because users were unfamiliar with the concept of trust in relation to Facebook, they did not immediately understand the capabilities

of the application. More time spent on user interface decisions, tutorial information, and documentation could address this issue. The second issue was the ambiguity between several of the trust concepts, which led to confusion on the part of the users. To a certain extent, the ambiguity is intentional, because without it, the application would not allow users to define their subjective views of trust. However, the ambiguity also led to users questioning the need for eight concepts where they felt that fewer could capture the same facets of trust.

All users completed each task despite usability concerns, and all stated that they found the application of trust to Facebook to be a useful concept when making privacy decisions on the site.

4.2.1 User Backgrounds

The four users selected for the study were students from the Networks and Distributed Systems M.Sc. course at Trinity College Dublin. Their ages ranged from 23 to 25. All regularly use Facebook for social networking, and all have some experience using and developing applications for the platform. Their technical knowledge and friendships with the author should be noted when considering the results of the study.

4.3 User Survey

4.3.1 Survey Overview

The short survey included in Appendix A.1 was hosted on SurveyMonkey. The link was posted to a group of approximately 200 Facebook users, as well as a mailing list of computer science postgraduates and staffmembers at Trinity. In total, 57 people responded to the survey. The survey was composed of ten questions designed to elicit demographic information and to ascertain users' attitudes and behaviours with respect to privacy and trust on Facebook.

Users were asked to identify how important they feel privacy is on Facebook, first with respect to sharing the information on their profile with friends, and then with respect to sharing that information with strangers.

They were also asked to rate the importance of each of the eight trust characteristics with respect to three Facebook use cases. First, they evaluated the importance of each factor in determining whether or not to accept a friend request from a person they personally know offline. Secondly, they rated the factors when deciding whether or not to accept a friend request from a friend of a friend. The third situation asked users to rate the concepts when confronted with the task of completing a transaction with a stranger on the Facebook Marketplace.

An error in the design of the survey resulted in belief not being present in the ranking questions. However, given the responses of the users in the small-scale study, it is likely that the attitudes users' displayed towards concepts such as honesty and credibility are closely aligned with their attitude toward the belief concept. Unfortunately, the results of the survey cannot be used to verify this on a larger scale.

Finally, users were given the option of providing any extra feedback they desired as freeform text.

4.3.2 Survey Findings

In total, 25 people installed the Facebook application, and 57 people completed the survey. Of those surveyed, the mean age was 29, and the median age was 26.

The demographic information is outlined in figures 4.1 and 4.2. Approximately 63% of those surveyed stated having degrees in a technical field such as Computer Science or Engineering. 40% use Facebook on a daily basis, 26% weekly, and 34% monthly or less regularly. Several respondents noted that they do not use Facebook, but because one purpose of the initial survey was to gain domain-specific knowledge with the aim of using the responses to create a specialised trust model at a later date, the survey was

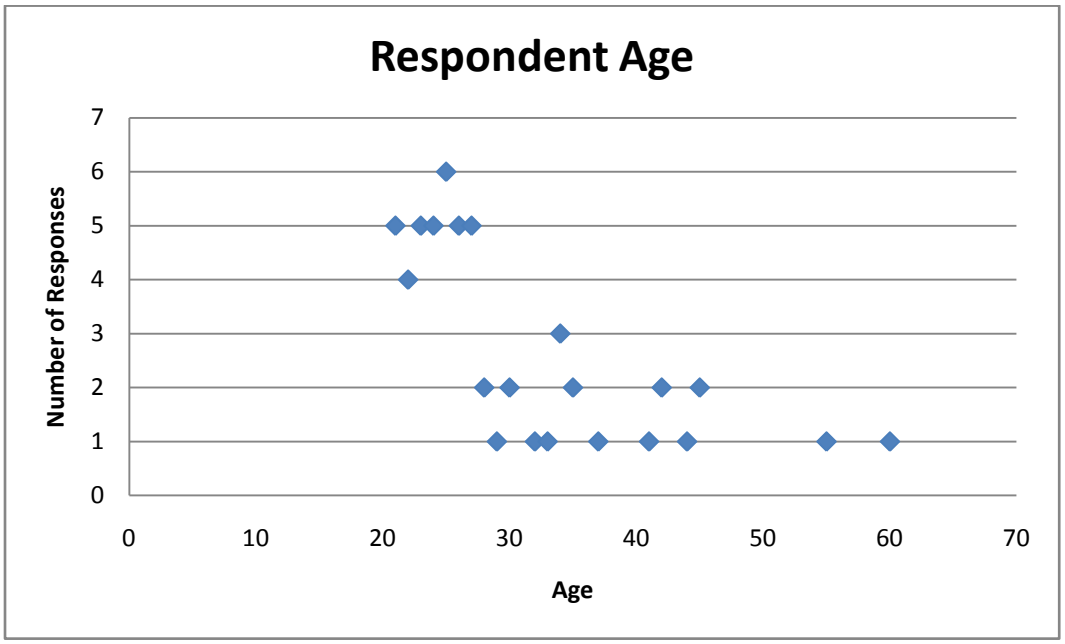

Figure 4.1: The age distribution of the initial survey respondents.

deliberately not tailored to these users.

Users were asked to generally rate the importance of privacy on Facebook, both with respect to friends and strangers. Figure 4.3 illustrates the distribution of responses. 15 of 53 respondents felt that privacy was not important with relation to providing profile information to their friends, whereas 10 felt that even with regard to friends, privacy was very important. With respect to strangers, most people felt that maintaining the privacy of their profiles was important – 24 respondents expressed that privacy is very important in such circumstances. One respondent stated that privacy was not important, even with respect to strangers viewing his or her profile.

The results of the questions that asked users to evaluate the trust model with respect to Facebook use cases are outlined in figures 4.4, 4.5, and 4.6.

For the first situation, which asked users to evaluate the importance of each trust characteristic when determining whether or not to accept a friend invitation from a real-world friend, most users expressed that each concept was not important because they know that they already trust their offline friends. Most users stated that competence, confidence, faith, and reputation were either unimportant or of average importance. Hon-

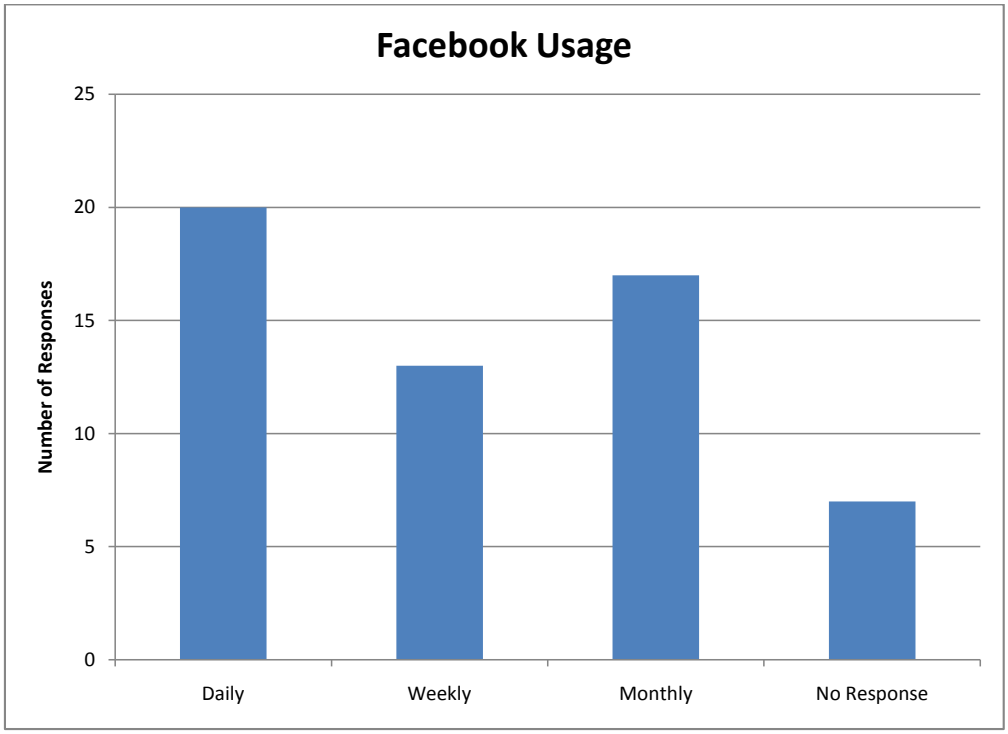

Figure 4.2: The Facebook usage responses of the initial survey.

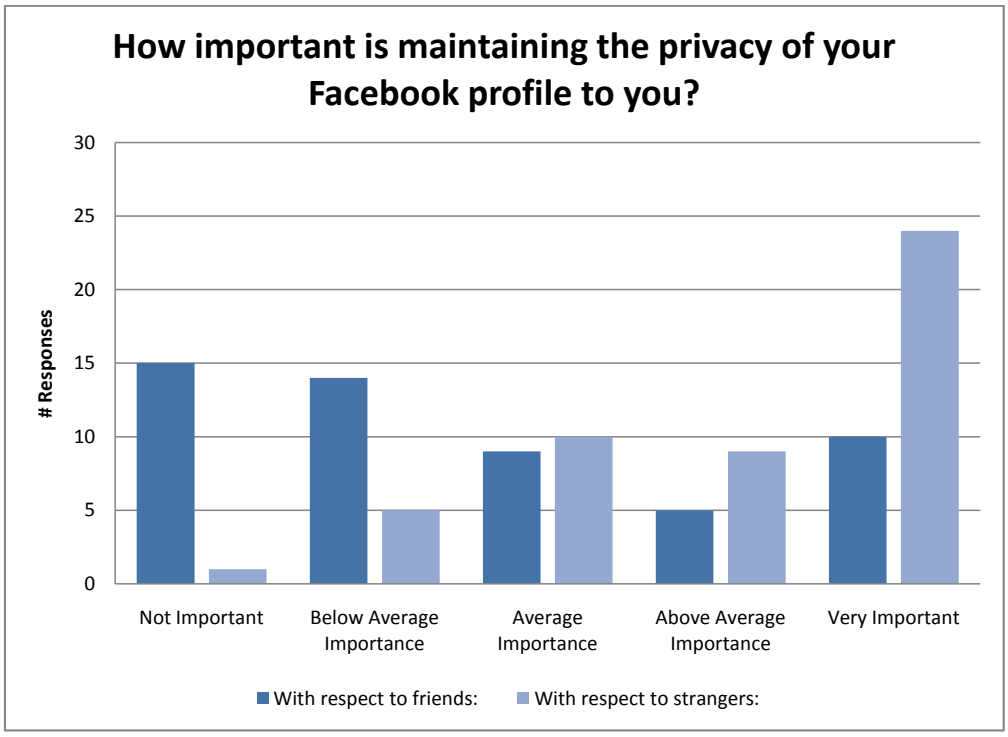

Figure 4.3: Users expressed the importance of privacy on Facebook, with respect to their friends and with respect to strangers.

esty, credibility and reliability were rated very important by more users than the other concepts.

In the second situation, users were asked to evaluate the importance of each concept when deciding to accept a friend invitation from a friend of a friend. Confidence and faith were still mostly rated as not important or of average importance. Credibility, honesty, reliability, and reputation were rated very important or of above average importance. Overall, all concepts were more highly rated than those in the first situation, because of the increase in perceived risk when determining whether or not to accept a friend request from someone unknown in real life.

The third situation asked users to rate the importance of each characteristic with respect to conducting a commercial transaction on the Facebook Marketplace. Each characteristic was rated very important by the majority of users. Credibility, honesty, reliability, and reputation were the most highly rated. 27 users responded that credibility was very important; 28 stated that honesty and reputation were very important; and 23 rated reliability as very important. The increase in importance of the concepts across the board follows the trend of the importance of trust increasing as risk increases.

4.4 Survey Analysis

The results of the survey provide insight into respondents' attitudes regarding privacy and trust on Facebook.

The importance of privacy changes drastically when people deal with friends on Facebook versus their interactions with strangers. The majority of users felt that privacy was of little or no importance with respect to their friends, suggesting that they trust their friends not to abuse the information on their profiles. However, people are strongly averse to strangers accessing that same information. Given that Facebook defaults to allowing any members of a user's network, known or unknown, to access the user's profile, users must personally alter their privacy settings in order to prevent strangers from accessing

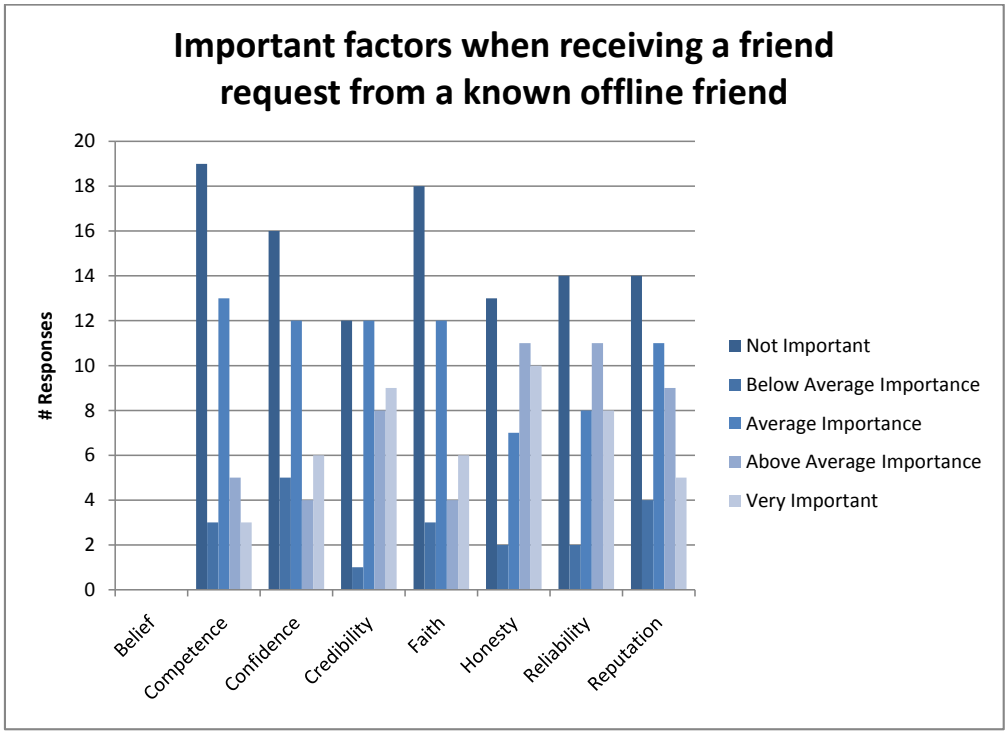

Figure 4.4: The results of users evaluating the importance of the eight trust concepts with respect to accepting or denying a friend request from an offline friend.

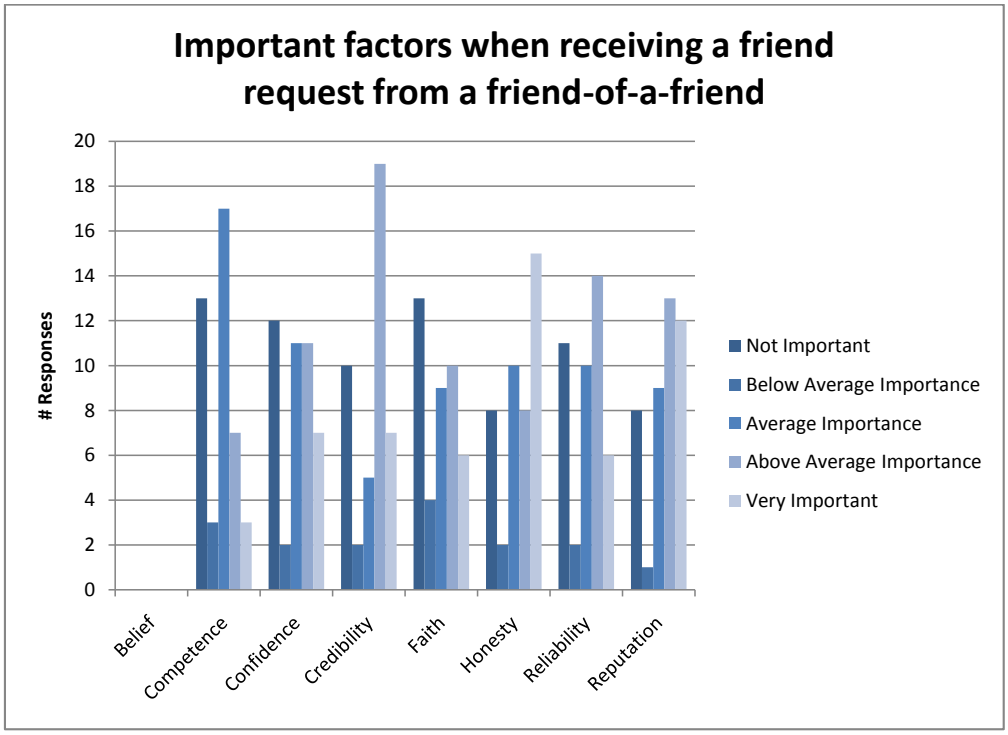

Figure 4.5: The results of users evaluating the importance of the eight trust concepts with respect to accepting or denying a friend request from a friend of a friend.

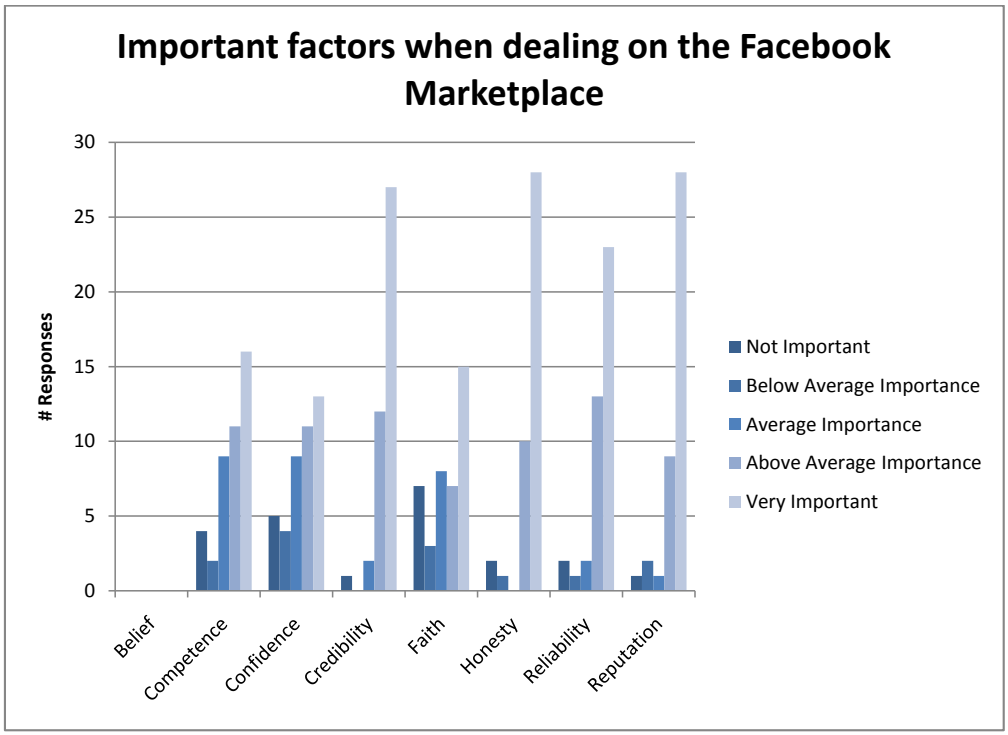

Figure 4.6: The results of users evaluating the importance of the eight trust concepts with respect to completing a transaction with a stranger on the Facebook Marketplace.

their information. This suggests that many users feel that Facebook is too lax with regard to user privacy by default.

Asking users to rate the importance of each trust concept when dealing with common Facebook use cases provided some interesting results. Because the perceived risk increases from one scenario to the next, many of the factors users deemed unimportant in the first scenario were rated increasingly important in the second and third situations. However, across all scenarios, credibility, reliability, honesty, and reputation received higher ratings than the other four, suggesting that Facebook users value those attributes of their friends' personalities higher than others. These results seem to make sense, especially due to the fact that many Facebook relationships originate offline. In the real world, people become offended when they realise that friends have been dishonest; it seems natural that the same attitude would translate to the online social network. Additionally, facets such as competence and faith do not have as much relevance when determining whether or not someone is a good friend, whereas credibility, reliability, and honesty are desirable

qualities in a friend.

4.5 Summary

This chapter presented the results of evaluating the first iteration of the myTrust application for Facebook. It presented the findings of a survey into users' habits and attitudes toward trust and privacy on Facebook, as well as their feelings about how well the application's trust model fits in the context of several Facebook use cases.

Chapter 5

Design and Implementation of Application, Round Two

5.1 Introduction

This chapter outlines the design decisions and implementation details involved in the first iteration of the myTrust application for Facebook. Section 5.2 details the overall design, including the influence of the first round survey on the resulting design. Section 5.3 covers the implementation details, and section 5.4 provides a summary of the chapter.

5.2 Design

Following the completion of the evaluation of the first version of the trust application, work began on designing and implementing a second iteration to address the issues that arose.

5.2.1 Influence of Survey Results on Design

The results of the survey demonstrated several problems with the first iteration of the application. Overall, users felt that the use of a trust model was beneficial when mak-

ing privacy decisions with respect to their Facebook profiles. However, the reliance on eight characteristics made useability and convenience a problem for many users. Because users had to use all eight concepts for each rating they provided, the time commitment discouraged users from rating lots of friends.

Additionally, because there existed some ambiguity between certain trust concepts (e.g., belief and faith), users expressed confusion regarding the necessity of all eight concepts.

To address these issues, the second iteration of the application needed to provide a way for users to rate large groups of their friends at once, and simplify the rating procedure as much as possible. In order to do so, the following features were implemented.

Additional Features

First, users should be able to define groups of their Facebook contacts and rate groups of people all at once. For instance, a user could define 'Friends,' 'Colleagues,' and 'Family' groups, populate them with their Facebook contacts, and apply a single rating to each group. That rating would then be applied to each member of the group, saving the user from having to apply individual ratings himself. In order to make this feature as powerful as possible, users can place a friend in multiple groups. Those overlapping groups can have their own ratings, so a friend who is also a colleague could easily be rated as more trustworthy than a general colleague, for instance. In a case where a friend does not belong in any of a user's groups, that friend can be rated as an individual.

Figure 5.1 demonstrates a user's list of created groups and the groups that result from overlapping membership. From this page, users can manage or delete an existing group and create new groups. Figure 5.2 shows the management page for a particular group. From here, the user can add or remove group members, control the group's rating, and see which rules apply to the group to deny resource access. Additionally, clicking a group member takes the user to that friend's individual page, an example of which is shown in figure 5.3.

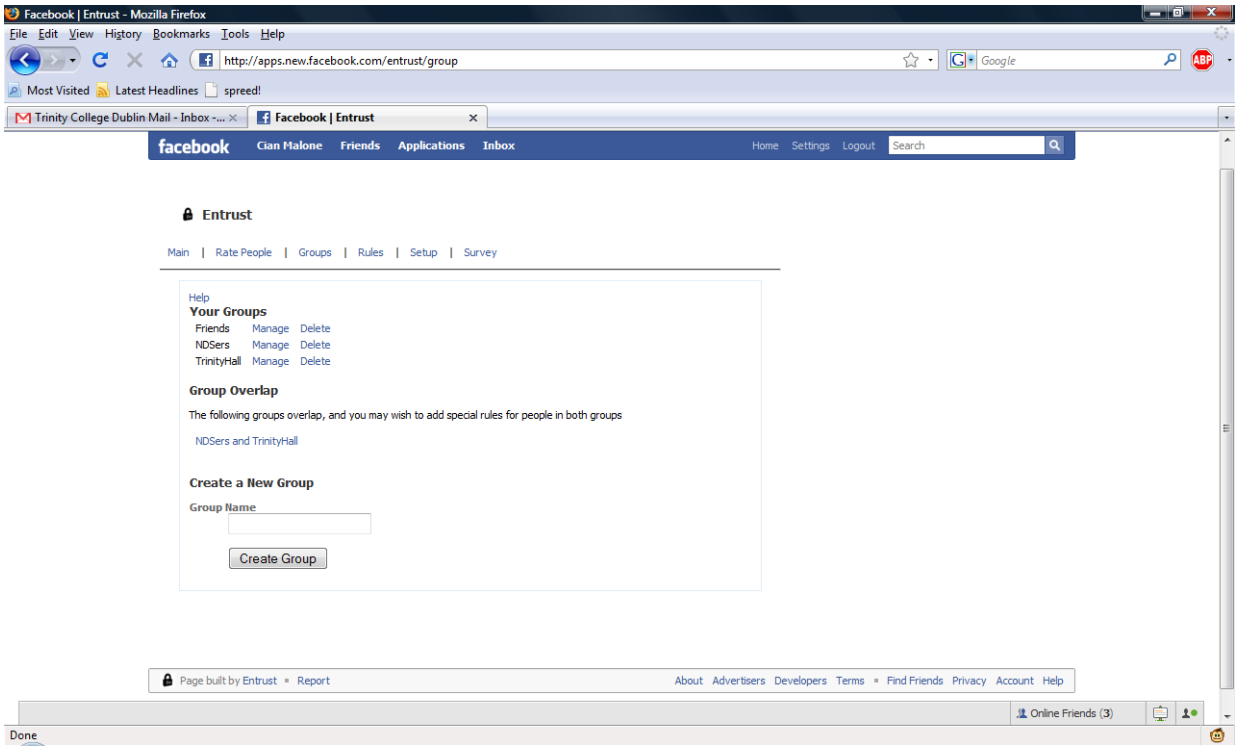

Figure 5.1: The second iteration of the application allows users to define groups of friends.

Secondly, users should be able to define tags and map the eight underlying trust concepts onto their tags. For example, a user could define two tags, 'Truthfulness' and 'Reputability,' and map belief, honesty, faith, and credibility onto 'Truthfulness' and the remaining characteristics onto 'Reputability.' From that point on, instead of having to rate friends and groups and define rules based on all eight concepts, that user would only have to supply two values, one for each tag. Additionally, the tagging feature supplies users with a way to combat any ambiguity among the concepts while still capturing their subjective opinions of the relationships between the concepts. If users prefer to use all eight concepts individually as in the first iteration, they can define tags ('Belief,' 'Competence,' etc.) and map a single concept onto each tag.

Figure 5.4 shows the setup page, where the user can define and edit tags, and control which concepts map to each user tag. Figures 5.5 and 5.6 demonstrate a user's general rules page and the page for managing a specific rule. The trust requirements for each rule are shown based on a user's tags, as are user and group ratings.

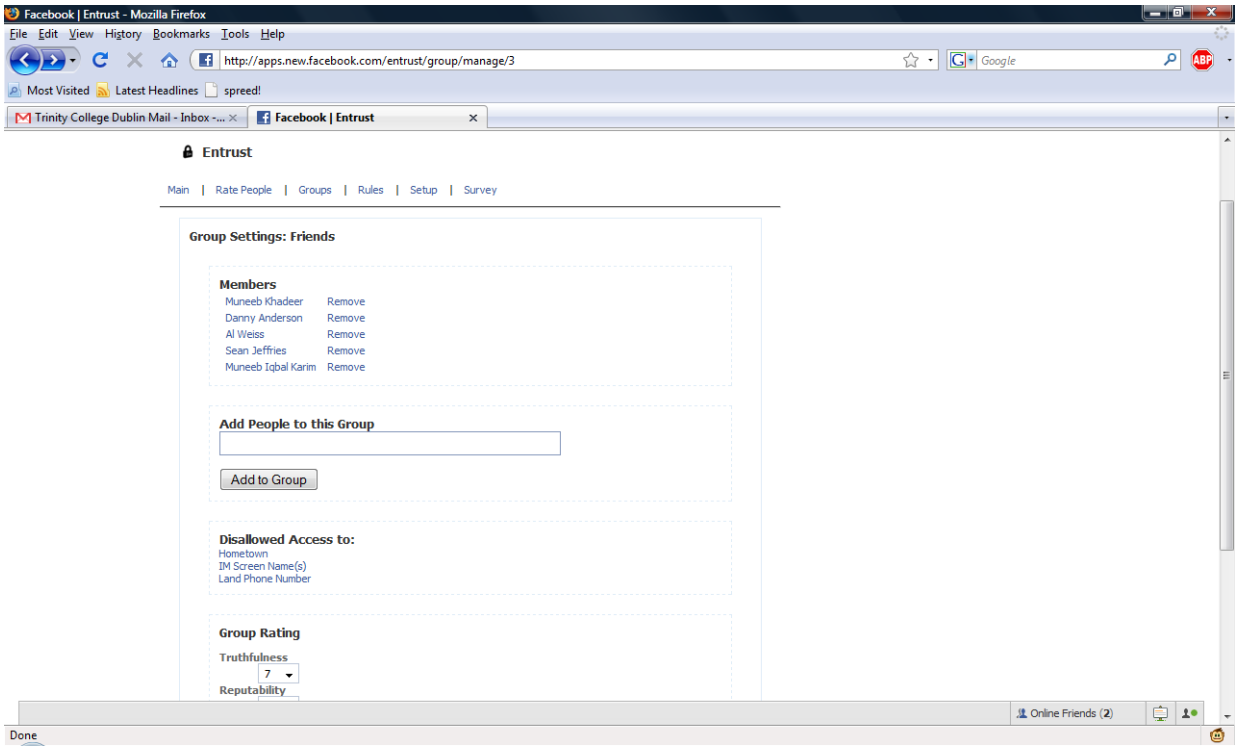

Figure 5.2: Users can control group membership and ratings, and get feedback regarding the rules that apply to the group.

5.3 Implementation

5.3.1 Zend Framework

The Zend Framework [30] is an Object Oriented framework for building Web applications in PHP. It combines a class library that provides common functionality such as database access and Web service integration with a set of coding conventions designed to promote code reuse, object decoupling, and other OO principles. In particular, most applications that use the Zend framework adhere to the Model-View-Controller (MVC) design paradigm.

The decision was made to use the Zend framework to develop the second iteration of the application in order to address the concern about the design of the TrustEngine class that was central to almost all functionality in the first version of the software. By adopting the framework and the MVC design, code could be refactored into discreet blocks

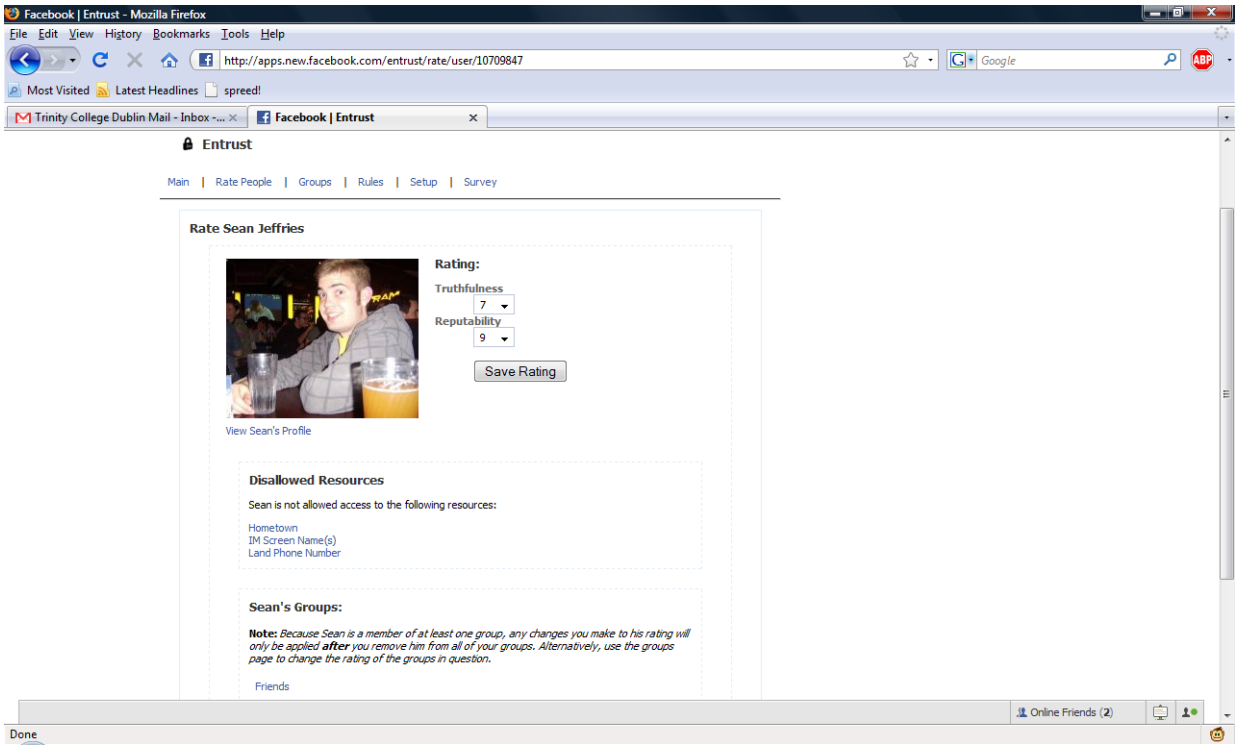

Figure 5.3: The 'Friend' page now shows the user which groups their friend is a member of.

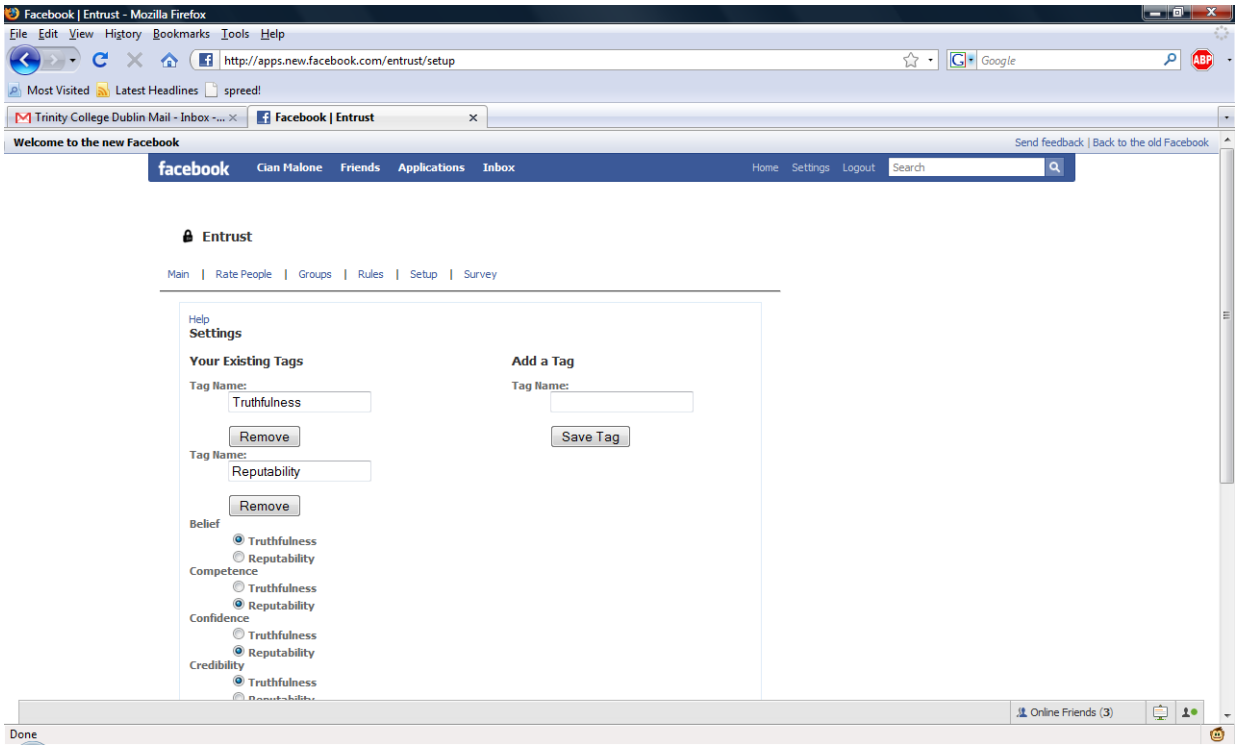

Figure 5.4: The second iteration of the application allows users to define tags that capture multiple trust concepts.

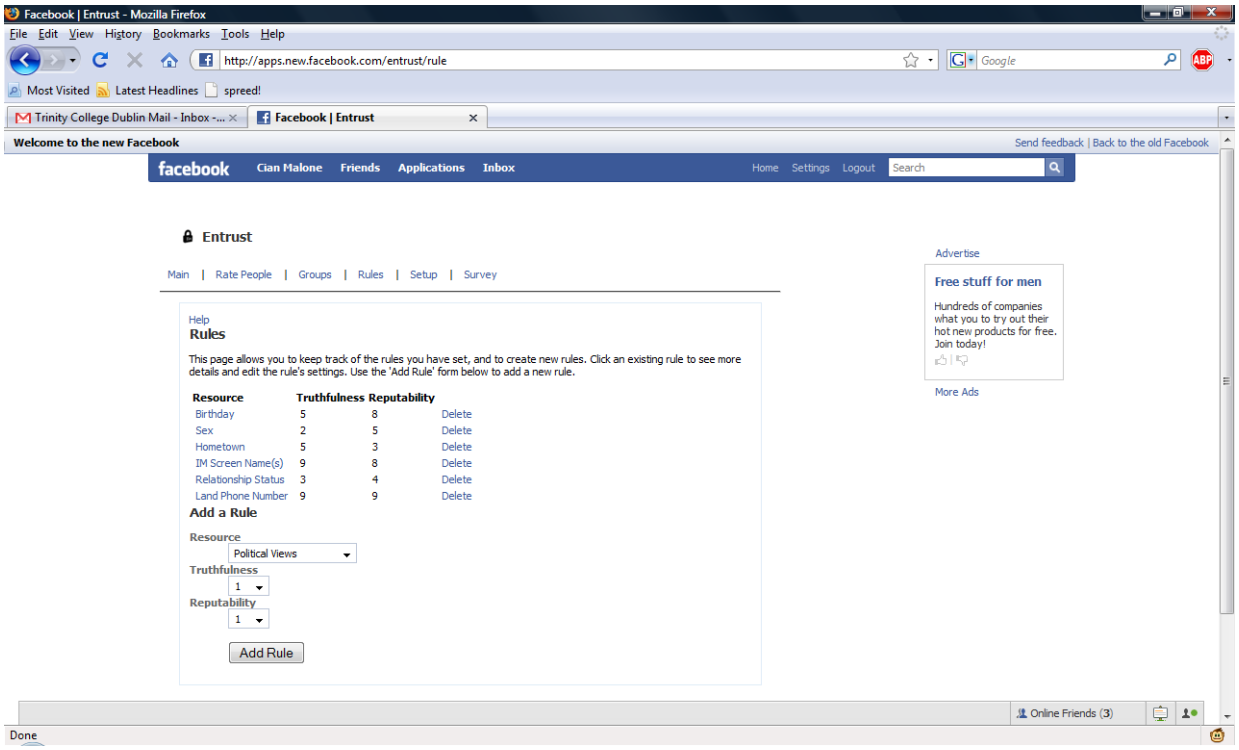

Figure 5.5: Users can create and edit rules based on their defined tags.

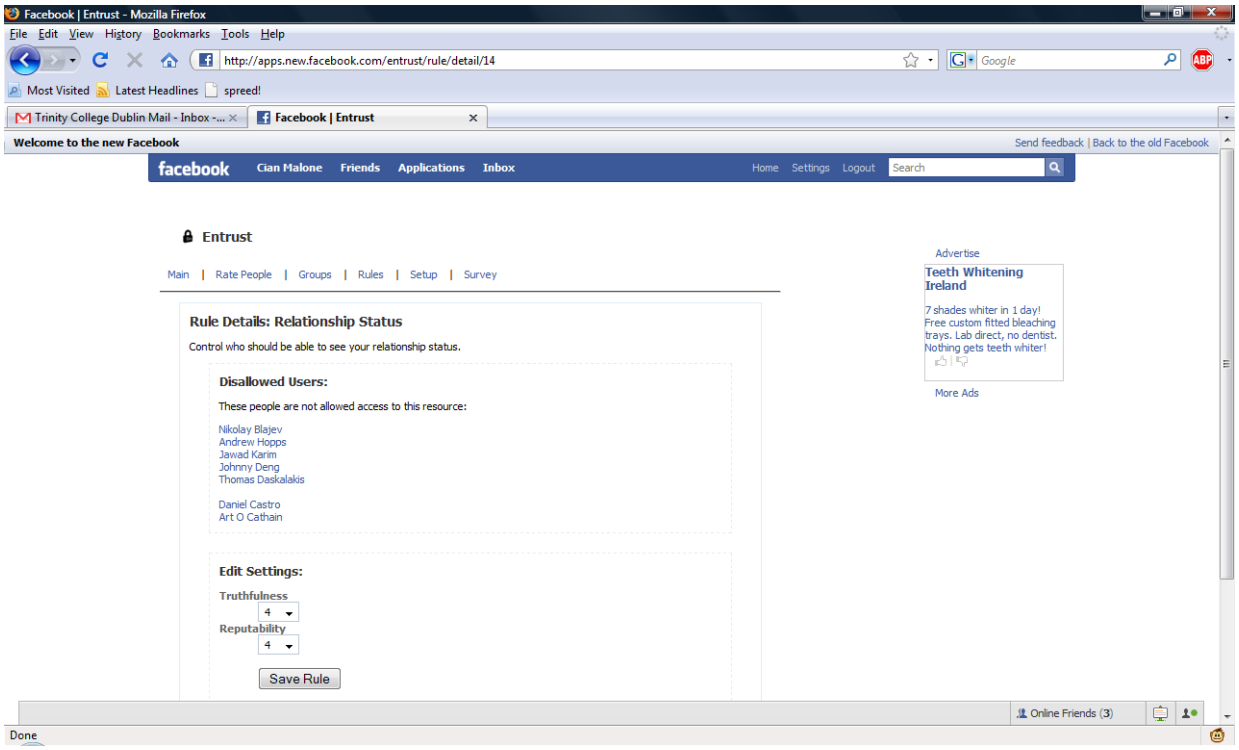

Figure 5.6: A rule's detail page allows users to edit the rule and view which groups and users are denied access to the resource.

of functionality related to specific features of the application. Additionally, achieving a standard layout for the application's pages was easier as a result of the use of view templates.

Integration with Facebook Platform

Little information exists currently detailing the ease of integration between PHP frameworks and the Facebook Platform. Initially, this was a cause for concern, and was one reason that the first iteration of the application did not rely on a framework. However, integrating the Zend framework with the platform did not prove difficult, and the benefits of the framework outweigh the minor issues that arose while integrating it with the platform.

5.3.2 Architecture

Figure 5.7 shows the architecture of the second iteration of the trust management application. The section shown is equivalent to the area inside the curved rectangle in figure 3.7, with the overall Facebook application architecture (Web server, API server, etc.) not shown. The use of the Zend Framework to govern the design of the second iteration results in a different architecture than that of the first version.

Model

The model contains the classes responsible for actually maintaining the state of the application. The model classes provide methods to the controllers in order to create, retrieve, update, and delete data from the persistence layer, through the use of a Zend_DB object that maintains a connection to the trust database server.

The User class is responsible for functionality such as retrieving or updating a given user's friend's rating from the trust repository. Additionally, the User maintains a list of Group, Rule, and Tag objects belonging to the user, to allow translation between the

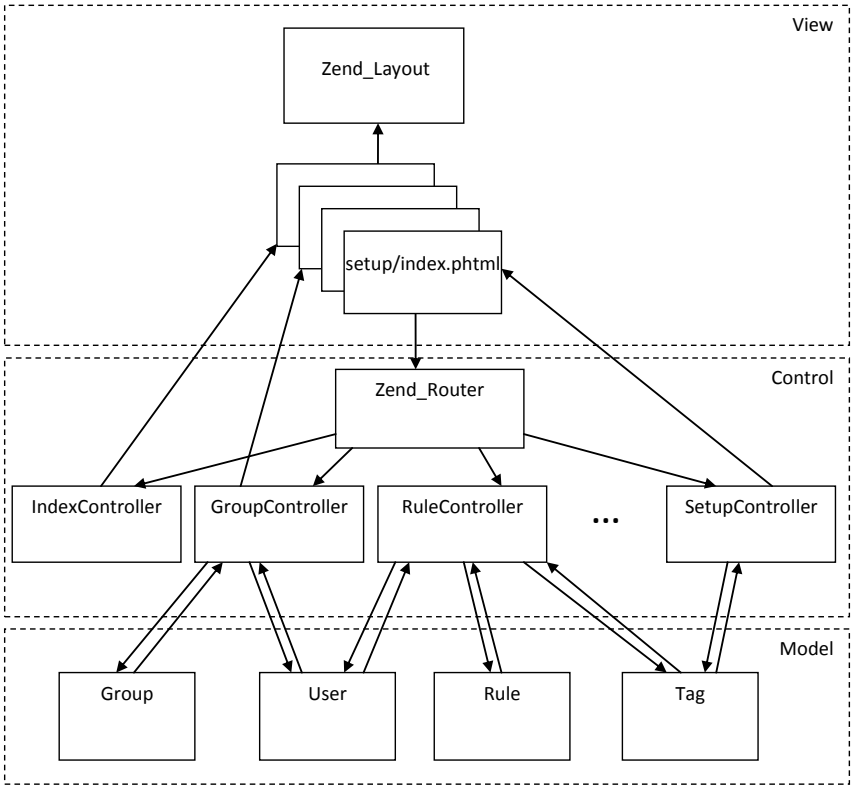

Figure 5.7: Entrust architecture

eight trust concepts that ratings are based on and the personalised tags a user creates to represent their trust model.

The Group class is responsible for controlling access to group ratings. Additionally, it provides the overlapping group functionality, allowing users to apply special ratings to those friends who are members of more than one group. Figure 5.8 shows some of the methods the Group class exposes to the GroupController.

The Rule class holds the details of the various rules a user has defined.

The Tag class is responsible for the personalisation functionality of the application. A Tag represents a subset of the eight trust concepts. As a result, the Tag class provides methods for calculating trust relative to the concepts the Tag represents.

```
public function addMember($memberId)
{
        $this->memberIds[] = $memberId;

        // Write the change to the DB
        $data = array( 'GroupID' => $this->groupId,
                               'MemberID' => $memberId,
                               'Updated' => Zend_Date::now());

        $this->dbAdapter->insert('GroupMembers', $data);
}

public function removeMember($memberId)
{
        $this->dbAdapter->delete('GroupMembers',
                        'GroupID = '.$this->groupId.' AND MemberID = '.$memberId);

        $this->getMembers();
}

public function getRating()
{
        $ratingExistsQuery = "SELECT COUNT(*) FROM GroupRating WHERE GroupID = ?";
        $ratingExists = ($this->dbAdapter->fetchOne($ratingExistsQuery, $this->groupId) == 1);

        if (!$ratingExists)
        {
                return NULL;
        }

        $sqlQuery = "SELECT * FROM GroupRating WHERE GroupID = ?";
        $rating = $this->dbAdapter->fetchRow($sqlQuery, $this->groupId);
        return $rating;
}

public function setRating($belief, $competence, $confidence, $credibility, $faith, $honesty,
                                                                        $reliability, $reputation)
{
        $ratingData = array(
                'Belief' => $belief,
                'Competence' => $competence,
                'Confidence' => $confidence,
                'Credibility' => $credibility,
                'Faith' => $faith,
                'Honesty' => $honesty,
                'Reliability' => $reliability,
                'Reputation' => $reputation);

        if ($this->getRating() == NULL)
        {
                $ratingData['GroupID'] = $this->groupId;
                $this->dbAdapter->insert('GroupRating', $ratingData);
        }
        else
        $this->dbAdapter->update('GroupRating', $ratingData, 'GroupID = ' . $this->groupId);
}
```

Figure 5.8: A selection of methods contained in the Group class

```
// Get the front controller instance and add custom routes
$front = Zend_Controller_Front::getInstance();
$front->setControllerDirectory('../application/controllers');
$front->throwExceptions($config->debug->throwexceptions);
$router = $front->getRouter(); // returns a rewrite router by default
// Add the user details route
$router->addRoute('user', new Zend_Controller_Router_Route('rate/user/:userId',
                      array('controller' => 'rate', 'action' => 'user')));
// Add the group details route
$router->addRoute('manageGroup', new
       Zend_Controller_Router_Route('group/manage/:groupId',
                                    array('controller' => 'group', 'action' =>
                                    'manage')));
$router->addRoute('manageSpecialGroup', new
       Zend_Controller_Router_Route('group/manage/:groupId1/:groupId2',
                                    array('controller' => 'group', 'action' =>
                                    'manage')));
// Add the group member removal route
$router->addRoute('removeUser', new
       Zend_Controller_Router_Route('group/removeUser/:groupId/:userId',
                                    array('controller' => 'group', 'action' =>
                                    'removeuser')));
// Add the group removal route
$router->addRoute('removeGroup', new
       Zend_Controller_Router_Route('group/remove/:groupId',
                                    array('controller' => 'group', 'action' =>
                                    'removegroup')));
// Add the rule details route
$router->addRoute('manageRule', new
       Zend_Controller_Router_Route('rule/detail/:resourceId',
                                    array('controller' => 'rule', 'action' =>
                                    'detail')));
// Add the rule removal route
$router->addRoute('removeRule', new
       Zend_Controller_Router_Route('rule/remove/:resourceId',
                                    array('controller' => 'rule', 'action' =>
                                    'remove')));
```

Figure 5.9: Setting routing rules for the application

Controller

The Zend_Router object intercepts HTTP requests, and based on a set of rules and regular expressions, determines which controller class is responsible for delivering the requested page. Based on routing rules, it also retrieves GET requests from the URL and can provide mod_rewrite-esque functionality for pretty-printing URLs. For instance, the URL '/entrust/rule/detail/2' prompts the router to pass control to the RuleController class, and specifically to the 'detailAction' method of the class, with the number 2 being a GET variable representing the resource id associated with the rule. Figure 5.9 shows the code that sets up the routing rules for the application. By programatically defining URL rules and the controllers and actions those rules

```
private function getCreateGroupForm()
{
        $form = new Zend_Form();
        $form->setAction('/entrust/group/createGroup')
                        ->setMethod('post');
        // Create text input element for group name
        $groupNameElement = $form->createElement('text',
                                'groupName', array('label' => 'Group Name'));
        $groupNameElement->addValidator('alnum')
                        ->addValidator('stringLength', false,
                                        array(4, 120));
        // Populate form with group name input and a submit button
        $form->addElement($groupNameElement)
                ->addElement('submit', 'createGroup',
                                array('label' => 'Create Group'));
        return $form;
}
```

Figure 5.10: The getCreateGroupForm method builds a form to allow users to create new groups

Each Zend_Action_Controller is responsible for a different feature of the site. Controllers were defined for handling each block of functionality – namely the application's index page, the group pages, the user-rating pages, the rule pages, the tag page, and an error page.

Each controller provides methods known as 'actions' within the framework. By default, every controller must define an indexAction method which is called when the controller's index page is requested (e.g. '/entrust/rule'). Further actions are defined simply by providing action methods for them.

The controllers interface with the model to gather, create, update, or delete the necessary data, and then pass control to the appropriate view. They also interact with the Facebook API to retrieve details about the current user and friends, and they create any necessary input forms using the Zend_Form methods before handing them off to the view for user interaction. Figure 5.10 shows a simple form-building method that incorporates input element validation (in this case, ensuring that the input is alphanumeric and within certain length bounds) to build the form used to create new groups.

When the form content is POSTed to the page, a call to `$form->isValid($_POST)` determines whether or not any elements are invalid, and displays relevant error messages to the user.

```
<?php $rule = $this->currentUser->getRuleByResourceId($this->resourceId); ?>

<h1>Rule Details: <?= $rule->getResourceName() ?></h1>
<p><?= $rule->getResourceDescription() ?></p>
<?php if (count($this->disallowedGroups) > 0) { ?>
        <div class="settingsPanel">
        <h3>Disallowed Groups: </h3>
        <p>Members of these groups are not allowed access to this resource:</p>
        <?php foreach($this->disallowedGroups as $group): ?>
        <a href="/entrust/group/manage/<?= $group->getId() ?>">
                <?= $group->getName() ?>
        </a><br />
        <?php endforeach; ?>
        </div>
<?php } if (count($this->disallowedUsers) > 0) {?>
        <div class="settingsPanel">
        <h3>Disallowed Users: </h3>
        <p>These people are not allowed access to this resource:</p>
        <?php foreach($this->disallowedUsers as $friend): ?>
                <a href="/entrust/rate/user/<?= $friend ?>">
                        <fb:name uid="<?= $friend ?>" linked="false" />
                </a><br />
        <?php endforeach; ?>
        </div>
<?php } ?>
<div class="settingsPanel">
        <h3>Edit Settings:</h3>
        <?= $this->detailForm ?>
</div>
```

Figure 5.11: The 'Rule Details' page view

View

Each view is responsible for rendering HTML and FBML to the browser. Because all logic is contained in the controllers and the model, each view is very simple. Additionally, the use of a Zend_Layout separates redundant elements of the views (such as a common header and CSS declarations) into one reusable template. As a result, each view can be very barebones.

For instance, the following code segment 5.11 is the entire view for the 'Rule Details' page.

5.3.3 Implementation Difficulties

The main issue with implementing the second iteration of the application was the fact that a large amount of refactoring and rewriting was necessary in order to redesign the software according to the framework. It would have been possible to continue with the

first iteration's architecture, but most likely it would have taken a considerable amount of time to build the required features on the existing codebase. Also, if other features needed to be implemented in the future, it would have remained very difficult to integrate them with the existing architecture. As a result, refactoring and redesigning the application, while requiring a time commitment up front, allows the application to be extended further in the future with minimal effort.

5.4 Summary

This chapter detailed the design and implementation process of the second iteration of the trust application for Facebook. First, issues from the previous version of the software were identified, and features such as concept tagging and user groups were designed to address these problems. Additionally, the application architecture was redesigned based on the use of the Zend PHP framework. The rest of the chapter outlined the revised architecture and the implementation difficulties that arose during the process of building the second version of the application.

Chapter 6

Evaluation, Round Two

6.1 Introduction

This chapter presents the results of evaluating the first iteration of the application. Section 6.2 discusses the small-scale user study conducted, and section 6.3 outlines the results of a survey conducted to determine Facebook users' attitudes toward trust, privacy, and the application. Section 6.4 provides an interpretation of the results of the survey. Section 6.5 gives a summary of the chapter.

6.2 User Study

Much like the evaluation of the first application, small-scale user study was conducted to evaluate user experience more intimately before releasing a wide-scale survey.

6.2.1 User Backgrounds

The backgrounds of the members of this study are very similar to those of the first study, and in fact, three of the four members of this study were members of the original group. All members of this study were Computer Science postgraduate students, and their ages ranged from 23 to 26. All use Facebook regularly. The feedback of the members who had

participated in the first study was especially valuable, because those users were able to objectively compare the two versions of the software.

6.2.2 Study Details

Users were given several tasks designed to elicit feedback about all features of the application. First, users were asked to create one or more tags and map the eight trust concepts onto those tags to create their own personalised models of trust. Next, users were asked to set up at least two groups, add members to the groups, and provide ratings for each group. In order to evaluate the overlapping groups feature, users added at least one friend to multiple groups. Users were asked to provide a rating for the overlapping group that resulted. Users were also directed to refer to individual friend's rating pages to note the effects of group membership and ratings on their friends' individual ratings. Finally, users were tasked with defining several rules governing access to any of their profile resources, and view the effects of those rules on their groups and rated friends.

6.2.3 Study Findings

Overall, reactions to the second version of the software were positive. All members of the study expressed that trust integration in the manner that the application provides would be beneficial to Facebook.

The three users who had been members of the original user study expressed that the group and tag features of the second iteration were preferable to the weighted individual ratings of the first version of the application. In particular, all users found the ability to rate large numbers of users at once by group to be very useful.

All users expressed that the use of tags to personalise the underlying trust model and simplify ratings was good, although all had some initial difficulty with understanding the mapping between tags and trust concepts. Additional time tuning the User Interface of this section of the application could improve comprehension. Also, the version of the

software used in the study defaulted to providing newly-created groups with no rating. All users expressed some confusion as a result of this, so before the wide-scale release of the application and survey, this behaviour was changed to make groups default to a rating of 1 for each tag upon creation.

One user expressed concern that he would offend his friends if they ever found out that he had limited access to certain parts of his profile, and that he would be offended if he found out the same about any of his friends. However, he admitted that currently, Facebook privacy controls work in the same manner. The issue of trust rating security is very important, and if a full-scale trust management solution ever were to be integrated with Facebook, significant design and implementation effort would be necessary to ensure user privacy. However, such a concern is outside the scope of this research.

6.3 User Survey

6.3.1 Survey Details

Similar to the first survey, a link was posted to approximately 200 Facebook users, as well as to a mailing list of Computer Science staff and postgraduate students at Trinity College Dublin. The survey first solicited some demographic information regarding age, gender, educational background, and regularity of Facebook usage. Additionally, users were asked about their opinion of privacy on Facebook with respect to their friends and with respect to strangers. The survey also asked respondents to outline their experience with Facebook's privacy controls, and whether or not they use trust as a factor when determining privacy settings and content to post to the site.

Users were asked whether or not they felt that trust integration with Facebook would be a useful feature. Finally, they were asked to evaluate the usability and value of the trust management application, and to provide any additional comments they wished to express.

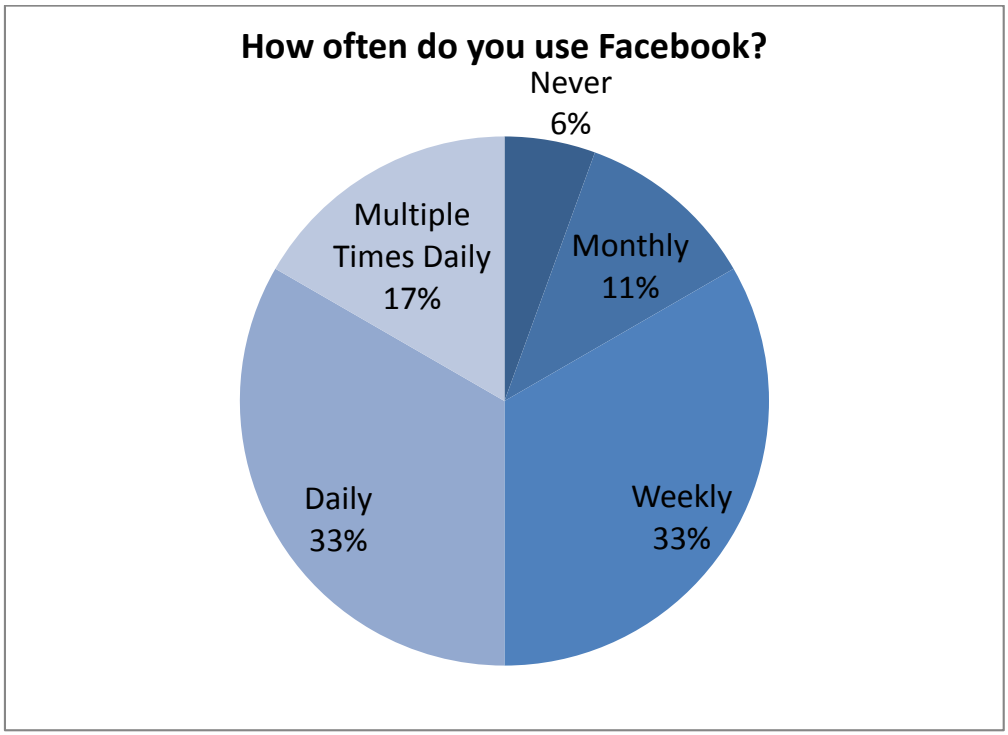

Figure 6.1: Users' Facebook usage.

A full copy of the questions posed to users is included as Appendix A.2.

6.3.2 Survey Findings

In total, 38 people responded to the survey. The average respondent age was approximately 28. Figure 6.1 illustrates the regularity of Facebook use among the respondents – 30 of the 38 respondents use Facebook weekly or more regularly, and 2 respondents never use Facebook. 27 respondents were male, and 26 of the respondents had degrees in Computer Science, Engineering, or another technical discipline.

Figure 6.2 shows the distribution of responses about the importance of privacy on Facebook. With respect to making information available to friends on Facebook, the importance of privacy was quite diverse. With respect to strangers however, almost 60% of respondents expressed that privacy was extremely important, and that they do not want strangers to be able to see their information. Although six users expressed that

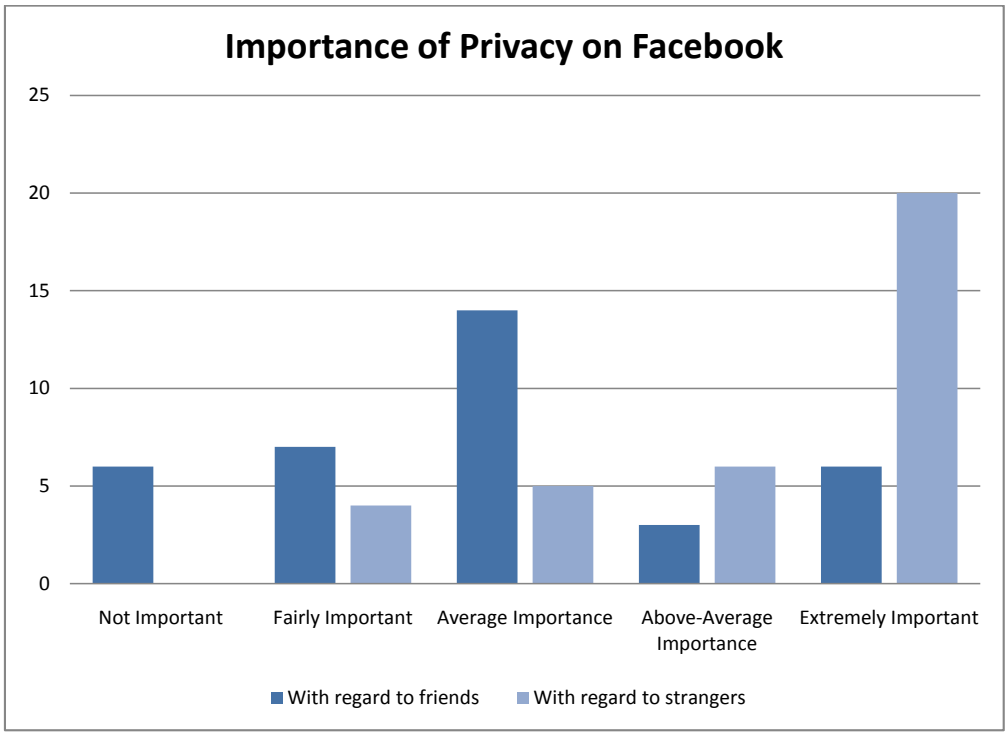

Figure 6.2: The importance of privacy on Facebook with respect to friends & strangers.

privacy was of no importance with regard to their friends, no users felt that privacy was unimportant when dealing with strangers on the network.

Of the 36 people that were regular Facebook users, one was unaware that Facebook had any privacy controls at all. 13 are aware of Facebook's privacy controls, but have never used them. Four of those 13 stated that they have not used the site's privacy features because they found them to be too complicated. 19 users said that they have used the privacy controls to govern access to their information. These results are illustrated in figure 6.3.

Figures 6.4 and 6.5 illustrate the results of asking users about the value of trust when making privacy decisions on Facebook. 26 users stated that they use trust as a factor when making decisions about whom to limit access to their profile information. Seven expressed that they do not use trust to make such decisions. Users expressed their reasons for using or not using trust when confronted with such choices. Common reasons for using trust were the fear of 'stalkers' and 'freaks' viewing private details, the potential negative impact

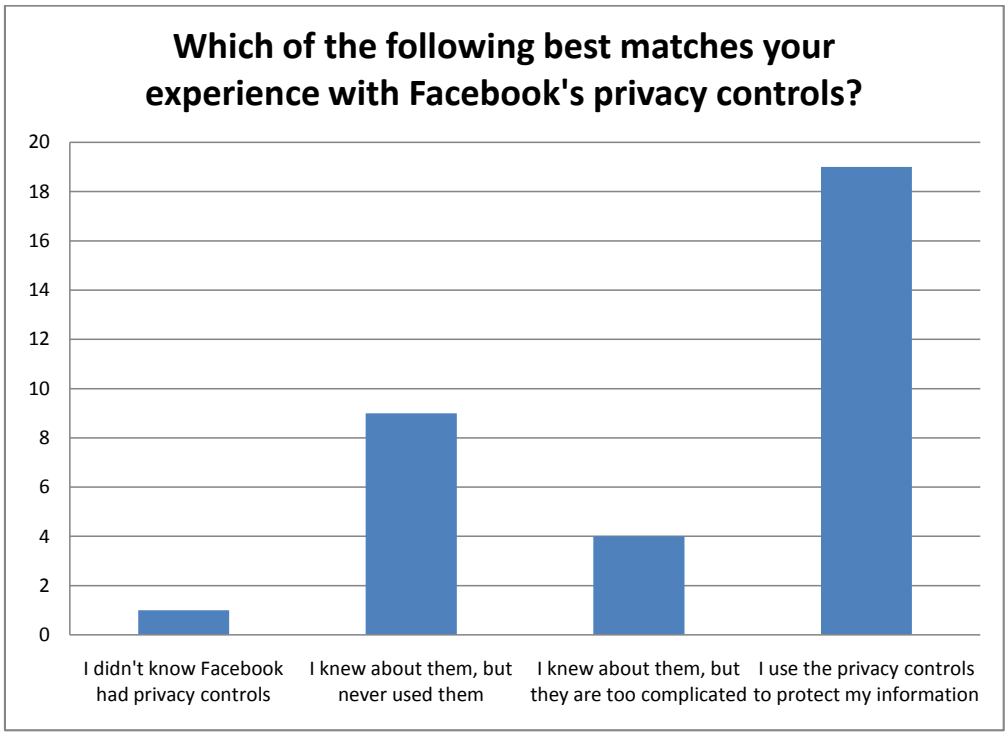

Figure 6.3: Users stated their experience with Facebook's existing privacy controls.

of the convergence of users' work and social lives on their profiles, people republishing private information and making it public, and the fact that the information on their profiles is sensitive. Users who stated that they do not consider trust a factor expressed reasons such as only having trusted friends and not putting any sensitive information on their profiles.

22 people felt that if Facebook offered trust as a mechanism for generating privacy settings, they would find it a valuable addition to site. 11 said they would not find such a feature useful.

11 respondents had used the application. Of these, 7 found it useful, and 3 did not. 5 felt that it was easy to use, and 5 did not. However, 10 of the 11 application users expressed that the application's features would be a useful way to integrate trust and privacy on Facebook.

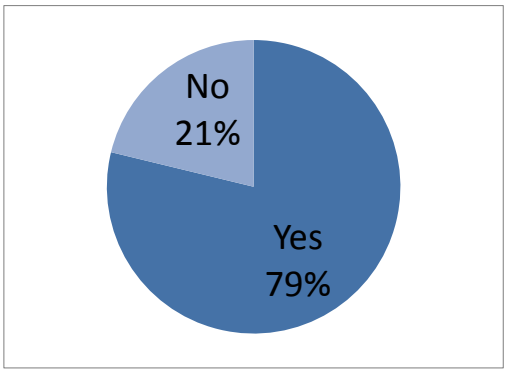

Figure 6.4: Users stated whether or not they use trust to decide their privacy settings.

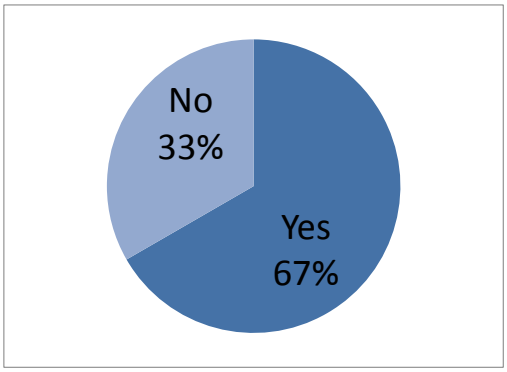

Figure 6.5: Users stated whether or not they would find trust integration valuable to Facebook.

6.4 Survey Analysis

The attitudes users expressed towards privacy on the site are consistent with those gathered during the evaluation of the first iteration of the application. Users feel that privacy is not especially important with regard to the people they are friends with on the site, because the majority of those friends are people that the user knows in real life. On the other hand, most users feel that privacy is extremely important with regard to strangers, and would not want strangers to be able to access their profiles.

When deciding whom to allow access to the information on their profiles, users find trust a valuable metric. The first survey determined that, in particular, factors such as honesty, credibility, and reliability are the most important values people use to tell who is trustworthy.

Several respondents took the time to provide additional feedback regarding their experiences with the application. A few mentioned that it was too difficult to use, reiterating the results of the survey. For instance, one user commented that the application was "too complicated and time consuming." Another mentioned that the tagging feature was difficult to understand, and that a "little more description on what was meant by 'tags' and what not during setup would be helpful." A third mentioned that it took them "some time to get around using the application." All of these statements are indicative of the fact that more time needs to be spent improving the usability of the application.

On the other hand, there were several positive comments about the application. One user stated that "facebook's privacy model is too absolute, [the application] allows for more finely grained privacy control." Another expressed that they felt that the group feature of the application was very useful and usable – "the feature of assigning a rating to a group and then adding to people to that group was very easy to use. I also thought this was a really good idea for facebook."

These user responses suggest that the majority of confusion emerged as a result of the tagging system and the process of mapping the underlying trust concepts onto their tags. The tagging system provides the most complex interaction possible in the application, so it makes sense that features such as group and rule creation were not difficult for users, whereas creating and mapping tags was.

One user suggested using a different approach to applying ratings altogether. They found that it is difficult to distinguish between trust on a 1-10 scale, and that perhaps taking a more survey-based approach by presenting users with hyptothetical situations and asking them whether or not they would allow a friend to borrow money or keep a secret for them would be more intuitive than the current approach. This is an interesting suggestion, and could be a worthwhile item of further research. Taking a survey-based approach toward creating and setting tags could also be a way to improve the usability of that feature.

One respondent reiterated the issue that a member of the user study brought up – if a

friend finds out that their access to the user's profile is limited, they may be offended. This is a difficult issue to solve. Currently, Facebook privacy controls allow users to specify that certain friends should not see all parts of their profiles. There is no difference, when a friend views the profile, between content being hidden and the content never being posted in the first place. However, the possibility that a friend could find out that they are denied access to certain parts of the profile still exists with the current privacy controls. A limited friend could shoulder-surf a trusted friend and notice that the trusted friend has access to a more comprehensive profile. Another possibility is that a user moves a friend who previously had full access to the user's profile into the list of people that have a limited view. A suspicious friend might surmise that his access to the profile has been limited when he notices that there is less information on the profile than there was previously. It is unclear how to solve this problem, either using the existing privacy infrastructure or with the trust system. In fact, Bo Fu noted that the issue of trust ratings changing over time posed the possibility of information leakage. Although miniOSN's trust ratings are private, as a friend's rating changes, that friend is granted access to different amounts of information. As a result, the friend can tell when their trust rating goes up or down.

Overall, users felt that the integration of trust management would be a valuable improvement to current privacy controls on Facebook. Users expressed satisfaction with the features provided by the application, but half of those who used it stated that it was difficult to use. In order to actually integrate the application with Facebook, more time would need to be devoted to improving user experience and simplifying the interface.

6.5 Summary

This chapter covered the evaluation of the second iteration of the trust management application for Facebook. The evaluation was conducted in much the same manner as that of the first version, with a four-user study to determine usability and get detailed feedback before releasing a survey to the general public. Several members of the user

study had participated in the first study, and so were able to offer detailed opinions about the differences between the two iterations of the software. The chapter also presented the findings of the survey.

Chapter 7

Conclusion

7.1 Project Summary

This research attempted to evaluate the applicability of a trust management system to the Facebook Online Social Network. Using Quinn's multifaceted, personalisable, and specialisable trust model, and Fu's miniOSN as a basis, an application was developed to provide Facebook users with a way to annotate friends with trust ratings, create rules governing information access, and personalise the trust model according to the concepts they find most important.

Originally, a close replication of the functionality of the miniOSN project was developed for Facebook, with the aim of evaluating the use of the features it provided on an existing large OSN. Users were asked about their general attitude towards privacy and trust when making decisions about the content they make visible on their profile. These attitudes were poorly matched to Facebook's default privacy settings regarding profile visibility and searchability. Users expressed that they value certain factors – honesty, credibility, and reliability – over others when making trust decisions on the site, and that as they engage in higher risk activities on the site, their reliance on all aspects of trust to make informed decisions increases.

Based on the results of the evaluation of the first iteration of the trust application,

a suite of new features were designed and implemented. In order to address issues of ratings being too complex, a tagging feature was built into the application. Users can define tags that are meaningful to them, and map the underlying trust concepts to those tags, simplifying ratings and capturing their subjective views of trust. Additionally, users can group their friends to apply a single rating to a large number of people at once. Users found this feature particularly useful to speed up the rating process and prevent the convergence of dissimilar groups of friends, such as colleagues and family members.

The reaction of users to trust-based privacy control was favourable. The majority of users expressed that they found the features of the application valuable, and would appreciate trust integration with the Facebook site when making privacy decisions.

7.2 Contribution

Several useful findings arose as a result of this research.

Some users expressed that they rarely have privacy concerns regarding the ability of friends to access the information they post to Facebook. Others stated that they only allow people to become their friend on the site if they already know they are trustworthy. However, most people find privacy extremely important when dealing with strangers viewing their profiles. By default, though, Facebook allows anyone within a user's network to access their profile.

Many users feel a need for trust integration with Facebook. These users expressed that the multifaceted trust model is useful for capturing their individual views of trust. With respect to trust, users stated that honesty, credibility, and reliability are the most important qualities of their Facebook friends. These three trust concepts were highly rated regardless of risk, but as the risk associated with activities on the site increases, all trust concepts become highly important.

The majority of users stated that the features provided by the application are useful. However, usability was an issue. Many people felt that the first iteration of the application

made rating friends and creating rules too cumbersome, because of the need to rate across eight values for each individual. The addition of tag-based ratings and groups to the second iteration addressed these issues, but users expressed that creating tags and mapping the trust concepts onto those tags was still overly complex.

7.3 Future Work

There are many possibilities for future avenues of research along the lines of this work.

Improved usability

Usability emerged as a major issue during the evaluation of each iteration of the application. Managing the complexity of trust and privacy settings to encourage their use is an open area of research that no existing OSN has successfully solved yet. In terms of this research, users had a hard time setting up personalised views of trust based on their tags. By providing more tutorial information and documentation, it may be possible to reduce the complexity.

One common problem that arose with the second version of the site was the fact that there were two paths through the group and settings pages – on the groups page, users could control group membership or save the group's rating, and on the settings page, users could create a new tag or change the meanings of existing ones. As a result, users sometimes clicked the wrong buttons and got confused when the application did not respond in the manner they expected. One possible solution is to make some actions asynchronous, so that users can interact with different parts of the page while their changes are propagated to the server.

During the evaluation of the second iteration of the application, the suggestion arose to apply a more natural survey-based approach to trust annotation. Instead of requiring users to apply a 1-10 rating for each friend, one respondent stated a preference for asking people to decide whether or not to confide secrets in the friend, lend money to the friend,

and other hypothetical trust-based scenarios in order to determine the amount of trust to grant that friend. Investigating such a feature could be beneficial to the application's usability, particularly if the approach could be applied to the tagging system to help users create and define tags in a more usable manner.

Trust management for other OSNs

Facebook is only one of a large number of communities on the Web today. The demographic that uses Facebook may be vastly different to those that use LinkedIn, YouTube, orkut, or others, and as a result, may have different ideas about the importance of trust within their network. Some communities may have no need for trust mechanisms, whereas others may find them essential. Within those that would be improved by trust, the specialised models that suit each network may be quite different.

Bebo licenses the Facebook platform, so the application could easily be ported to that network. Google's OpenSocial [31] provides similar functionality to the Facebook Platform for a variety of social networks, including MySpace, orkut, Ning, and LinkedIn. Porting the trust management features to OpenSocial would allow for a comparative study of the importance of trust and its constituent concepts to many more networks.

Alternatively, a trust management service for social networks based on the FOAF standard (similar to TidalTrust's FOAF trust extension) would allow comparisons with networks that support the open standard, as opposed to Facebook or Google's proprietary technologies.

Exporting trust annotations for use across the Web

Recently, Facebook released FBConnect, a proprietary technology designed to allow external sites to access Facebook's social graph. FBConnect was built to address concerns raised by communities that it was impossible for an individual to maintain and move their data, because it was owned by Facebook, MySpace, or other commercial OSNs. These communities devised standards such as FOAF, OpenID, and others with the aim of de-

centralising control of users friend connections, online identities, and content. FBConnect allows external sites access to an authenticated user's data while maintaining Facebook's control and ownership of the data. By combining FBConnect or an open standard for data interchange with the trust application, it would be possible for trust annotations from a user's OSN to follow him as he browses the Web. For instance, if a user is reading a friend's blog, a widget on the page could allow the user to view and update the friend's trust rating. When reading product reviews on Amazon, trusted friends' reviews could be highlighted to help users find reliable information. A trust-integrated messaging client, similar to the one Quinn developed to test the myTrust trust management service, could pull trust data and friend connections from Facebook for display alongside contacts' instant messages.

With more ways to use their trust annotations to help find and evaluate information on the Web, people would be more likely to take the time necessary to think carefully about trust in their communities. The use of a specialised trust model would allow for different priorities of trust concepts among the various communities on the Web, while retaining individual users' personalised opinions of those concepts.

7.4 Final Remarks

Trust is still a wide open area of research in computer science. With the explosive growth of online social networking in recent years, privacy concerns have become more and more important, because breaches affect large groups of the population. This project showed that users are unsatisfied with the existing approaches to privacy on Facebook, and that they feel that trust is a natural way to express privacy requirements.

It is conceivable that as trust research continues, trust could become a core feature of popular online communities such as Facebook. Currently, the networks that offer such features tend to be highly specialised, such as Advogato's community of software programmers, or Epinions's group of trusted product reviewers, and all rely on a single-

faceted view of trust. As Facebook and its peers mature, and users become more aware of the implications of placing personal information online, a multi-faceted trust-based system of privacy controls becomes more useful in order to appeal to the diverse populations of the sites.

Bibliography

[1] Facebook. http://www.facebook.com. Retrieved 25 August, 2008.

[2] Myspace. http://www.myspace.com. Retrieved 25 August, 2008.

[3] Bebo. http://www.bebo.com. Retrieved 25 August, 2008.

[4] Quinn, K. 'A Multi-faceted Model of Trust that is Personalisable and Specialisable', PhD Thesis. Trinity College, Dublin. 2003.

[5] Knowledge and Data Engineering Group, Trinity College, Dublin. http://kdeg.cs.tcd.ie/ Retrieved 25 August, 2008.

[6] Fu, B. 'Trust Management in Online Social Networks', Master's Thesis. Trinity College, Dublin. 2007.

[7] Acquisti, A. and Gross, R. 'Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook', Proceedings of Privacy Enhancing Technologies Workshop (PET). 2006.

[8] Acquisti, A. and Gross, R. 'Information revelation and privacy in online social networks', WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society. Alexandria, VA, USA. 2005.

[9] boyd, d. 'Reflections on Friendster, Trust and Intimacy', Ubiquitous Computing (Ubicomp 2003), Workshop application for the Intimate Ubiquitous Computing Workshop. Seattle, WA, USA. 2003.

[10] Ellison, N., Lampe, C., and Steinfield, C. 'Spatially Bounded Online Social Networks and Social Capital: The Role of Facebook', International Communication Association. Dresden. 2006.

[11] YouTube. http://www.youtube.com. Retrieved 25 August, 2008.

[12] LinkedIn. http://www.linkedin.com. Retrieved 25 August, 2008.

[13] Joinson, A. 'Looking at, looking up or keeping up with people?: motives and use of facebook', CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. Florence, Italy. 2008.

[14] Richter, H. and Strater, K. 'Examining privacy and disclosure in a social networking community', SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security. Pittsburgh, Pennsylvania, USA. 2007.

[15] Sophos Labs. 'Facebook privacy breach exposed users' hidden dates of birth', http://www.sophos.com/pressoffice/news/articles/2008/07/facebook-birthday.html. 16 July, 2008. Retrieved 5 August, 2008.

[16] Facebook Developer Platform. http://developers.facebook.com. Retrieved 25 August, 2008.

[17] Corritore, C. L., Kracher, B., and Wiedenbeck, S. 'An Overview of Trust', http://cobacourses.creighton.edu/trust/articles/trustpaper2-9-01_final.rtf . 2006. Retrieved 25 August, 2008.

[18] Marsh, S. 'Formalising trust as a computational concept', PhD Thesis. University of Stirling. 1994.

[19] Page, L., Brin, S., Motwani, R., Winograd, T. 'The PageRank Citation Ranking: Bringing Order to the Web', Technical Report. Stanford Digital Library Technologies Project. November, 1999.

[20] Kamvar, S.D., Schlosser, M.T., and Garcia-Molina, H. 'The EigenTrust Algorithm for Reputation Management in P2P Networks', In Proceedings International WWW Conference. Budapest, Hungary. 2003.

[21] Douceur, J. R. 'The Sybil Attack', 1st International Workshop on Peer-to-Peer Systems. Cambridge, MA, USA. 2002.

[22] O'Donovan, J. and Smith, B. 'Is Trust Robust? An Analysis of Trust-Based Recommendation', IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces. Sydney, Australia. 2006.

[23] Golbeck, J. 'Computing and Applying Trust in Web-Based Social Networks', PhD Thesis. University of Maryland. 2005.

[24] FOAF Project. http://www.foaf-project.org/ Retrieved 25 August, 2008.

[25] Ziegler, C. N. and Lausen, G. 'Towards Decentralized Recommender Systems', PhD Thesis. University of Freiburg. 2005.

[26] Advogato.org. http://www.advogato.org. Retrieved 25 August, 2008.

[27] Levien, R., Aiken, A. 'Attack resistant trust metrics for public key certification', 7th USENIX Security Symposium, San Antonio, Texas. January 1998.

[28] Epinions. http://www.epinions.com/ Retrieved 25 August, 2008.

[29] FilmTrust. http://trust.mindswap.org/FilmTrust/ Retrieved 25 August, 2008.

[30] Zend Framework. http://framework.zend.com/ Retrieved 25 August, 2008.

[31] OpenSocial. http://code.google.com/apis/opensocial/ Retrieved 25 August, 2008.

Appendix A

User Surveys

The following surveys were used to evaluate Facebook users' attitudes and behaviours towards privacy and trust, as well as their opinions about the features of the trust model and trust management application.

The results of each survey are on the DVD accompanying this dissertation.

A.1 Survey One

A.1.1 Section One: Demographics and Privacy Attitude

1. What age are you?

2. What country do you live in?

3. If you have a college degree, what field is it in?

4. How regularly do you visit Facebook?

Daily

Weekly

Monthly

5. How important is maintaining the privacy of your Facebook profile information to you?

With regard to friends?

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

With regard to strangers?

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

A.1.2 Section Two: Trust Scenarios

1. You receive a friend request from someone you are friends with in real life, and you must decide whether or not to accept the request. How important are each of the following trust characteristics to you in this situation?

Competence

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Confidence

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Credibility

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Faith

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Honesty

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Reliability

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Reputation

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

2. You receive a friend request from a person you don't know, but who shares mutual friends with you, and you must decide whether or not to accept the request. How important are each of the following trust characteristics to you in this situation?

Competence

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Confidence

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Credibility

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Faith

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Honesty

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Reliability

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Reputation

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

3. You want to buy something on the Facebook Marketplace, but the seller is a stranger. How important are each of the following trust characteristics to you in this situation?

Competence

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Confidence

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Credibility

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Faith

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Honesty

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Reliability

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

Reputation

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

A.1.3 Section Three: Additional Feedback

1. Additional Feedback

A.2 Survey Two

A.2.1 Section One: Demographics

1. What age are you?

2. What gender are you?

3. If you have a college degree, what field is it in?

A.2.2 Section Two: Facebook Privacy

1. How often do you use Facebook?

Never

Monthly

Weekly

Daily

Multiple Times Daily

2. How important do you feel the privacy of your information is on Facebook?

With regard to friends?

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

With regard to strangers?

Not Important

Below Average Importance

Average Importance

Above Average Importance

Very Important

3. Which of the following best matches your experience with Facebook's privacy controls?

I didn't know Facebook had privacy controls

I knew about them, but never used them

I knew about them, but they are too complicated

I use the privacy controls to protect my information

Other (please specify)

A.2.3 Trust and Privacy

1. When deciding whom to allow access to your Facebook profile, is trust a factor?

Yes

No

Why/Why Not?

2. If Facebook allowed you to represent the amount that you trust your friends and generate privacy settings from those trust values, would you find this useful?

Yes

No

3. Have you tried the Entrust application for Facebook? (http://apps.facebook.com/entrust)

Yes

No

4. If you answered yes to the previous question, did you think the application...

was useful?

Yes

No

was easy to use?

Yes

No

would be a good way to integrate trust and privacy on Facebook?

Yes

No