

Trust Management in Online Social Networks

Bo Fu

A dissertation submitted to the
University of Dublin, Trinity College
in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science

Submitted September 2007

DECLARATION

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Bo Fu
14th September 2007

PERMISSION TO LEND AND/OR COPY

I agree that Trinity College Library may lend or copy this dissertation upon request.

Bo Fu

14th September 2007

ACKNOLEGEMENTS

A loving heart is the truest wisdom.

~ Charles Dickens

I would like to thank many people for helping me during my M.Sc. research work. First, and foremost, I would like to deeply thank my supervisor, Dr. Declan O'Sullivan for his guidance, words of encouragement, kindness and patience. I shall never forget his support in the past months, without his supervision, there would not have been the steady progress in my research work.

Thanks also to the members of KDEG research group for providing opportunities to collaborate and learn. In particular, special thanks to Dr. Karl Quinn for his assistance and advice.

Thank you to all my NDS colleagues for allowing me taking up their valuable time with countless discussions on trust and online social networks, and special notes to Andrew Carton and Gábor Bernáth for sharing your knowledge and ideas.

Thanks to everyone who took part in the survey and interviews, without you, I would not have been able to collect precious data for this research work.

I would also like to express my appreciation to all my wonderful friends who provided me with love, caring, and inspiration during difficult times.

Finally, I am so very grateful for the family I am blessed with. To my Mum and Dad, Wang Shulan and Fu Zhongwei, who put a lifetime of effort and love into my upbringing. Thank you for everything you have done for me.

ABSTRACT

Since the launch of the first online social networking website *USENET* [Usenet] in 1979, we have seen a dramatic increase of online social networks in recent years that allow users to discover, extend, manage, and leverage their personal as well as professional networks online. Business-oriented online social network *LinkedIn* [linkedIn] adds more than ten new members every minute [Reid and Gray, 2007]; while *MySpace* [MySpace] with 47.3 million members since its launch in January 2004, adds more than one hundred and fifty new members every minute [Kawamoto & Sandoval].

Like most new technologies, such a young and exciting online social networking phenomena with rapidly growing communities welcomes innovation.

Trust, with broad definitions and concepts, somehow, works mysteriously. For many years, in various disciplines such as psychology, philosophy and sociology, we have tried to understand what trust means and how it works.

Described [Shneiderman, 2000], as “ancient social traditions were designed to elicit trust during uncertain encounters, handshaking demonstrated the absence of weapons; clinking of glasses evolved from pouring wine back and forth to prove it was not poisoned”, what can users of online social networks depend upon?

We must facilitate trust into online social networking websites, in order to provide users with the necessary tools and much desired freedom to express their subjective views of trust just like in the offline world.

This thesis reviews the state of the art in trust and trust mechanisms employed in online social networks, applies a multi-faceted model of trust that is personalisable and specialisable which aims to capture subjective views on trust of a broad population in a real world application.

Also, we describe the design and implementation of a small scale online social network with particular emphasis on its trust management approach. We then evaluate the proposed trust management solution, by comparing the performance of an online social network, namely *miniOSN*, with such an integrated trust mechanism to that of another popular online social networking website in created scenarios.

We demonstrate that the integrated trust management solution does provide personalisation for trust in *miniOSN*, where users can tailor their unique views on trust for certain individuals in different contexts. Finally, we point out the limitations of the proposed approach, and conclude with further directions for research in the area of trust management in online social networks.

TABLE OF CONTENTS

DECLARATION	ii
PERMISSION TO LEND AND/OR COPY	iii
ACKNOLEGEMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xii
CHAPTER ONE: INTRODUCTION	1
1.1 Motivation.....	1
1.2 Research Question	2
1.3 Objectives and Goals	2
1.4 Project Approach	3
1.5 Contribution	3
1.6 Overview of Thesis	3
CHAPTER TWO: STATE OF THE ART	5
2.1 Introduction.....	5
2.2 Social Networks and Online Social Networks.....	5
2.3 Online Social Networks – A Brief History	6
2.4 Categorisation of Online Social Networks	10
2.5 Trust	11
2.5.1 Definition	11
2.5.2 Characteristics of Trust.....	12
2.6 Trust Mechanisms in Online Social Networks	13
2.6.1 Current methodology	13
2.6.2 Current Issues.....	14
2.7 Related Work	16
2.8 Summary	18
CHAPTER THREE: OSN USER BEHAVIOUR AND ATTITUDE	
SURVEY	19
3.1 Introduction.....	19
3.2 Questionnaire Design.....	19
3.3 Questionnaire Execution.....	20

3.4 Questionnaire Participants	20
3.5 Survey Findings	22
3.5.1 Category One – Active OSN users	22
3.5.2 Category Two – No Longer Active OSN users	24
3.5.3 Category Three – Not Users of OSNs as yet	26
3.5.4 Desired Trust Features and Opinions on the Proposed Solution ...	27
3.6 Survey Analysis	29
3.7 Summary	30
CHAPTER FOUR: DESIGN	31
4.1 Introduction.....	31
4.2 Influences from a Multi-faceted Model of Trust that is Personalisable and Specialisable.....	31
4.3 Influences from OSN User Behaviour and Attitude Survey.....	32
4.4 <i>MiniOSN</i> and Its Trust Mechanism.....	33
4.5 Summary	43
CHAPTER FIVE: IMPLEMENTATION	44
5.1 Introduction.....	44
5.2 Ruby on Rails.....	44
5.2.1 Design Principles	45
5.2.2 MVC Architecture	45
5.2.3 Metaprogramming.....	51
5.2.4 Test Driven Development	51
5.2.5 Rails Plugins	52
5.3 Building <i>MiniOSN</i> with Ruby on Rails	52
5.3.1 RESTful Rails Routes	54
5.3.2 The <code>Attachment_fu</code> Plugin, <i>ImageMagick</i> and <i>RMagick</i> Libraries	54
5.3.3 XHTML Friends Network Microformat.....	58
5.4 <i>miniOSN</i> Architecture.....	61
5.5 Implementation Difficulties	62
5.6 Summary	62
CHAPTER SIX: EVALUATION.....	63
6.1 Introduction.....	63

6.2 Scenario Description.....	63
6.2.1 Scenario One.....	64
6.2.2 Scenario Two.....	64
6.3 Comparison of <i>miniOSN</i> and <i>Bebo</i> in the Given Scenarios.....	65
6.4 Evaluation Interviews.....	66
6.4.1 Volunteer Background.....	67
6.4.2 Results.....	69
6.4.3 Analysis.....	69
6.5 Summary.....	70
CHAPTER SEVEN: CONCLUSIONS.....	72
7.1 Project Summary.....	72
7.2 Contribution.....	73
7.3 Future Work.....	74
7.4 Final Remarks.....	77
BIBLIOGRAHPHY.....	79
APPENDICES.....	83
APPENDIX I – <i>A Survey of Online Social Networks</i>	83
APPENDIX II – Evaluation Questionnaire.....	90

LIST OF FIGURES

Figure 2- 1: Trust Calculation Overall Framework	17
Figure 3- 1: <i>A Survey of Online Social Networks</i> Participant Background	21
Figure 3- 2: Participant Experience with Notable OSNs	22
Figure 3- 3: Access settings of user profiles – Category One.....	22
Figure 3- 4: User satisfaction towards current access control methods.....	23
Figure 3- 5: Would you trust random strangers to view your profile?	24
Figure 3- 6: Is it necessary that only certain people can view certain parts of your profile?.....	24
Figure 3- 7: Access settings of user profiles – Category Two.....	25
Figure 3- 8: Why did you stop using OSNs?	25
Figure 3- 9: Why have you never used OSNs?.....	26
Figure 3- 10: Do you trust all your connected friends with all parts of your profile?.....	27
Figure 3- 11: Views on the eight attributes of trust	28
Figure 3- 12: Would you rate how much you trust your friends in OSNs?	28
Figure 4- 1: Screenshot of the Signup Page.....	34
Figure 4- 2: Screenshot of the Edit Account Page.....	34
Figure 4- 3: Screenshot of a User Profile.....	35
Figure 4- 4: Screenshot of a User Blog and Comments by Others.....	35
Figure 4- 5: Rachel’s Connected Friends.....	36
Figure 4- 6: Screenshot of Editing a Friendship	37
Figure 4- 7: Screenshot of Setting Trust Requirements for a Blog.....	38
Figure 4- 8: Rachel’s Current Trust Ratings for Ross	39
Figure 4- 9: Chandler’s Current Trust Ratings for Ross.....	39
Figure 4- 10: Ross’s Current Trust Ratings for Rachel	40
Figure 4- 11: Screenshot of the Photo Upload Page	41
Figure 4- 12: Setting Trust Rating Requirements for an Uploaded Photo.....	41
Figure 4- 13: Setting Trust Values for a Friend and Friends of the Friend	42
Figure 5 - 1: The MVC Architecture	46
Figure 5 - 2: Rails and MVC.....	47
Figure 5 - 3: The User Table Migration File Snippet	48
Figure 5 - 4: The User Model File Snippet.....	49

Figure 5 - 5: The Users Controller File Snippet	50
Figure 5 - 6: The New User Sign Up View Snippet	51
Figure 5 - 7: The Login View Snippet.....	51
Figure 5 - 8: <i>miniOSN</i> File Directory	53
Figure 5 - 9: The Photos Table Migration Snippet	56
Figure 5 - 10: The Photo Model Snippet	57
Figure 5 - 11: The Friendships Table Database Snippet.....	59
Figure 5 - 12: The Friendship Model Snippet.....	60
Figure 5 - 13: <i>miniOSN</i> Architecture	61

LIST OF TABLES

Table 2- 1: Brief Timeline of Online Social Networking	8
Table 2- 2: Most Popular Websites in July, 2007 – Based on Market Share of Visits	9
Table 2- 3: Top Twenty OSNs in September, 2006.....	10

CHAPTER ONE: INTRODUCTION

By seeking and blundering we learn.

~ Johann Wolfgang Von Goethe

1.1 Motivation

In recent years, we have seen dramatic increases and growing popularity of online social networks (OSNs) such as *MySpace* [MySpace] and *Facebook* [Facebook]. As OSNs mature, issues that centre around proper use of such networks are also growing and making headlines. We feel that the area of trust management for OSNs is of increasing importance, especially given the exponential growth of online communities.

Trust, has been heavily studied by researchers in psychology, philosophy and sociology; research in these fields show that trust is a subjective view that varies greatly among people, situations and environment. However, this very subjectivity of trust has been overlooked in OSNs. At the moment, trust in OSNs is expressed by using simple access control methods which all tend to take a very simplified view of trust and use a one-size-fits-all approach for all users. Trust cannot be expressed subjectively, nor can it be personalised. We feel there is the strong need for an innovative design for the model of trust in OSNs that enables users to make annotations of trust freely and confidently.

Much research has been carried out in the field of computer science in relation to trust management, various algorithms, systems and models have been produced, such as PGP [Zimmerman, 1995], REFEREE [Chu et al, 1997], SULTAN [Grandison et al, 2001], FOAF [Dumbill et al, 2002], TRELIS [Gil et al, 2002], Jøsang's trust model [Jøsang A., 1996], Marsh's trust model [Marsh, 1994] and many more. In particular, a multi-faceted model of trust that is personalisable and specialisable [Quinn, 2006] has been designed in the Knowledge and Data

Engineering Group (KDEG) from the Computer Science Department in Trinity College Dublin.

Built on a large amount of literature review and findings from the state of the art studies, the multi-faceted model of trust aims to portray the broad subjective views on trust. Much like how trust is viewed and expressed in the real world, the model allows users of such a trust management system personalise their trust for certain individuals in the virtual environment.

1.2 Research Question

Motivated by addressing current issues such as a lack of personalisation in trust management approaches in OSNs, this research aims to answer the questions of whether a multi-faceted model of trust that is personalisable and specialisable be welcomed in OSNs, would an application of the model satisfy user needs when expressing their subjective views on trust in the OSN environment, and most of all, would the proposed solution address issues we found related to the state of the art.

1.3 Objectives and Goals

The main aim of this thesis is to find out how well can a multi-faceted model of trust that is personalisable and specialisable capture a variety of subjective views on trust for OSN users.

In order to undertake the research, the following goals were derived:

- Research the state of the art in current trust mechanisms employed in OSNs, in order to identify issues associated with these trust management approaches.
- Research the state of the art in trust, focusing on the characteristics of trust that remain true regardless how trust is modeled.
- Find out whether OSN users are happy with current trust mechanisms used in notable OSNs, in order to determine whether there is the need for implementing an innovative system.

- Design and build a small scale online social network, *miniOSN*, which applies the multi-faceted model of trust that is personalisable and specialisable.
- Simulate scenarios and compare the performance of *miniOSN* to that of a popular online social network in the aforementioned scenarios.
- Evaluate the trust management approach in *miniOSN* by gathering user opinions.

1.4 Project Approach

We first studied trust and its characteristics, reviewed the state of the art in trust management mechanisms deployed in notable OSNs, and identified current issues as well as problems associated with these methods in use.

We then researched user behaviours in OSNs and whether current trust management systems satisfy user needs.

To address issues found in the literature review, we proposed a solution that uses the multi-faceted model of trust that is personalisable and specialisable.

We designed and built an OSN that is powered by Ruby on Rails (RoR) with the application of the proposed trust model.

And finally, we evaluated our work with scenarios and interviews.

1.5 Contribution

This thesis describes the design, the implementation and the evaluation of a small scale online social network: *miniOSN*, the trust management system employed in this OSN is strongly influenced by the multi-faceted model of trust that is personalisable and specialisable.

Also, we show how well received such an approach could be in the online community as well as the limitations of the model.

And finally, we identify possible future research work in the topic area.

1.6 Overview of Thesis

This thesis is organised as follows.

- Chapter Two first introduces the concept of Online Social Networks, the categorisations used as well as a brief history of them. It then provides a review and analysis of the state of the art in trust and its characteristics, and most of all, the current trust mechanisms used in notable online social networks.
- Chapter Three concentrates on a survey designed to gather user opinions of current trust management approaches being used, and presents our findings as well as analysis of the results.
- Chapter Four describes the design of a small scale online social network named *miniOSN* and the trust management approach it takes.
- Chapter Five talks about the implementation of *miniOSN* and the technologies we use as well as difficulties encountered.
- Chapter Six presents the evaluation of *miniOSN*, with particular emphasis on its trust management system.

And finally, we conclude in Chapter Seven, discussing the extent to which the original objectives and goals were achieved during this research project, as well as identifying several possibilities for future work in the research of trust management in online social networks.

CHAPTER TWO: STATE OF THE ART

Love all, but trust a few.

~ William Shakespeare

2.1 Introduction

This chapter first introduces the concepts of social networks and online social networks in section 2.2, we then provide a brief history of online social networking in section 2.3. Categorizations of online social networks are discussed in section 2.4, followed by literature review of trust and trust mechanisms used in online social networks in section 2.5 and 2.6. We then discuss related work and present the multi-faceted model of trust that is personalisable and specialisable in section 2.7, and finally, section 2.8 provides a summary of the chapter.

2.2 Social Networks and Online Social Networks

The term, **social network**, was first coined by Professor J. A. Barnes in the 1960s [Barnes, 1967], describing associations of people drawn together by family, work, hobby, etc.; for support such as emotional, instrumental, appraisal and information. Such networks operate on many levels, from the family level up to as high as the level of nations; and play important roles in communications among people, organizations and nations; as well as the way how problems are solved and how organizations are run.

In its simplest form, a social network is a map of the relevant ties between the individuals, organizations, nations, etc., being studied.

Much like social networks, through the media of the Internet, **Online Social Networks (OSNs)** have many similar characteristics and in the digital age, they also play important roles in communication among individuals and organizations.

In recent years, we have seen a dramatic increase of online social networks such as *Bebo* [Bebo], *Facebook* and *MySpace* just to name a few; where one can set up a profile about oneself, invite friends to join the site and link these people together with the purpose of sharing information and resources.

2.3 Online Social Networks – A Brief History

The concept of social networking dates back to 1930s, when Vannevar Bush first introduced his idea about “memex” [Vannevar, 1996], a “device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility”, and predicted that “wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified.”

The first online social networks were called *USENET* newsgroups [Usenet], designed and built by Duke University graduate students Tom Truscott and Jim Ellis in 1979. In its simplest form, *USENET* represents democracy. The basic element of *USENET* is a post, each individual post consists of a unique contribution from some user placed in a subject area, called a newsgroup [Hauben & Hauben, 2004].

Table 2-1 below shows a brief timeline of the history of online social networking [Teten & Allen, 2005, p.42].

1971	Ray Tomlinson invents email.
1973	First group chat program.
1975	First mailing list, called <i>MsgGroup</i> . First computer conferencing system.
1978	First Multi-User Dungeon (MUD) for multi-user gaming.
1979	<i>USENET</i> newsgroups created.
1984	Birth of the <i>Fido</i> network of Bulletin Board Systems (BBSes).
1985	Whole Earth Letronic Link (WELL) community begins.
1988	Internet Relay Chat (IRC) invented.
1991	Tim Berners-Lee posts “World-Wide Web: Executive Summary” to <i>USENET</i> Group.

	“Gopher”, the first simple menu-driven client to Internet resources launches.
1992	Berners-Lee creates his “What’s New?” page, arguably the first blog.
1993	Howard Rheingold publishes <i>The Virtual Community</i> . Mosaic Web browser is released.
1994	“Christ is coming” is the first spam on <i>USENET</i> .
1995	Ward Cunningham launches the first wiki. <i>AltaVista</i> , the first full Web search engine, launches.
1996	ICQ: first peer-to-peer instant messaging appears. January: 100,000 Web servers
1997	April: 1,000,000 Web servers. <i>Slashdot</i> , the first blog to enable reader comments, goes online. Jorn Barger coins the term “Weblog.” <i>SixDegrees.com</i> , first site based on the “six degrees of separation” concept, launches.
1998	Open Directory Project (DMOZ), later acquired by <i>Netscape</i> .
1999	Peter Merholz coins the term “blog” as a contraction of “Weblog.” <i>LiveJournal</i> and <i>Blogger</i> launch. <i>Kuro5hin</i> , a blog where users vote for what goes to the front page, launches. <i>Napster</i> launches.
2000	<i>HotOrNot.com</i> created with zero capital
2001	<i>Wikipedia</i> , an open collaborative wiki encyclopedia project, goes live. Movable Type (leading blog software) initial beta release. <i>Ryze</i> social network service launches.
2002	10,000,000 th Web server goes live. 10,000,000 th post on <i>Blogger</i> . <i>Friendster</i> launches.
2003	Venture capital investment in social network space exceeds \$50 million. <i>Wikipedia</i> hits 100,000 articles. Howard Dean campaign uses blog and <i>Meetup</i> to organize more than 100,000 supporters. <i>LiveJournal</i> and <i>Friendster</i> pass 1 million accounts. <i>Skype</i> released. <i>LinkedIn</i> , social network focused on business professionals, secures Series A financing of \$4.7 million led by Sequoia Capital. <i>MySpace</i> , social network focused on music and entertainment, launches.
2004	<i>Skype</i> hits 10 million downloads.

	Social Networking <i>Metalist</i> (SocialSoftware.BlogsInc.com) lists more than 200 different social networking systems.
2005	<i>Skype</i> hits 100 million downloads.
2006	<i>Google</i> acquires <i>YouTube</i> , video social network, for a stock transaction worth \$1.65 billion.
2007	IBM launches enterprise social networking suite. <i>LinkedIn</i> surpasses 10,000,000 members. Germany social networking site <i>OpenBC/Xing</i> successful IPO. <i>Wikipedia</i> exceeds 1,700,000 English articles. <i>Technorati</i> indexes more than 80 million blogs.

Table 2- 1: Brief Timeline of Online Social Networking

Since its first launch in the late 70s, in most recent memory, we have seen significant growth of OSNs. To date, there are hundreds of online social networking sites in Europe alone. Studies [Hitwise Data Centre, 2007] by *Hitwise* [Hitwise] have shown that in July, 2007, ranked by market share of visits across all industries, the most popular websites based on U.S. Internet usage, *MySpace* was the most visited website. Among the top twenty most popular websites, online social networks took significant places, as Table 2-2 below shows.

Rank	Website	Market Share
1	www.myspace.com	6.33%
2	www.google.com	4.8%
3	mail.yahoo.com	4.47%
4	mail.myspace.com	4.02%
5	www.yahoo.com	3.94%
6	www.hotmail.com	1.64%
7	www.ebay.com	1.57%
8	search.yahoo.com	1.56%
9	www.msn.com	1.48%
10	www.facebook.com	1.03%
11	www.youtube.com	0.77%
12	search.msn.com	0.62%
13	mail.live.com	0.51%
14	images.google.com	0.47%
15	blog.myspace.com	0.41%

16	www.gmail.com	0.4%
17	www.wikipedia.org	0.37%
18	music.myspace.com	0.36%
19	mail.aol.com	0.34%
20	my.yahoo.com	0.32%

Table 2- 2: Most Popular Websites in July, 2007 – Based on Market Share of Visits

Another study done by Hitwise shows that in the month of September 2006, one out of every twenty U.S. Internet visits landed on one of the top twenty social networking websites [Hitwise Data Centre, 2006]. Out of the twenty online social networking websites, *MySpace* is the undisputed leader, receiving 81.92% of those visits. Other online social networking sites with above-average growth in market share of visits include *Bolt*, *Bebo*, *Orkut*, and *Gaia Online*. Table 2-3 shows the entire list of the top twenty websites in the study.

Rank	Name	Domain	Market Share	Average Session Time
1	MySpace	www.myspace.com	81.92%	30:22
2	Facebook	www.facebook.com	7.24%	8:17
3	Xanga	www.xanga.com	1.86%	12:00
4	Yahoo! 360	360.yahoo.com	1.21%	11:42
5	BlackPlanet.com	www.blackplanet.com	1.12%	20:19
6	Bebo	www.bebo.com	1.02%	25:39
7	Classmates.com	www.classmates.com	0.85%	7:14
8	LiveJournal	www.livejournal.com	0.76%	12:27
9	Hi5	www.hi5.com	0.62%	14:31
10	Tagged	www.tagged.com	0.58%	20:33
11	Gaia Online	www.gaiaonline.com	0.51%	47:01
12	Sconex	www.sconex.com	0.42%	18:20
13	Friendster	www.friendster.com	0.42%	19:52
14	Bolt.com	www.bolt.com	0.36%	1:43
15	Windows Live Spaces	spaces.live.com	0.31%	4:37
16	Orkut	www.orkut.com	0.30%	21:33
17	myYearbook	www.myyearbook.com	0.17%	8:56
18	CrushSpot.com	www.crushspot.com	0.14%	30:31

19	miGente.com	www.migente.com	0.13%	14:01
20	Piczo	www.piczo.com	0.07%	6:04

Table 2- 3: Top Twenty OSNs in September, 2006

2.4 Categorisation of Online Social Networks

By their served purposes, OSNs mostly can be put into categories that centre around the following topics: business, education, socializing and entertainment.

Business oriented OSNs help registered individuals make connections, build business contacts and maintain professional networks for potential career opportunities; as well as allowing organizations to advertise their products and services. Examples of such OSNs are *LinkedIn* [LinkedIn], *Ecademy* [Ecademy], *Doostang* [Doostang], *XING* [XING] and *Plaxo* [Plaxo].

Educational OSNs usually focus on groups of people who wish to gain knowledge in the same field mostly through the forms of blogs and link sharing with a great variety of subject matter. Examples of such networks can be found in many institutions, where intranets are set up for specific schools, faculties, or classes.

Socializing OSNs aim to provide users with a virtual environment in which online communities can exchange news, keep in touch with friends and family, and make new connections. Usually, various features are implemented which allow users to keep journals, post comments and news, upload pictures and videos as well as send each other messages. Such OSNs tend to centre around themes, such as music, movies, resource sharing, personal life, etc., and are designed to be either user-centric or topic-centric, where online communities can focus on developing profiles all about oneself or developing particular hobbies. Several examples of this type of OSNs are *43 Things* [43Things], *CarDomain* [CarDomain], *Friendster* [Friendster], *Hi5* [Hi5], and *MOG* [MOG].

Closely associated with socializing OSNs are entertaining OSNs, where focuses on personal aspects of the online communities are less visible, compared to the entertainment attributes these communities may offer to the network. For

example, on *YouTube* [YouTube], focus is shifted away from personal profiles, and the video sharing feature is greatly valued. Since its launch in early 2005, *YouTube* has quickly become the home of video clip entertainment, it now accounts for 29% of the U.S. multimedia entertainment market [USA Today, 2006].

By registration requirements, OSNs can be grouped into two main categories, sites that are open to anyone and sites that are invitation only. Usually, anyone is welcomed to set up an account and put up a representation of oneself in open-invite OSNs, such as *Graduates.com* [Graduates], *Friends Reunited* [Friends Reunited] and many more. However, in some sites, in order to join, you need to be invited by a trusted member, *aSmallWorld* [aSmallWorld] is an example of such OSNs where registered members include celebrities like Naomi Campbell.

The predominant business model for most OSNs is advertising. It is free for anyone to join, and revenue is made by selling online advertising on these websites. However, a number of OSNs charge their members for the information or services they provide, such as *LinkedIn* where employers can advertise their vacancies looking for suitable candidates.

2.5 Trust

Trust, has been significantly studied in psychology, philosophy, sociology, as well as computer science. It is an elusive notion that is hard to define, since the term “trust” stands for a diversity of concepts depending on the person you ask.

Section 2.5.1 presents a collection of the definitions of trust, and section 2.5.2 discusses properties of trust.

2.5.1 Definition

To some, trust is predictability, where evidence of one’s reputation suggests a most-likely outcome; to others, trust is dependability, where one truly believes in another and accepts not understanding some things; yet, to many, trust is letting

others make decisions for you and knowing that they would act in your best interest.

Trust has many rich meanings in different contexts and stands for a variety of concepts depending on the person you approach.

So what is trust? Several notable definitions of trust are presented below.

Mui et al. [Mui et al., 2002] defined trust as “a subjective expectation an agent has about another’s future behaviour based on the history of their encounters.”

Grandison and Sloman [Grandison & Sloman, 2000] introduced context and defined trust as “the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context.”

Olmedilla et al. [Olmedilla et al., 2005] stated that “Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X).”

In summary, trust can not be defined by a single consensus, there is a wide and varied range of synonyms for trust, and the answer to “what is trust” can not be easily provided. Hence, significant challenges are presented for modeling trust in the semantic Web, therefore, it is important for us to concentrate on the core characteristics of trust as discussed in the next section.

2.5.2 Characteristics of Trust

Trust has several main characteristics [Golbeck, 2005; Dey, 2001] that remain true regardless how trust is defined, as discussed below:

Trust is Asymmetric.

Between two parties, trust level is not identical. A may trust B 100%, however, B may not necessarily feel the same way about A; B may only trust A 50% for example.

Arguably, trust can be transitive.

Let's say that A and B know each other very well and are best friends, B has a friend named C whom A has not met. But since A knows B so well and trusts B's choices in making friends, A may trust C to a certain extent even though they have never met. Now let's say C has a friend named D whom neither A nor B knows well, A could find it hard to trust D. Hence, some argue that as the link between nodes grow longer, trust level decreases.

However, others [Grandison, 2003; Abdul-Rahman, 2004] disagree with the statement that trust is transitive, [Zinnermann, 1994] states that if I have a good friend whom I trust dearly, who also trusts that the president would not lie, does that mean that I would therefore trust that the president would not lie either?

Trust is personalised.

Trust is a subjective point of view, two parties can have very different opinions about the trustworthiness of the same person. For example, a nation may be divided into groups who strongly support the political party in charge and groups who would strongly disagree.

Trust is context-dependent.

Trust is closely associated with overall contexts, in other words, trust is context-specific [Gray, 2006]. One may trust another enough to lend that person a pencil, but may find the person hard to trust with a borrowed laptop for instance.

2.6 Trust Mechanisms in Online Social Networks

2.6.1 Current methodology

Current trust mechanisms used in OSNs have been limited to simple access control mechanisms, where authorization is required to contact, to write on, or to read all or part of a user's profile, given that blogging or commenting features are enabled. Communities in OSNs are usually categorized into groups such as, one's family, friends, neighbours, etc., with all or limited access to one's photos, blogs and other resources presented in one's profile.

To date, ways to control accesses to a person's profile are the only deployed trust mechanisms in OSNs, where users can block certain other members as well as granting various levels of access to specified groups of people.

In *Bebo* for instance, a user can get URL for his/her profile which then is viewable to anyone with a browser, or he/she can set the profile "private" which means that only the connected friends to this user are authorized to view the profile and everything presented in it.

In *Yahoo! 360°* [Yahoo!360], access control mechanism is refined by letting users set their profiles and blogs viewable to the general public, their friends, friends of their friends or just the users themselves. The site allows users the freedom to create specific friend categories, such as friends in work, friends met while traveling, etc. Users can then control whether to be contacted via email or messenger by anyone in the *Yahoo! 360°* network, people whom one is connected to, or only those in the defined categories.

In *Facebook*, privacy settings of a profile is further refined by allowing the owner of a profile grant different levels of access to sections of a profile such as contact information, groups, wall, photos, posted items, online status, and status updates. Also, users can decide whether they'd like the search engine to list them if a profile is searchable. As well as the freedom of deciding whether to notify friends with the latest activities, users can also select which parts of the profile are to be displayed to the person who tries to contact you through a poke, message, or friend request.

2.6.2 Current Issues

One-size-fits-all approach

Among a list of notable OSNs [Information Today database, 2007] we have found that controlling access seems to be the only way to express trust, where users group their connections into categories and grant all or limited access to these specified categories. Studies [Ralph, Alessandro et al. 2005] of *FaceBook* have

shown that many people who are connected to one person are not necessarily “friends” as such, but simply people whom that person does not dislike. Hence, there is a great variety of the levels of trust among these connected “friends” of a person. However, this variety of trust level has not been captured in OSNs, and users can not annotate their variety of trust in a person, nor can they personalise that trust depending on the situation. In certain situations we want private information to be known only by a small group of people and not by random strangers. Such information may be where you live, how much money you make, etc., in an OSN environment, you probably would dislike the idea of random strangers reading comments left by your friends detailing a trip you are about to take, for safety reasons. In other instances, we are willing to reveal personal information to anonymous strangers, but not to those who know us better. For example, if desired, one can state one’s sexuality on a profile page and broadcast that to the world, however, one may not be ready to reveal that very piece of information to the family and friends whom one trusts most.

The inflexible, one-size-fits-all approach cannot provide users the freedom to express their various levels of trust in a given network, therefore, fail to provide a tailored trust model for users in OSNs.

Lack of personalisation

As discussed earlier, there are several trust characteristics that remain true regardless how trust is modelled in the OSN scenario. These properties of trust have not been captured well in OSNs since current trust management approaches only employ simple access control methods.

If A is connected to B in *Bebo* for example, A can then view all other friends connected to B as long as these other friends acquire URLs for their profiles. The system presumes that trust is transitive, and as long as B is connected to A, B should trust A viewing profiles of all other friends of B’s. Also, overall contexts is overlooked, the fact that trust is context-dependent is not taken into account. Once a user has a public profile, the system assumes that all resources in this user’s profile, regardless whether it’s a picture, a video, a blog entry or comments left by others, are equally trusted to be accessed by a third party.

Many other OSNs that employ similar methods in controlling access to user profiles face the same problems, and this lack of personalisation when modelling

trust betray the fundamental characteristics of trust we found in the literature review.

Single-faceted approach

Currently, trust in OSNs is modelled as granting various levels of access controls to specified groups of users, there is no other option given. This single-faceted approach takes a simplified view on trust and fails to provide a variety of definitions for trust as we have found in the state of the art studies.

2.7 Related Work

Previously, a multi-faceted model of trust that is personalisable and specialisable was designed and developed by Quinn [Quinn, 2006] from the Knowledge and Data Engineering Group (KDEG) [KDEG] in the Department of Computer Science and Statistics, Trinity College Dublin.

While reviewing trust management systems in computer science, Quinn found that current methods “tend to use a single synonym, or definition in the use of trust... such approaches can only provide a generic, non-personalised trust management solution”. To address this problem of the lack of potential for personalizing trust management, a multi-faceted model of trust that is personalisable and specialisable was proposed, implemented and evaluated. In the proposed model, trust is divided into concrete concept and abstract concept with attributes of their own, where the former includes credibility, honesty, reliability, reputation and competency attributes, and the later with belief, faith and confidence attributes. Ratings are then given to each of the eight attributes, and trust is calculated as the weighed average of these ratings.

The overall structure of this *Model of Trust* that is personalisable and specialisable is separated across four models [Quinn, 2006, p.52], outlined in blue, as Figure 2-1 shows below [Quinn, 2006, p.50]. Outlined in red [Quinn, 2006, p.51], “the *myTrust Management Service* utilises a personalised model of trust, a domain specific model of trust, associated trust data, and trust policy to provide trust based recommendations to applications that operate in Internet environments.”

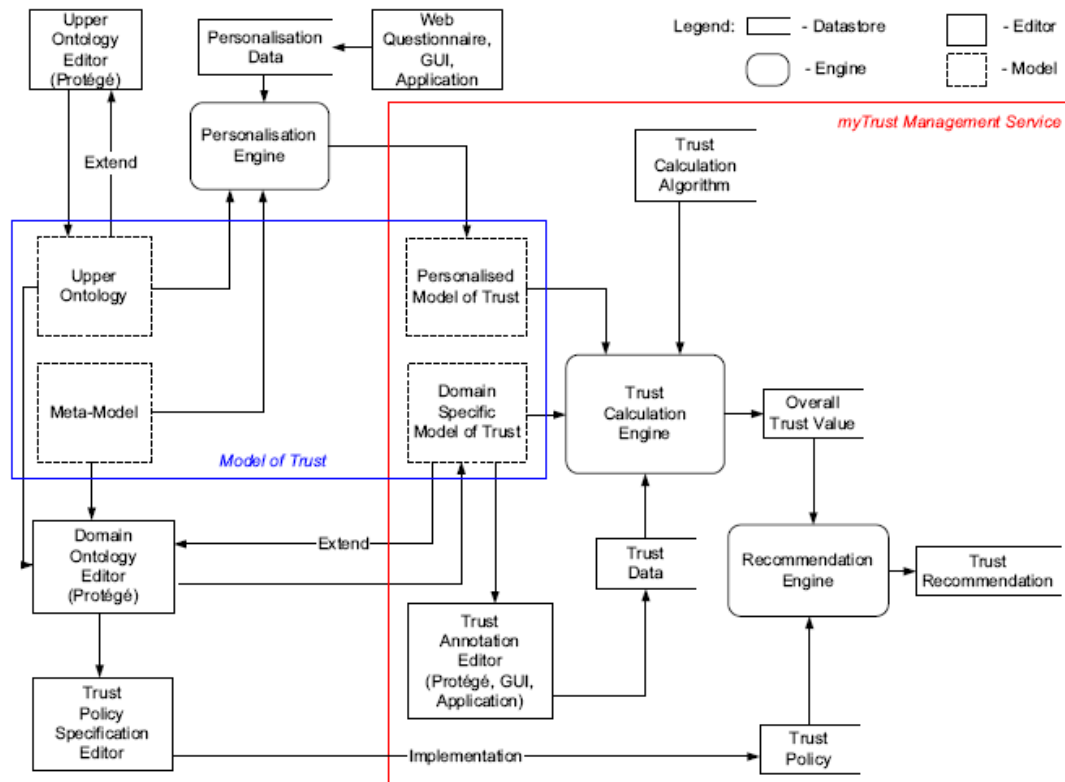


Figure 2- 1: Trust Calculation Overall Framework

Quinn specifies the following in his design:

The upper ontology provides a set of trust concepts that are used in the generation of personalised models of trust and are also used to engineer specialised models of trust. The relationships that can exist between the extensible set of trust concepts is governed by the trust meta-model...A domain specific model is the instantiation of the upper-model and meta-model towards a given application domain. In domain specialisation the trust concepts in the upper ontology are sub-classed and domain specific properties are added. Domain models are kept separate to allow developers to capture and scope a range of domains, which can be used independently in applications. Personalised models of trust are generated from the upper ontology and meta-model on a per user basis. A personalised model contains the set of relationships that may exist between trust concepts as provided by an individual [Quinn, 2006, p.52].

The claim for this model is that it has “the ability to capture an individual’s subjective views of trust, also, capture the variety of subjective views of trust that

are exhibited by individuals over a large and broad population”, which in turn, provides “a tailored and bespoke model of trust”. In addition to demonstrating its personalization capabilities, Quinn demonstrated how the model could be specialised to any application domain.

The two applications that were used to trial the model and approach were web services composition and access control in a ubiquitous computing environment. However, Quinn did speculate in his conclusions that the model would be suitable for use in the OSN domain.

2.8 Summary

This chapter discussed social networks, online social networks and the categorisations of them, presented a collection of definitions of trust, as well as main characteristics of trust. In addition, a review of current trust mechanisms used in notable OSNs was provided and we also identified issues associated with current trust management systems used in OSNs. And finally, we presented the foundation that this research is built upon – a multi-faceted model of trust that is personalisable and specialisable.

CHAPTER THREE: OSN USER BEHAVIOUR AND ATTITUDE SURVEY

Change your thoughts and you change your world.

~ Norman Vincent Peale

3.1 Introduction

This chapter concentrates on *A Survey of Online Social Networks*. First, the motivation behind such a survey is presented, and the design and the execution of the questionnaire are discussed in section 3.2 and section 3.3. We then give background information of participants in section 3.4. The findings are presented in section 3.5 followed by analysis of the results, discussed in section 3.6. Finally, the chapter is summarized in section 3.7.

Given the lack of flexible and personalised trust management features within notable OSNs and our belief that such features would be welcomed by users, we decided to explore with users whether the multi-faceted model of trust proposed by Quinn that enables personalization and the freedom of annotating trust subjectively would be welcomed in OSNs. Also, what would be the desired functionalities if such a trust management approach is to be integrated into OSNs? With these questions in mind, *A Survey of Online Social Networks* was designed. A complete list of the questions in this survey can be found in Appendix I.

3.2 Questionnaire Design

The questionnaire groups participants into three categories as follows, people who are currently using OSNs, people who have used OSNs in the past but are no longer active, and finally, people who have never used OSNs. With the former two categories, the survey aimed to find out user behaviour in relation to the trust management aspect in OSNs, and gather user experience with existing trust

mechanisms. With the last category, we aimed to find out why some have not or will not use OSNs. Most importantly, without excluding anyone, regardless of participants' experience with OSNs and current trust mechanisms, we ask for their desired trust features as well as their opinions on a proposed model of trust.

3.3 Questionnaire Execution

A trial questionnaire was first designed and road tested in a computer science postgraduate class, where a group of twelve people took part in the survey, which has helped the refinement of the official questionnaire.

Considering their flexibility, feasibility and easy data gathering factors, online questionnaires was convenient as we were aiming at a large audience, therefore, *SurveyMonkey* [SurveyMonkey] was chosen to host the survey on the 27th of May, 2007, over a period of two weeks time. Invitations to take part in the survey were sent out via email, to targeted third level institutions (including, Dublin City University [UCD], Dublin Institute of Technology [DIT], National University of Ireland, Galway [NUI, Galway], University College Cork [UCC], University College Dublin [UCD], Trinity College Dublin [TCD]) in Ireland, and interested parties were encouraged to distribute the questionnaire further.

3.4 Questionnaire Participants

In total, 393 people took part in answering the online questionnaire. Among which, 59% were male, 41% were female. Mostly in their early 20s, 68% of respondents were undergraduate students, 21% of postgraduate student and with the remaining being college employees. Most survey participants come from science related background, with a high 70% of people either studying for or having a degree in engineering, computer science or information technology related fields, as Figure 3-1 shows.

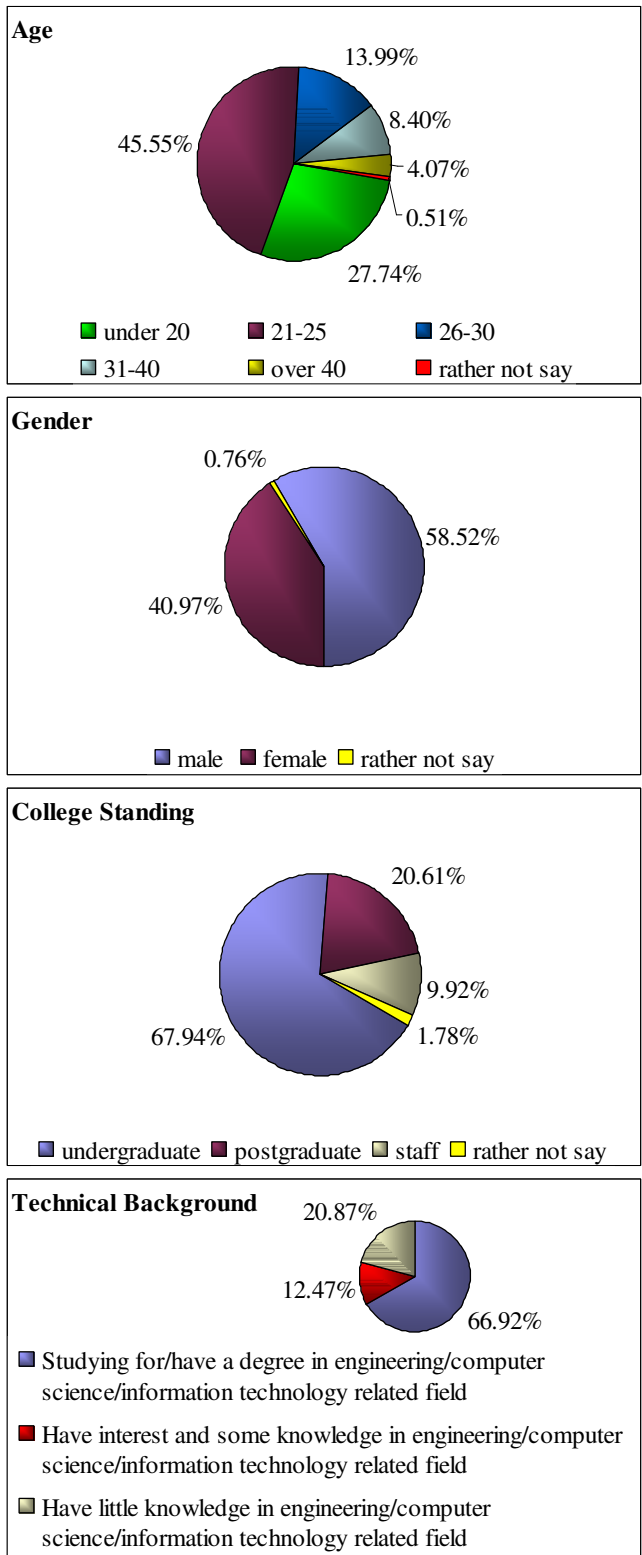


Figure 3- 1: A Survey of Online Social Networks Participant Background

3.5 Survey Findings

3.5.1 Category One – Active OSN users

Among 243 respondents who are currently using OSNs, *Bebo* is the most popular online social networking website with 87.90% of people have had experience with it, closely followed by *YouTube*, *MySpace* and *Facebook*, as Figure 3-2 shows.

The majority of the profiles are set to be viewable by the general public, while 20% of people allow only directly linked friends to view their profiles, as Figure 3-3 shows.

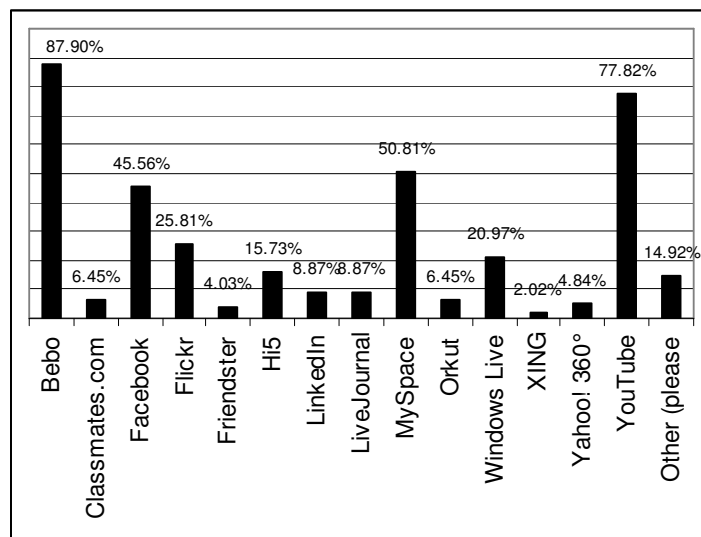


Figure 3- 2: Participant Experience with Notable OSNs

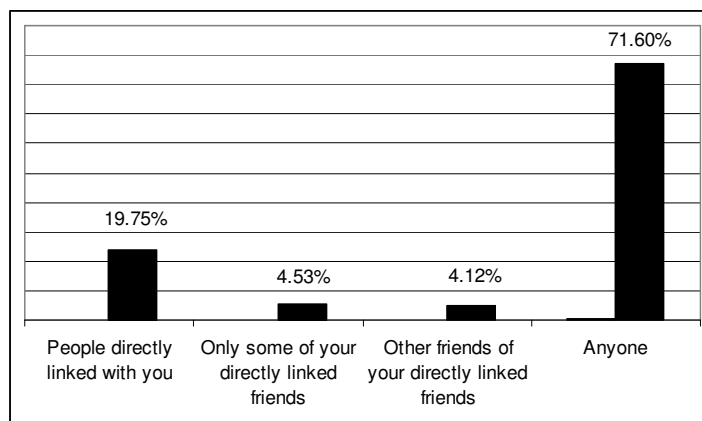


Figure 3- 3: Access settings of user profiles – Category One

We asked the question of whether these users are happy with the available ways of controlling access to their profiles. As Figure 3-4 shows, most people are pleased with current access control methods, while around 20% of the respondents are not concerned with it and less than 10% of people are not satisfied with it. Among reasons given for their unpleasant experience hence dissatisfaction, almost every comment of those 10% of people was related to the lack of better access controls to user profiles. For example, despite having a private profile, emails can still be sent to these people from others on the site.

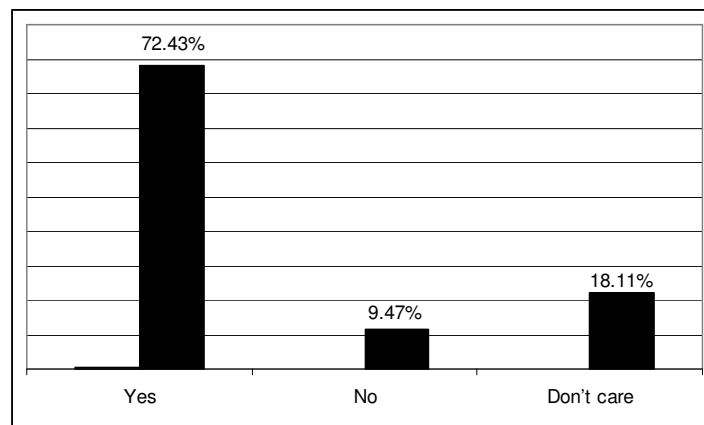


Figure 3- 4: User satisfaction towards current access control methods

Since the majority of this category has public profiles, we asked the question of whether they trust random strangers to view their profiles, as well as the question of whether access control really is necessary. As Figure 3-5 shows, despite having public viewable profiles, only 25% of these people actually stated the fact that indeed, they do trust anyone and everyone viewing their profiles. Most people however, claimed that they do not, while also a large number of people are not bothered by it at the same time. We have found a similar contradictive response regarding the necessity of access control in OSNs, as Figure 3-6 shows, only less than 20% of these people think it is not necessary, while most people, nearly 55% of the respondents believe that controlling access is necessary, and around 25% of people do not care about having controls over their profiles.

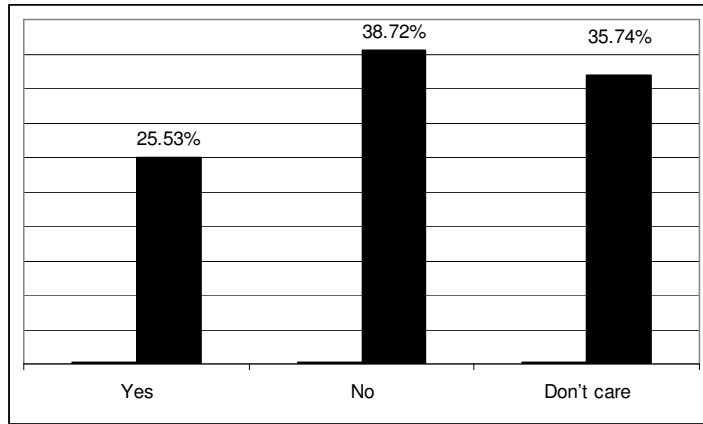


Figure 3- 5: Would you trust random strangers to view your profile?

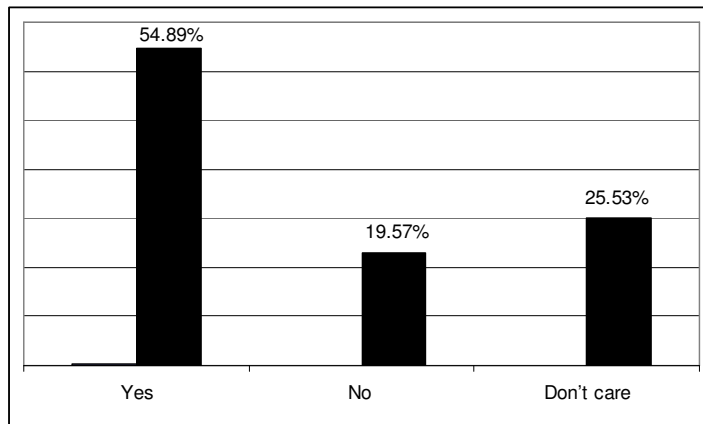


Figure 3- 6: Is it necessary that only certain people can view certain parts of your profile?

3.5.2 Category Two – No Longer Active OSN users

During their memberships of the 50 respondents in this category, 46% of people had set their profiles accessible by anyone, as Figure 3-7 shows, 26% allowed only directly linked people to view their profiles.

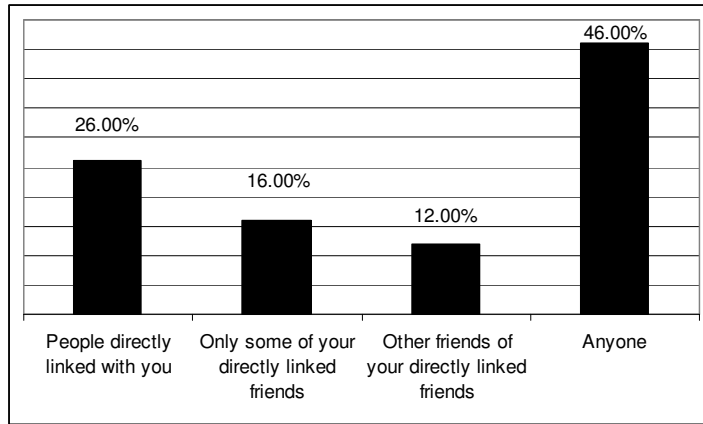


Figure 3- 7: Access settings of user profiles – Category Two

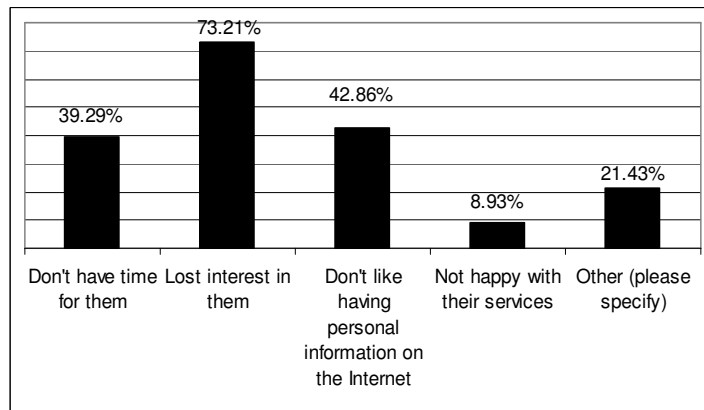


Figure 3- 8: Why did you stop using OSNs?

When asked about why have you stopped using OSNs, as Figure 3-8 shows, besides the given options, this category of people gave several interesting reasons. For instance, a lot of people lost interest in OSNs, sometimes due to unpleasant personal experience, or the completion of research or work related projects, or simply do not have time for them any more. In our survey, 5% of people in category two view OSNs as a rather sad way of replacing real life associations, especially since a lot of sites keep records of the number of visits a profile gets, some feel that OSNs have been turned into a form of popularity contest for certain personalities. However, at the same time, many acknowledged the fact that OSNs are cheap alternatives to keep updated with others, but a refinement in their structure is needed. In particular, privacy concerns were on top of the list, many have mentioned unpleasant experiences during their membership. Such as on some sites, comments left by close friends are displayed to everyone who were connected to an individual or sometimes, anyone with a browser; also, being

contacted unwillingly by random strangers or friends of friends whom they barely knew; especially the unfortunate fact that ways to stop these from happening do not always seem to work, distress and frustration had been caused due to the limited methods that are available.

When asked whether they think access controls of profiles are necessary in OSNs, this group of people had a similar response to category one. Among 47 participants who answered this question, 66% of people believed that it is necessary, only 6% of people disagreed, with the remaining not caring.

3.5.3 Category Three – Not Users of OSNs as yet

We were interested to find out why this group of people have never used OSNs, among 57 respondents, some had no interest, some had no time, others dislike the idea of having private information on the Internet and a small number of people have not heard of OSNs, as Figure 3-9 shows. Again, privacy concerns and the lack of freedom of controlling access to information have been mentioned by the 21.05% of people who stated otherwise when answering the question.

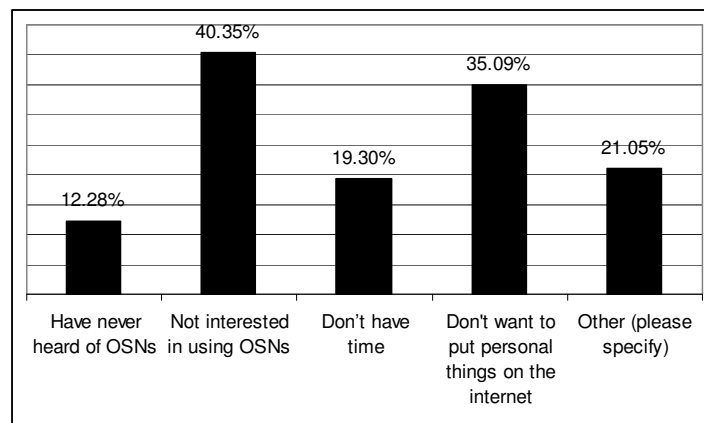


Figure 3- 9: Why have you never used OSNs?

Among 52 participants from this category, we asked whether it is likely for them to use OSNs in the future and whether they believe controlling access to profiles are necessary, 44% of people stated that they would start using OSNs in the future and 69% of whom think it is necessary to control access, only 4% of people disagreed.

3.5.4 Desired Trust Features and Opinions on the Proposed Solution

We asked 334 people the question whether they would trust all their directly linked friends to view all parts of their profiles, and as Figure 3-10 shows below, although 49.40% candidates trust all connected friends, 31.14% of people state that they would only trust some of these friends, while 6.59% participates clearly state that they do not trust them. The finding suggests that a total of 38% people think it is necessary to present only certain parts of their profiles to certain friends.

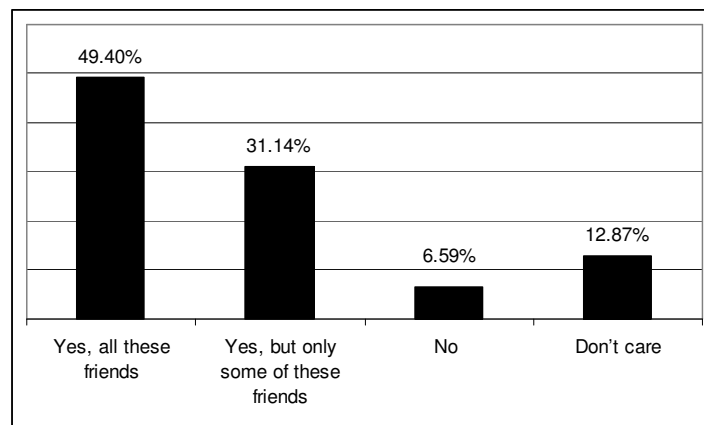


Figure 3- 10: Do you trust all your connected friends with all parts of your profile?

If a multi-faceted model of trust that calculates weighted average of the eight trust attributes: credibility, honesty, reliability, reputation, competency, belief, faith and confidence, is to be integrated into OSNs, would that be welcomed? Would rankings of these eight attributes of a person portrait subjective views of trust in OSNs? With the aim of finding out more on our proposed solution, we asked our participants' views on desired trust features in OSNs as well as their feelings towards a rating feature.

We asked 315 participants which of those eight attributes of trust are most important in their opinions, as Figure 3-11 shows, honesty appears to be the most important factor, closely followed by credibility and reliability as well as reputation.

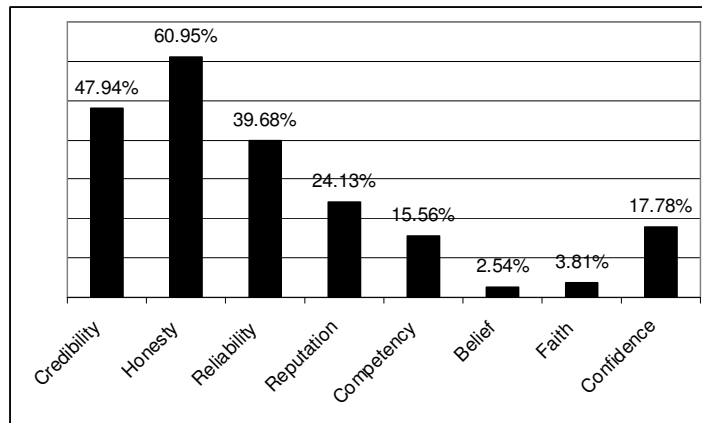


Figure 3- 11: Views on the eight attributes of trust

When asked would you like to see the ratings others have given you, 44% of participants said yes, 36% said no and with the remaining not caring about it. However, when asked whether they'd like to rate others, as Figure 3-12 shows, 211 people think it's unnecessary, only 9% of respondents believe that it would be helpful, another 10% of people do not care and with the remaining not being able to decide on the subject.

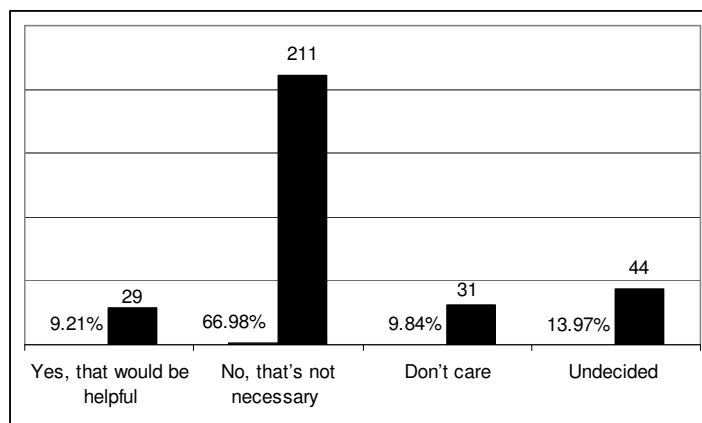


Figure 3- 12: Would you rate how much you trust your friends in OSNs?

Among 315 participants, only less than 10% of people were interested in giving trust ratings for their friends in OSNs, close to 10% of people are not concerned with the feature, and almost 14% of candidates were unable to decide on the topic, while the majority of people, with over 66% of respondents claim that they would not use such a feature in OSNs. Various reasons were given disapproving the idea of rating, such as, while rating a movie or a friend's photo sounds like fun,

however, OSNs are not meant to be taken too seriously; many stated that rating friends in OSNs seems cruel, and were worried that setting scores for each other would encourage bullying behaviour.

3.6 Survey Analysis

Several issues have been discovered during the survey, as discussed below:

Current trust mechanisms need to be refined.

Single-faceted, simple access control methods are being used in OSNs, where users can set their profiles viewable to either anyone or specified groups, even though trust levels vary among members of defined groups, users can not adjust their levels of trust among their connected friends. Most mentioned unpleasant experiences in OSNs are related to a lack of, or unsatisfying privacy control as well as access control methods of user profiles, while a large number of OSNs fail to allow users to express their various degrees of trust in a person, or a group of people context-specifically. Users cannot express their subjective views on trust freely, and the fundamental trust characteristics mentioned in section 2.5.2 are not captured in OSNs.

Better control of user profiles is welcomed.

As our findings suggest, a large number of users do not trust anyone and everyone to view all parts of their profiles, and believe controls are indeed necessary in OSNs. However, existing trust mechanism in OSNs have not achieved user satisfaction, hence, refinement of trust management is welcomed in OSNs.

Users are unsure about a multi-faceted model of trust with rating features.

Contradictive findings in relation to rating features suggest that on one hand, users think that such facilities would help in gaining better control of online profiles, on the other hand, they find it hard to rate someone they know personally. Such opinions could be the result of a lack of understanding regarding the proposed solution, as for a large percentage of candidates, since the word “rating” is so open to interpretation, it would be very hard for them to simply imagine what ratings

could be like without having the slightest ideas of how-to go about doing it. Also, we need to recognise limitations of the questionnaire, phrasing of the questions and limited open-ended questions in the survey could restrict the amount of quality data.

3.7 Summary

In this chapter, we discussed the design and execution of *A Survey of Online Social Networks*, as well as presenting several facts and figures from the questionnaire. Our analysis of the findings suggested that current trust mechanisms in OSNs fail to provide personalisation for users, which motivated us to research deeper into the topic area.

CHAPTER FOUR: DESIGN

Design in art, is recognition of the relation between various things, various elements in the creative flux. You can't invent a design. You recognize it, in the fourth dimension. That is, with your blood and your bones, as well as with your eyes.

~ D. H. Lawrence

4.1 Introduction

This chapter presents the design of a small scale online social network, *miniOSN*, in particular its trust management approach. There are several important influences on the design of the trust management system, which are discussed in section 4.2 and section 4.3. We then present main functionalities in *miniOSN* in relation to trust management in section 4.4. And finally, section 4.5 is a summary of this chapter.

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

Influenced by Quinn's trust model, we believe that applying the multi-faceted model of trust in OSNs would help users to express their subjective views on trust. Therefore in *miniOSN*, trust is defined in abstract and concrete concepts, with eight trust attributes, namely, credibility, honesty, reliability, reputation, competency, belief, faith and confidence. And ratings can be given to all eight trust attributes depending on how users view trust.

Also, an important change is made to the model: weighted average ratings of the eight trust attributes are taken away from the design in this research for two reasons discussed below.

Presumed equal importance

If a weighted average is to be calculated of the eight trust attributes, it is presumed that these attributes are equally as important as each other, which takes away the very subjectivity factor we are trying to achieve when modelling trust in OSNs. If a system is to calculate averages of the eight trust ratings, the freedom of expressing trust personally would be taken away from the users as a result, which contradicts with our research goals.

Suggested comparison

When associating scores with individuals, comparison of the scores is therefore suggested. However, such scores are simply representations of subjective views on the trustworthiness of the individuals in question. Comparing two people's average ratings may become misleading. For example, two people may have the same weighted average rating, however, one has high reliability rating and low honesty rating, while the other has high honesty rating and low reliability rating, how could we compare that one can be trusted more than the other?

4.3 Influences from OSN User Behaviour and Attitude Survey

From our initial survey, we had found mixed opinions in relation to the proposed rating feature in OSNs. We recognize that the proposed solution probably would work very well in an e-market environment like *Amazon* [Amazon] and *eBay* [eBay] where users have no previous connections with one another, and are building relationships from scratch, rating a person seems reasonable and acceptable. However, in the OSN environment, this is certainly not the case. Most users of OSNs are already friends with one another, and they are building their online relationships on existing friendships, where rating a person they already know personally seems difficult.

Due to reasons discussed above, in our design, we decided to take into account the human feeling factor, and decided to make the ratings given to connected friends only viewable to the person who rated them.

4.4 *MiniOSN* and Its Trust Mechanism

With influences from Quinn's trust model and considerations for user requirements, we introduce *miniOSN*, an online social network with a trust rating feature implemented, with the goal of letting owners of profile resources – be it a picture, a blog entry or a comment – decide: whom to see what, when they want them to, depending on the situation.

miniOSN has functionalities of a basic online social networking website, it allows users to create accounts for themselves with a username and password and a valid email address. Users of *miniOSN* can then set up representations of themselves, upload photos, post blog entries, as well as leaving comments in connected friends' profiles.

Fictional characters from the situational comedy *Friends* [Friends] have been created as users of *miniOSN* to be able to show features and functionalities of the network.

Figure 4-1 shows the screenshot of the signup page in *miniOSN*, Figure 4-2 shows the page where a registered user can edit his/her profile details. Figure 4-3 shows a member named Rachel and her current profile, with thumb nailed photos and two blog entries, links are also provided to see all of her uploaded photos as well as posted blogs; clicking on the title of one of her blogs will direct her to the page where that blog is located, along with a collection of comments left by others, as shown in Figure 4-4, Monica has left a comment in relation to her blog titled *What have I become?*.

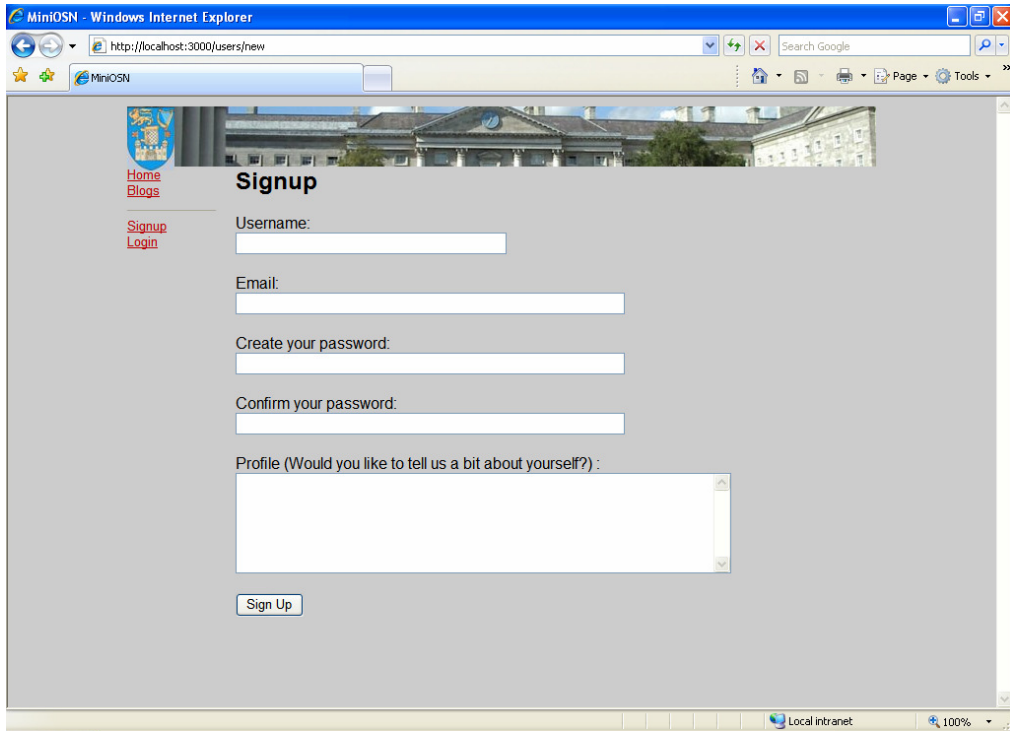


Figure 4- 1: Screenshot of the Signup Page

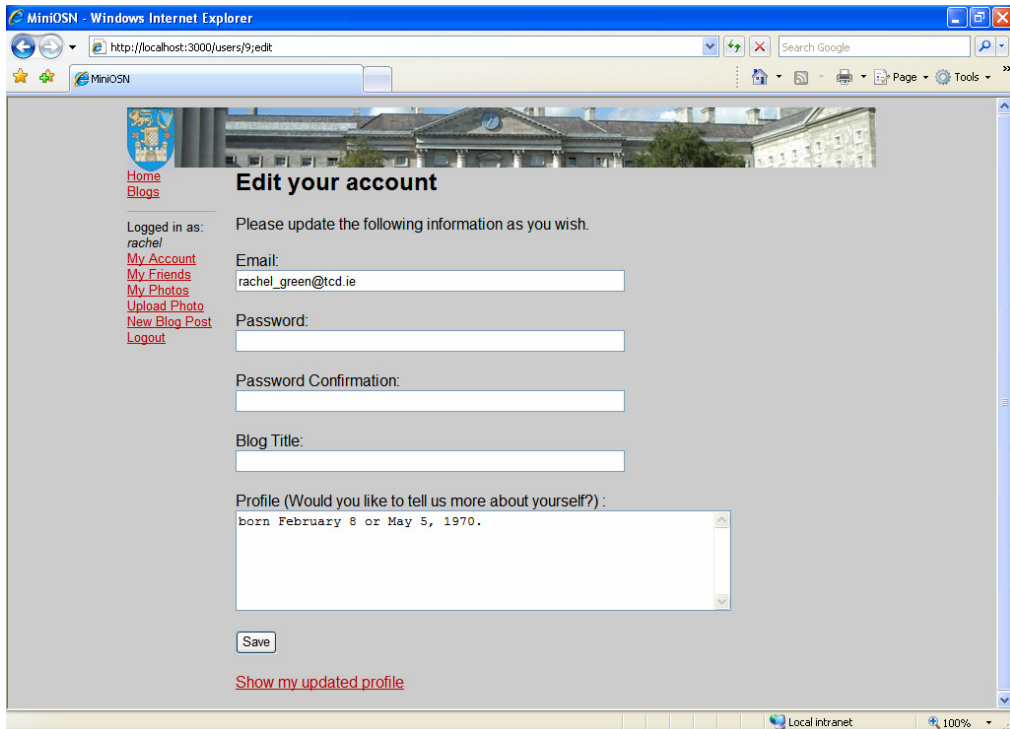


Figure 4- 2: Screenshot of the Edit Account Page



Figure 4- 3: Screenshot of a User Profile

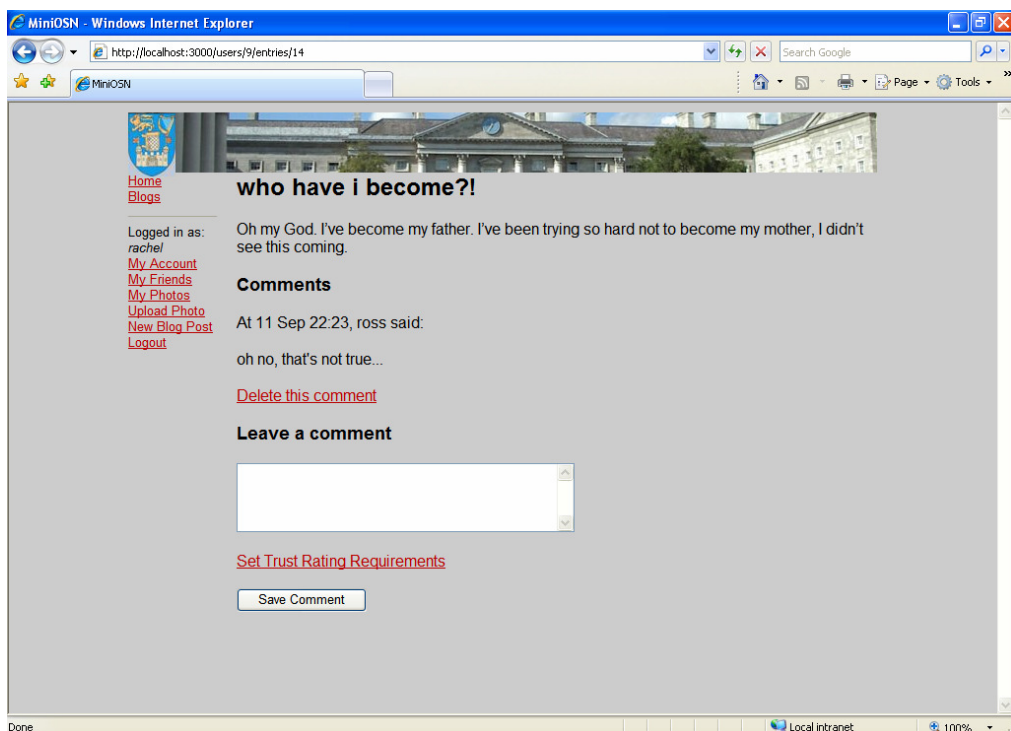


Figure 4- 4: Screenshot of a User Blog and Comments by Others

The link “My Friends” on the left hand side of the menu bar directs a user to the page where a list of all his/her connected friends is presented. As Figure 4-5

shows a list of all Rachel's connected friends in *miniOSN*, by clicking on a friend's name, she will be directed to that friend's profile. The "Edit Friendship" link will bring a user to the page where he/she can specify information with regards to a certain friendship, as Figure 4-6 shows, Rachel can edit whether she works with Chandler, or related to him by family, etc.

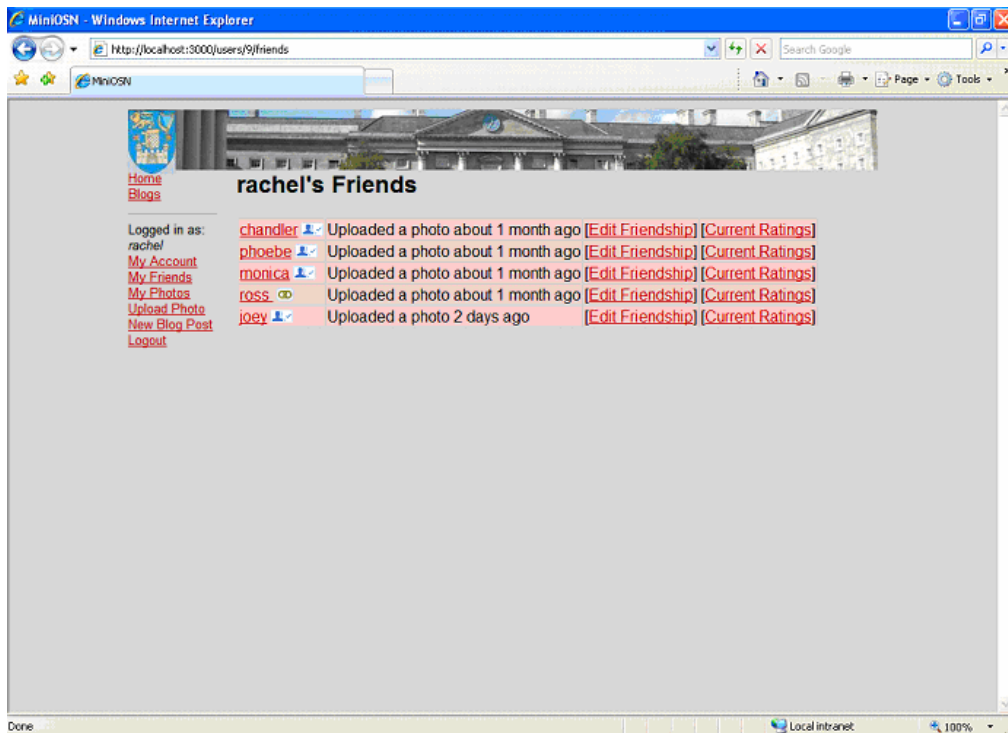


Figure 4- 5: Rachel's Connected Friends

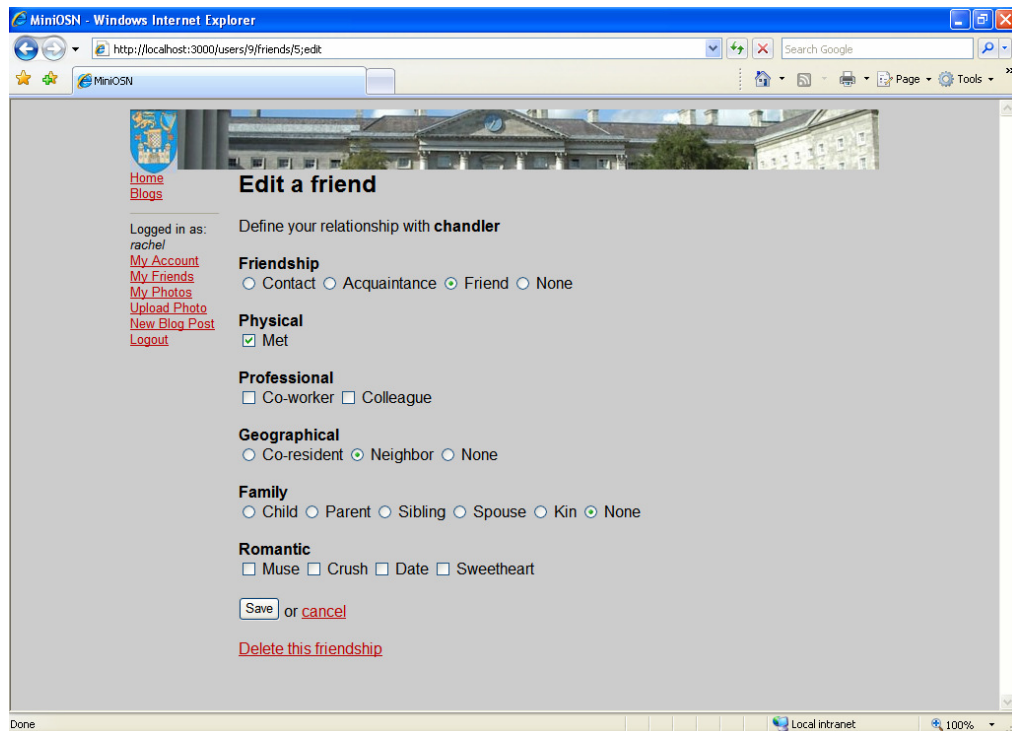


Figure 4- 6: Screenshot of Editing a Friendship

When designing the trust management mechanism for *miniOSN*, a great deal of effort was put into its ability of capturing the characteristics of trust as found in the literature review, hence, all implemented trust features are designed with the aim of mirroring properties of trust as discussed in section 2.5.2.

miniOSN allows users to set trust rating requirements, if desired, for each uploaded photo, blog and comment, Figure 4-7 shows an example of the trust attribute matrix for a blog entry, before posting the blog, users can specify requirements of trust rating values in order to control access to this resource. Similarly, by adjusting required values of the eight trust attributes before uploading a photo or leaving a comment, users can decide which friend(s) can view them.

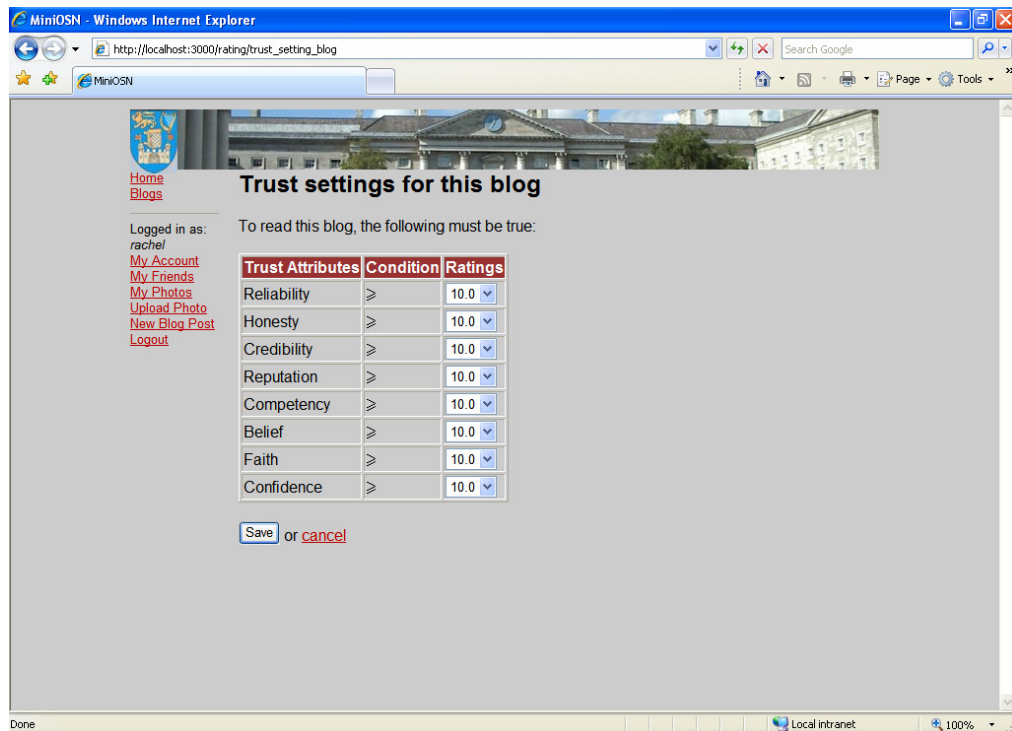


Figure 4- 7: Screenshot of Setting Trust Requirements for a Blog

Several important features of *miniOSN* in relation to its trust management approach are listed below:

Each user holds ratings of his/her connected friends in the database

Trust is personalised, two people can have very different opinions of the trustworthiness of the same person, as we have found in the literature review. To be able to capture this property of trust, in *miniOSN*, each user holds ratings of each one of their connected friends in the database, identified by `user_id`. For instance, in Figure 4-5, user Rachel has five connected friends in her profile, each “Current Ratings” link will then bring her to the page where that friend’s current trust ratings are shown. Figure 4-8 shows the list of Ross’s current trust ratings according to Rachel, which are a different set of values than the ones Chandler has given Ross as Figure 4-9 shows.

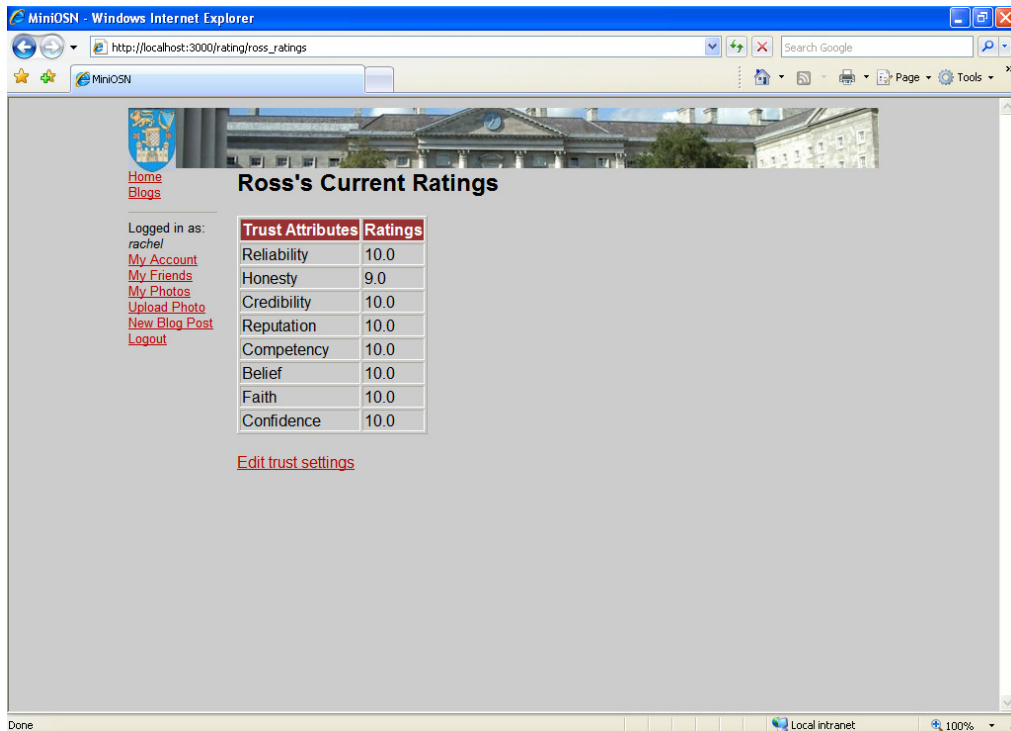


Figure 4- 8: Rachel's Current Trust Ratings for Ross

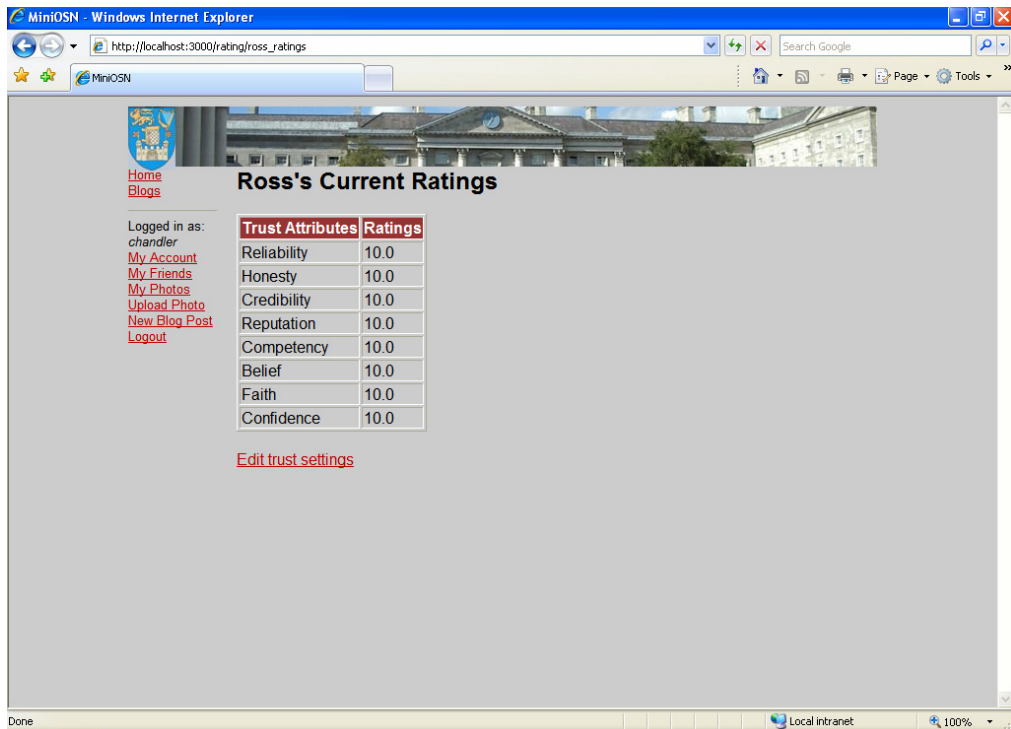


Figure 4- 9: Chandler's Current Trust Ratings for Ross

Trust is also asymmetric, by letting each user hold records of connected friends, this also enables the possibility of expressing this asymmetry property of trust. For

example, as talked about earlier, Rachel could have a rating of nine for Ross's honesty attribute, Ross, on the other hand, thinks Rachel is an excellent friend, as Figure 4-10 shows.

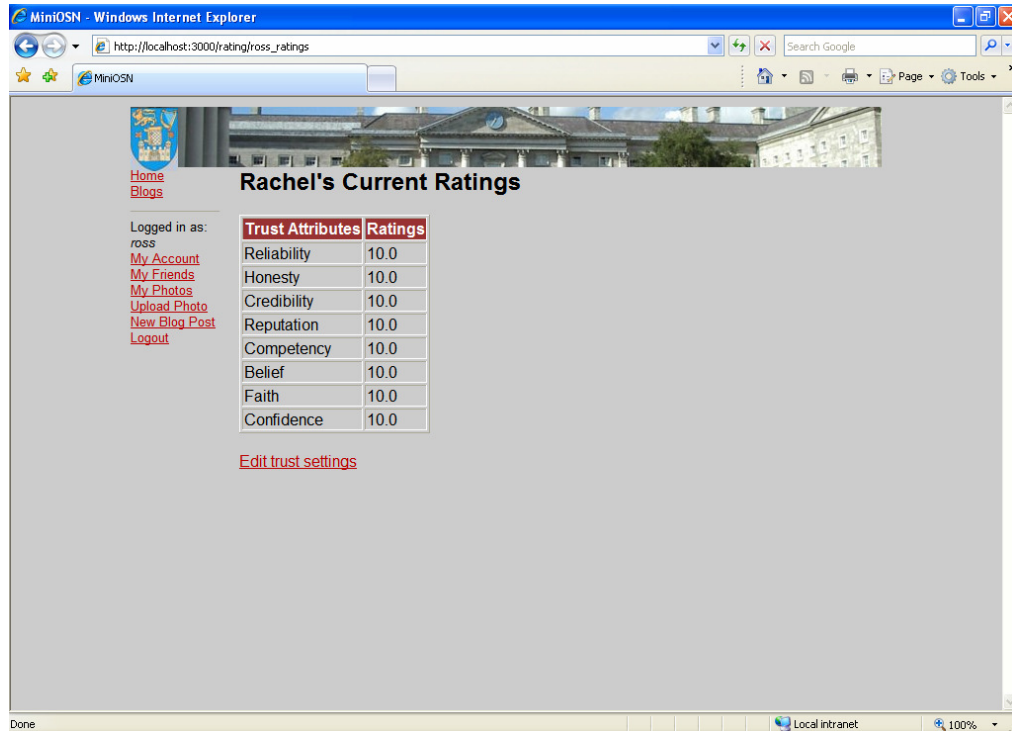


Figure 4- 10: Ross's Current Trust Ratings for Rachel

The owner of a resource is able to set the trust requirements before distributing that resource

In order to let users of *miniOSN* have complete control over their profiles and resources in them, whether it is uploading a picture, posting a blog entry or leaving a comment in someone else's profile, as long as you own this resource, you can then decide the trust rating requirements for this resource.

For example, in the diagram below, see Figure 4-11, user Rachel can click on the "Set Trust Rating Requirements" link before uploading a picture in her profile, which will then direct her to the page where the trust rating matrix is, as Figure 4-12 shows, she can then change the values of them against the ratings she has given her connected friends in order to grant different levels of access control to her profile or certain resources in it.

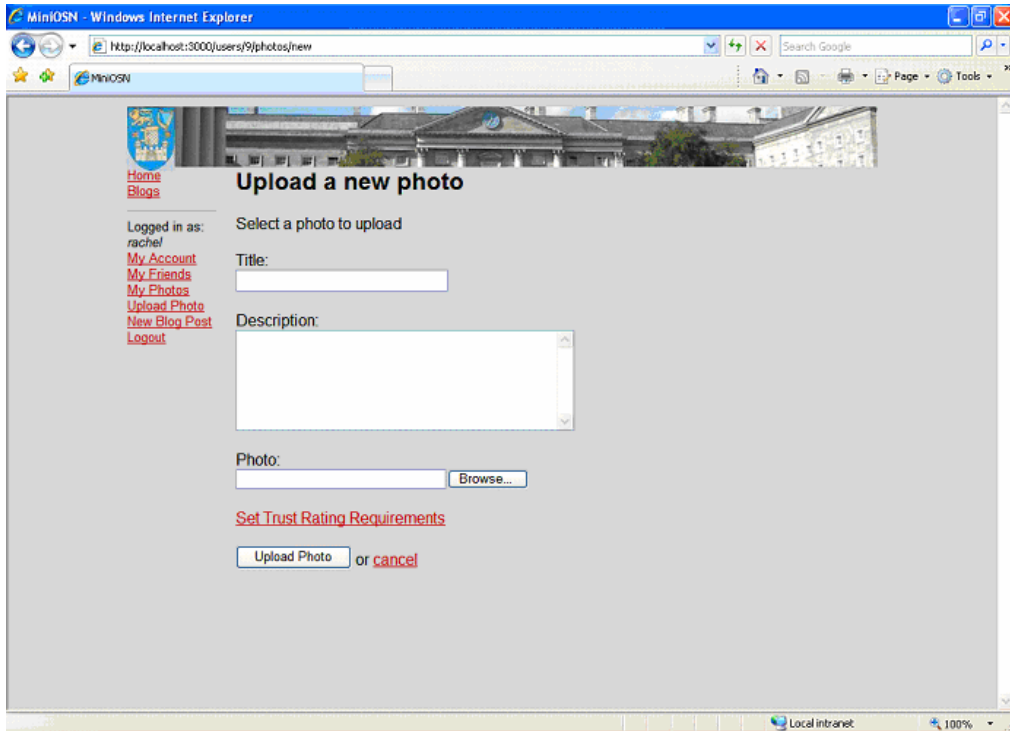


Figure 4- 11: Screenshot of the Photo Upload Page

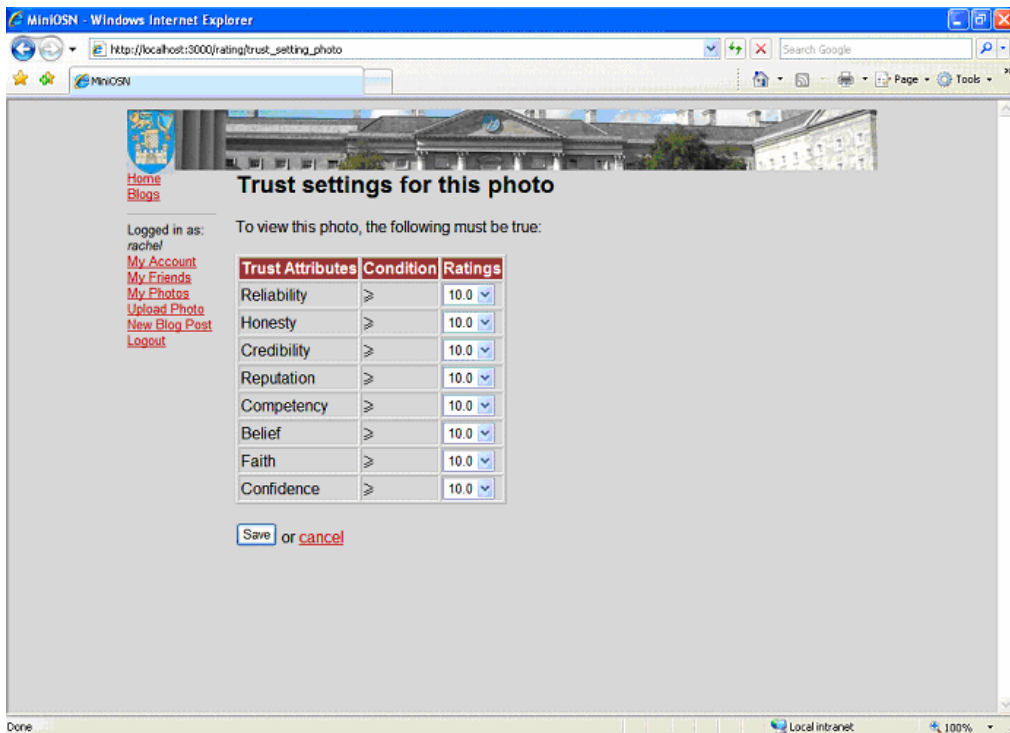


Figure 4- 12: Setting Trust Rating Requirements for an Uploaded Photo

All users and resources have default ratings of 10 out of 10

Findings from our initial survey suggest that users may find rating a friend difficult, therefore, they can choose not to use the implemented rating feature of *miniOSN*, by simply ignoring the trust rating values since all are set to default 10 out of 10, which means that all connected friends can access all resources in a profile, until a user makes changes to trust ratings of connected friends and/or trust rating requirements for certain resource(s) in the profile.

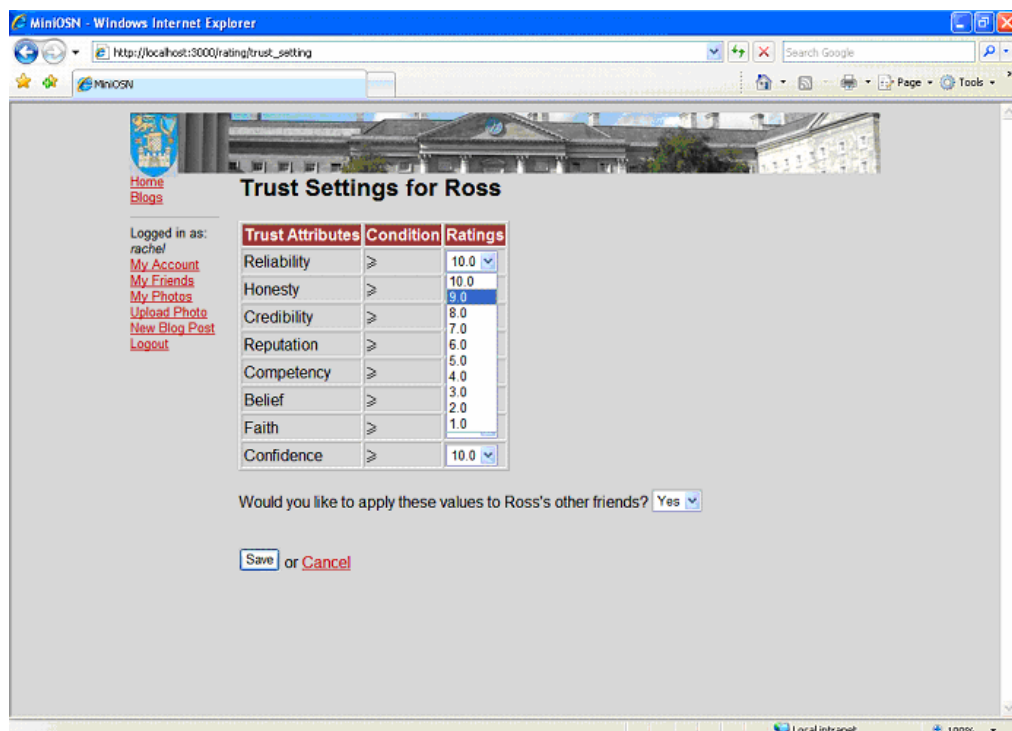


Figure 4- 13: Setting Trust Values for a Friend and Friends of the Friend

Users decide whether to transfer trust values to other friends of a friend

In *miniOSN*, users decide whether they would like to express trust transitively. Once a connected friend's trust ratings have been set, the owner of the profile can then decide whether the same set of ratings should be transferred to all other friends of this certain friend, whom the owner of the profile is not currently connected to. For example as previous Figure 4-13 shows, once Rachel chooses option "Yes" to the question "Would you like to apply these values to Ross's other friends?", all other friends of Ross whom Rachel is not connected to would have the same trust level as Ross's.

Users decide their connected friends to start with whatever ratings they want them to

Although by default, all ratings are set to 10 out of 10, however, the owner of a profile can adjust these settings and decide on whatever ratings they would like their friends to start with. Hence, the freedom of expressing various levels of trust among connected friends is provided to *miniOSN* users.

Any connected friend's trust ratings can be reset whenever it is desired

In order to allow users express trust context-specifically in *miniOSN*, the owner of a profile can change trust ratings for their connected friends whenever it is desired, depending on the situation. For example, if a certain blog should not be seen by a particular friend, the owner can adjust trust ratings of that friend so the blog in question is not accessible by that person.

4.5 Summary

We have introduced *miniOSN* in this chapter, and discussed several important influences on the trust management approach that were considered when designing this online social network. We have made changes to the multi-faceted model of trust that is personalisable and specialisable, and discussed the reasons behind such changes. In addition, we have also presented functionalities in *miniOSN* accompanied by screenshots of the trust management system deployed.

CHAPTER FIVE: IMPLEMENTATION

One thing only I know, and that is that I know nothing.

~ Socrates

5.1 Introduction

This chapter concentrates on the implementation process of *miniOSN*, and the technologies used. In section 5.2, Ruby on Rails is briefly discussed, and the various advantages for implementation of the project outlined. Then, in section 5.3, the set of technologies involved when building *miniOSN* are described. Section 5.4 provides the architecture of *miniOSN* and explains how the website works, followed by technical issues and challenges encountered in section 5.5. Finally, we summarise this chapter in section 5.6.

5.2 Ruby on Rails

In recent years, Ruby on Rails (RoR) has become more and more popular with web developers due to its ability of building stable, scalable and maintainable applications quickly and easily. Many real world applications such as *Basecamp* [Basecamp] and *37signals* [37Signals] are powered by RoR.

RoR is simply a set of libraries and tools, known as the framework, written in Ruby to allow rapid development of web applications.

Ruby [Ruby] was first introduced by Yukihiro Matsumoto, it is an open source, object oriented programming language, with scripting feature similar to Python and Perl. Its object oriented concept from C++ and Java also maintains the reliability of programming in addition to maintaining the security of code.

5.2.1 Design Principles

Rails [Rails] is an open source framework, designed by David Heinemeier Hansson. Rails provides out-of-the-box scaffolding, which can quickly construct most of the logic and views needed for a basic website, the WEBrick web server and other helpful development tools.

It has a couple of main design principles [Bradburne, 2007, p. 4]: don't repeat yourself (DRY) and convention over configuration. "DRY" is self-explanatory. If you have defined something once, you should not have to define it elsewhere. For instance, once you have defined the column names in a database schema, you should not have to repeat them elsewhere in your code. This reduces the amount of work and prevents inconsistencies in your code.

"Convention over Configuration" means a developer only needs to specify unconventional aspects of the application. If you establish a set of naming conventions and suchlike, you can substantially cut down on the amount of configuration that is required to set up handler mappings, view resolvers, ModelAndView instances, etc. This is a great benefit with regards to rapid prototyping, and can also lend a degree of consistency. For example, if there is a class `User` in the model, the corresponding table in the database is called `users` by default. It is only if someone deviates from this convention, such as calling the table "`registered_members`", that he needs to write code regarding these names.

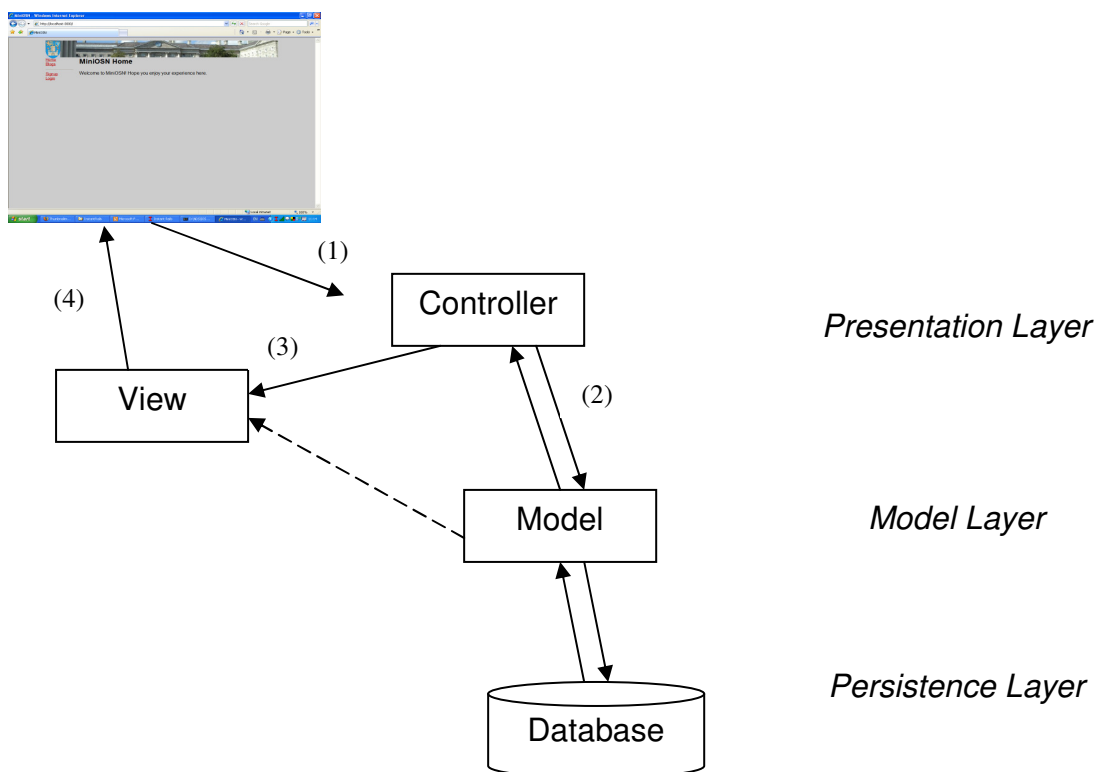
5.2.2 MVC Architecture

The Ruby on Rails framework implements the model-view-controller (MVC) architecture, which is a set of design patterns that allows you to separate the data model, the user interface, and the control logic of your application.

- The *Model* in a Rails application holds all business logic, as well as maintains the state of the application. Models are "smart" domain objects that knows how to persist themselves to a database, in other words, they are the gate keeper as well as the data store.

- The *View* is simply a presentation of the data, it is responsible for generating user interface based on the data in the Model. Views are “dumb” templates that are responsible for inserting pre-built data between HTML tags.
- The *Controller* is responsible for arranging the application, it handles incoming requests, such as save new user account, update user profiles, etc., by manipulating the Model and directing data to the View.

The following diagram shows the MVC architecture in abstract terms.

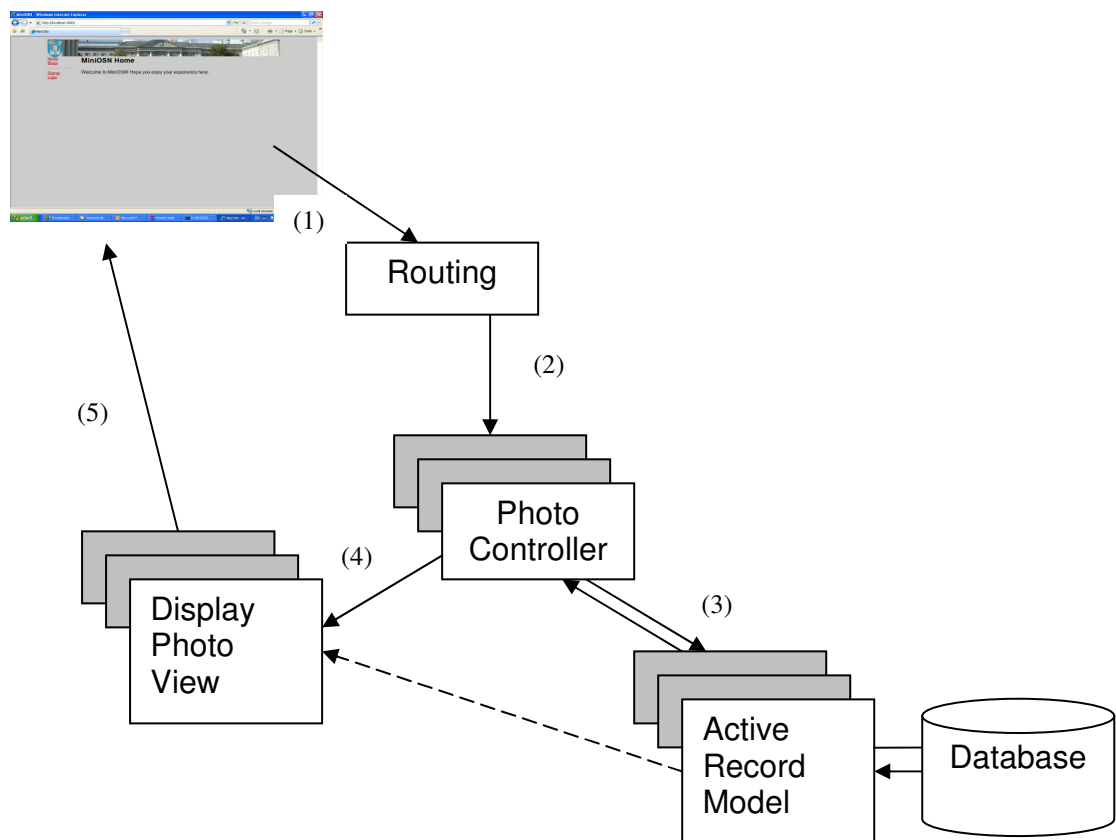


- (1) Browser sends request
- (2) Controller interacts with Model
- (3) Controller invokes View
- (4) view renders next browser screen

Figure 5 - 1: The MVC Architecture

For example, the application has previously displayed an upload photo page and user with `user_id 9` has just clicked on the Upload Photo button after attaching an image file from his/her computer. This Upload Photo button links to `http://localhost:3000/users/9/photos/create_photo/26,`

where `create_photo` is an action in the application and 26 is the internal id for the photo being uploaded. The application first checks that user with id 9 is logged in, the routing component in Rails receives the incoming request and immediately picks it apart. In this case, it takes, `photos`, as the name of the controller and, `create_photo`, as the name of an action, and, 26, is by convention extracted into an internal parameter called `photo_id`. As the result of this analysis, the router knows it has to invoke the `create_photo` method in the controller class `PhotosController`. As Figure 5-2 shows below.



- (1) `http://localhost:3000/users/9/photos/create_photo/26`
- (2) Routing finds Photo controller
- (3) Controller interacts with model
- (4) Controller invokes view
- (5) View renders next browser screen

Figure 5 - 2: Rails and MVC

The advantage of using such an MVC architecture is that the code is cleanly separated into logical sections that are easy to develop, understand, maintain, and

control. Because Rails make such architectural decisions for you, you can then spend more time working on the application and less time worrying about the information flows within the system.

In a Rails application, all of the interaction with the database is performed through *ActiveRecord*, which is an Object/Relationship Mapping (ORM) library that maps the data stored in a database to a class in your application. This allows you to access your data without having to worry about the SQL queries or even exactly how the data is accessed.

ActionPack provides the “view” and “controller” of the MVC stack, and is simply a collection of libraries and tools to help you build web applications. The view part of ActionPack is used to create the web pages themselves. The controller part of ActionPack is the glue that holds the application together.

For example, in *miniOSN*, to add a user account system along with an interface to allow users to sign up, log in and log out, we first create the database migration file, as the code snippets in Figure 5-3 shows.

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.column :username, :string, :limit => 64, :null => false
      t.column :email, :string, :limit => 128, :null => false
      t.column :hashed_password, :string, :limit => 64
      t.column :enabled, :boolean, :default => true, :null => false
      t.column :profile, :text
      t.column :created_at, :datetime
      t.column :updated_at, :datetime
      t.column :last_login_at, :datetime
    end
    add_index :users, :username
  end

  def self.down
    drop_table :users
  end
end
```

Figure 5 - 3: The User Table Migration File Snippet

We then define the validations for the model, so that it holds user data and stores information such as each user’s login details, as Figure 5-4 Shows. We use a one-way hashing algorithm to store users’ passwords, since SHA-256 is part of the standard Ruby libraries.

```

require 'digest/sha2'
class User < ActiveRecord::Base
  attr_protected :hashed_password, :enabled
  attr_accessor :password

  validates_presence_of :username
  validates_presence_of :email
  validates_presence_of :password, :if => :password_required?
  validates_presence_of :password_confirmation, :if=>:password_required?

  validates_confirmation_of :password, :if => :password_required?

  validates_uniqueness_of :username, :case_sensitive => false
  validates_uniqueness_of :email, :case_sensitive => false

  validates_length_of :username, :within => 3..64
  validates_length_of :email, :within => 5..128
  validates_length_of :password, :within=>4..20, :if=>:password_required?
  validates_length_of :profile, :maximum => 1000

  def before_save
    self.hashed_password=User.encrypt(password)if!self.password.blank?
  end

  def password_required?
    self.hashed_password.blank? || !self.password.blank?
  end

  def self.encrypt(string)
    return Digest::SHA256.hexdigest(string)
  end

  def self.authenticate(username, password)
    find_by_username_and_hashed_password_and_enabled(username,
    User.encrypt(password), true)
  end
end
end

```

Figure 5 - 4: The User Model File Snippet

We now build the users controller and use the `show_by_username` method to display the profiles of users referenced by their usernames, as the following code snippets show.

```

class UsersController < ApplicationController
  def index
    @users = User.find(:all)
  end

  def show
    @user = User.find(params[:id])
  end
end

```

```

def show_by_username
  @user = User.find_by_username(params[:username])
  render :action => 'show'
end

def new
  @user = User.new
end

def create
  @user = User.new(params[:user])
  if @user.save
    self.logged_in_user = @user
    flash[:notice] = "Your account has been created."
    redirect_to index_url
  else
    render :action => 'new'
  end
end

def edit
  @user = logged_in_user
end

def update
  @user = User.find(logged_in_user)
  if @user.update_attributes(params[:user])
    flash[:notice] = "User updated"
    redirect_to :action => 'show', :id => logged_in_user
  else
    render :action => 'edit'
  end
end
end
end

```

Figure 5 - 5: The Users Controller File Snippet

And finally, we completing the application by creating the views that correspond to the actions we have created for users controllers as code snippets shown in Figure 5-6 and Figure 5-7.

```

<h2>Signup</h2>

<%= error_messages_for :user %>

<% form_for :user, :url => users_path do |f| -%>
  <p>Username:<br /><%= f.text_field :username, :size => 40 %></p>
  <p>Email:<br /><%= f.text_field :email, :size => 60 %></p>
  <p>Password:<br /><%= f.password_field :password, :size => 60
    %></p>
  <p>Password Confirmation:<br />
    <%= f.password_field :password_confirmation, :size => 60 %></p>
  <p>Profile:<br /><%= f.text_area :profile, :rows => 6, :cols => 60
    %></p>

```

```
<%= submit_tag 'Sign Up' %>
<% end -%>
```

Figure 5 - 6: The New User Sign Up View Snippet

```
<h2>Login</h2>

  <% form_for :user, :url => {:action => 'authenticate'} do |f| -%>
    <p>Username:<br /><%= f.text_field :username, :size => 30 %></p>
    <p>Password:<br /><%= f.password_field :password, :size => 30
      %></p>
    <%= submit_tag 'Login' %>
  <% end %>
```

Figure 5 - 7: The Login View Snippet

5.2.3 Metaprogramming

Rails uses a technique called metaprogramming to create domain-specific language (DSL), i.e., web applications are the domain, and Rails is a language that helps us describe our problems within this domain.

The ORM ActiveRecord provides a DSL for accessing the data, which means that we can use commands like `find_user_by_username` instead of having to go through lengthy sections of code that connect to a database, perform a SQL query, and then process the results.

5.2.4 Test Driven Development

The Rails framework comes complete with integrated automated testing tools, which make it incredibly simple to write unit, functional, and integration tests. As writing the tests is so simple, it makes sense to write them at the same time as you develop the code, instead of leaving them till the very end of the development cycle and not being able to give them enough attention.

5.2.5 Rails Plugins

Rails plugins are either extension or modification of the core framework, they are self-contained libraries made especially for Rails. They are a great way to reuse someone else's code or to package your own code for reuse. Plugins provide [RoR, 2007]:

- a way for developers to share bleeding-edge ideas without hurting the stable code base
- a segmented architecture so that units of code can be fixed or updated on their own release schedule
- an outlet for the developers so that they don't have to include every cool new feature under the sun

5.3 Building *MiniOSN* with Ruby on Rails

miniOSN runs on Instant Rails 1.7 [Instant Rails] for Windows, which is a one-stop Rails runtime solution containing Ruby, Rails, Apache, and MySQL, all pre-configured and ready to run.

A CD-ROM containing the complete source code of *miniOSN* Rails application is also provided with this thesis.

Figure 5-8 shows a file directory of the *miniOSN* application, and a list of all the controllers that have been created in the project.

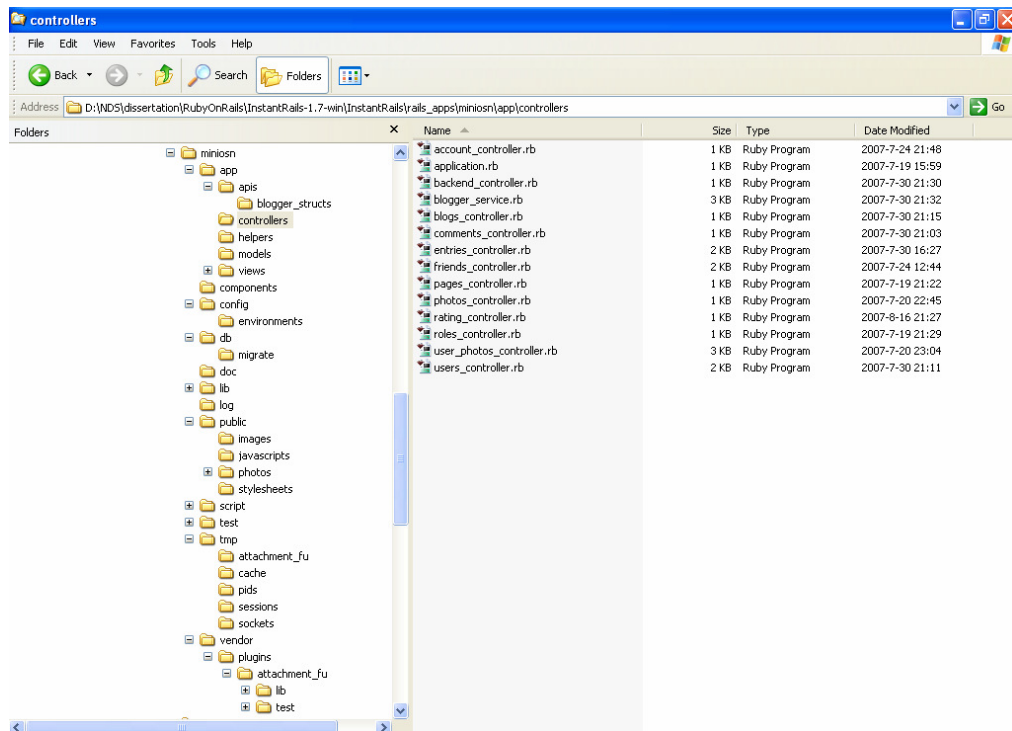


Figure 5 - 8: *miniOSN* File Directory

miniOSN has the following main functionalities:

- A login system, allows users to create user accounts and add profiles about themselves. A registered user can then log in with a username and a password.
- A blogging engine, allows members to create their own blogs.
- A photo gallery, for each user, allows members to upload their photos to their profiles and supports thumbnails of each photo.
- A trust management system, allows users to annotate trust ratings among connected friends, and grant access to specified friend by setting trust requirements against that person's trust rating values.

Strongly influenced by *Practical Rails Social Networking Sites* [Bradburne, 2007], all features follow the MVC architecture closely, separate data models, user interface and control logic of the application. Several techniques used are discussed in section 5.3.1, 5.3.2 and 5.3.3.

5.3.1 RESTful Rails Routes

Rails supports mappings between URLs and controllers based on the concept of Representational State Transfer (REST) [Bradburne, 2007, p. 22] REST describes a set of architectural principles for building a system such as the Web. By REST principles, the Web is considered to simply be a collection of resources, and a web page is a representation of a specific resource. By utilizing the HTTP protocol, actions such as getting, setting, or deleting objects can be performed on these resources. Also, other representations of resources, such as in XML can be provided.

To make building REST resources easy, we can state that a controller provides access to a resource in the Rails routes file `config/routes.rb`. Doing this automatically sets up a number of mappings. We then simply need to provide the code to implement the standard REST methods that Rails expects.

Rails uses the HTTP request methods GET, POST, PUT, and DELETE together with the URL, meaning that the same URL can have different responses depending on which HTTP method is used. For example, an URL `/users` with HTTP request method GET would invoke the `index` action in Rails and return a collection of all current users registered, while the same URL with HTTP request method POST would trigger the `create` action in the Rails application and create a new user.

If other actions besides the standard actions provided are desired, we can specify extra actions for a collection or for a member of the resource in the `routes` file. Also, Rails allows you to respond to requests for XML responses of these actions.

5.3.2 The `Attachment_fu` Plugin, *ImageMagick* and *RMagick* Libraries

Like most OSNs, *miniOSN* allows registered users upload photos from their PCs to their online profiles. In order to work with uploaded files easily,

`attachment_fu` plugin [`attachement_fu` plugin] was used to help managing files, as well as `ImageMagick` [`ImageMagick`] and `RMagick` [`RMagick`] libraries, to create thumb nailed versions of the photos. For reasons discussed below:

Image file size concerns.

Digital images can easily be 2MB or 3MB in size. Which is of no problem if only dealing with a small number of files, however, it becomes very difficult to manage when thousands of users each upload hundreds of photographs, in that case, either a very generous hosting provider or a dedicated server with a large amount of online storage is needed. This also has an effect on the bandwidth that the site will consume. Almost certainly, the site will be hosted with a shared host, a virtual private server (VPS), or a dedicated machine at a collocation, meaning that an allocated amount of upload and download bandwidth is needed.

Security concerns.

If we allow users to upload and store any type of files, and allow others to download these files without checking the file size or file type to make sure that it is a valid image, it is possible that malicious users could take advantage of the system, i.e., to store other types of files, to attempt to hack or break the site by uploading illegal or malicious files.

The `attachment_fu` plugin

Developed by Rick Olson, the `attachment_fu` plugin automatically recognizes file types and can be configured to automatically create different sizes of thumbnails for images. Also, it is possible to configure several methods of storing the uploaded data, by default, it stores uploaded files in the database.

We used the `ruby script/plugin` command to install the `attachment_fu` plugin, from the following vendor address:

http://svn.techno-weenie.net/projects/plugins/attachement_fu

To make use of `attachment_fu`, `has_attachment` statement needs to be added in the `Photo` model which was created to store file uploads in *miniOSN*.

Figure 5-9 shows code snippet of the database, where `content_type` is the

Multi-Purpose Internet Mail Extensions (MIME) format of the uploaded file, `filename` is the original naming, bytes of the uploaded file is stored in `size`; if the file is a thumbnail, `parent_id` stores the id of the parent file and `thumbnail` is the name of the size of thumbnail as specified in the `has_attachment` statement; width and height of an image file are stored in pixels, and finally, `db_file_id` is used to store the uploaded files.

```
class CreatePhotos < ActiveRecord::Migration
  def self.up
    create_table :photos do |t|
      t.column :user_id, :integer
      t.column :title, :string
      t.column :body, :text
      t.column :created_at, :datetime

      # the following columns are required for attachment_fu
      t.column :content_type, :string, :limit => 100
      t.column :filename, :string, :limit => 255
      t.column :path, :string, :limit => 255
      t.column :parent_id, :integer
      t.column :thumbnail, :string, :limit => 255
      t.column :size, :integer
      t.column :width, :integer
      t.column :height, :integer
    end
    add_column :users, :photos_count, :integer
  end

  def self.down
    drop_table :photos
    remove_column :users, :photos_count
  end
end
```

Figure 5 - 9: The Photos Table Migration Snippet

ImageMagick and RMagick libraries

To create thumbnails for the images, two libraries have been installed: namely *ImageMagick* and *RMagick*.

ImageMagick is an open source set of general-purpose image processing libraries. It can read, convert and write images in a variety of formats including DPX, EXR, GIF, JPEG, JPEG-2000, PDF, PhotoCD, PNG, Postscript, SVG, and TIFF.

RMagick is a Ruby interface to the *ImageMagick* libraries, that makes resizing, cropping, and rotating images easy, also, it is possible to apply special effects such as blur and sharpen as well as producing composite images.

We used the Windows binary version of *RMagick*, namely, *rmagick-win32 RMagick 1.15.9 binary gem for Ruby 1.8.6* which includes the *ImageMagick* libraries, from the *RMagick* RubyForge project page:

<http://rubyforge.org/project/rmagick>

After installing the *ImageMagick* first, we used the `gem install` command to install the binary version of *RMagick* gem.

When creating the `Photo` model, we can then simply specify that the image processor uses *RMagick*, as Figure 5-10 shows below.

```
class Photo < ActiveRecord::Base
  has_attachment:storage => :file_system,
                    :resize_to => '640x480',
                    :thumbnails => { :thumb => '160x120', :tiny => '50>' },
                    :max_size => 5.megabytes,
                    :content_type => :image,
                    :processor => 'Rmagick'
  validates_as_attachment
  belongs_to :user
end
```

Figure 5 - 10: The Photo Model Snippet

Two thumbnails are created for uploaded images, one named `thumb`, with 160×120 pixels, and the other `tiny`, which is specified as `50>`, meaning that an image will be resized with the width of 50 pixels while keeping the aspect ratio of the original image. The `has_attachment` statement also specifies that only file types that are images will be accepted and that the maximum size of an uploaded file will be 5MB.

Also, we used a validation method introduced by the `attachment_fu` plugin, `validates_as_attachment`, which ensures that the attachment meets the requirements of being a file, i.e., having a size, a content type, and a file name.

The relationship with the `User` model is also stated.

5.3.3 XHTML Friends Network Microformat

Microformats [Microformats] are simple, open data formats that allow you to add semantic information to XHTML documents and allow users and applications to extract meaning from that page based on the markup. Microformats build on existing standards rather than trying to develop a whole new markup system. They are designed to address small, specific uses or sections of markup, such as contact details or calendar entries. By default, they do not change the way a page is shown in the browser, but the extra information that they provide makes it easy for software to understand the data on the page.

XHTML Friends Network (XFN) microformat [XFN] was used in *miniOSN* when displaying a link to a user, so that extra information about the relationship to this user can be added.

XFN was developed by Matthew Mullenweg, Eric Meyer, and Tantek Çelik, outlines the relationships between individuals by defining a small set of values that describe personal relationships. In HTML and XHTML documents, these are given as values for the `rel` attribute on a hyperlink. XFN allows authors to indicate which of the blogs they read belong to friends, whom they have physically met, and other personal relationships. Figure 5-11 shows the migration of the friendships table in *miniOSN*, and Figure 5-12 shows the Friendship model using XFN.

```
class CreateFriendships < ActiveRecord::Migration
  def self.up
    create_table :friendships do |t|
      t.column :user_id, :integer, :null => false
      t.column :friend_id, :integer, :null => false
      t.column :xfn_friend, :boolean, :default => false, :null =>
        false
      t.column :xfn_acquaintance, :boolean, :default => false, :null
        => false
      t.column :xfn_contact, :boolean, :default => false, :null =>
        false
      t.column :xfn_met, :boolean, :default => false, :null => false
      t.column :xfn_coworker, :boolean, :default => false, :null =>
        false
      t.column :xfn_colleague, :boolean, :default => false, :null =>
        false
      t.column :xfn_coresident, :boolean, :default => false, :null =>
        false
      t.column :xfn_neighbor, :boolean, :default => false, :null =>
        false
    end
  end
end
```

```

t.column :xfn_child, :boolean, :default => false, :null =>
  false
t.column :xfn_parent, :boolean, :default => false, :null =>
  false
t.column :xfn_sibling, :boolean, :default => false, :null =>
  false
t.column :xfn_spouse, :boolean, :default => false, :null =>
  false
t.column :xfn_kin, :boolean, :default => false, :null => false
t.column :xfn_muse, :boolean, :default => false, :null => false
t.column :xfn_crush, :boolean, :default => false, :null =>
  false
t.column :xfn_date, :boolean, :default => false, :null => false
t.column :xfn_sweetheart, :boolean, :default => false, :null =>
  false

end
add_index :friendships, [:user_id, :friend_id]
end
def self.down
  drop_table :friendships
end
end
end

```

Figure 5 - 11: The Friendships Table Database Snippet

```

class Friendship < ActiveRecord::Base
  belongs_to :user
  belongs_to :friend, :class_name => 'User', :foreign_key =>
  'friend_id'

  def xfn_friendship=(friendship_type)
    self.xfn_friend = false
    self.xfn_acquaintance = false
    self.xfn_contact = false

    case friendship_type
    when 'xfn_friend' : self.xfn_friend = true
    when 'xfn_acquaintance' : self.xfn_acquaintance = true
    when 'xfn_contact' : self.xfn_contact = true
    end
  end

  def xfn_friendship
    return 'xfn_friend' if self.xfn_friend == true
    return 'xfn_acquaintance' if self.xfn_acquaintance == true
    return 'xfn_contact' if self.xfn_contact == true
    false
  end

  def xfn_geographical=(geo_type)
    self.xfn_coresident = false
    self.xfn_neighbor = false
    case geo_type
    when 'xfn_coresident' : self.xfn_coresident = true

```

```

    when 'xfn_neighbor' : self.xfn_neighbor = true
    end
end

def xfn_geographical
  return 'xfn_coresident' if self.xfn_coresident
  return 'xfn_neighbor' if self.xfn_neighbor
  false
end

def xfn_family=(family_type)
  self.xfn_child = false
  self.xfn_parent = false
  self.xfn_sibling = false
  self.xfn_spouse = false
  self.xfn_kin = false

  case family_type
  when 'xfn_child' : self.xfn_child = true
  when 'xfn_parent' : self.xfn_parent = true
  when 'xfn_sibling' : self.xfn_sibling = true
  when 'xfn_spouse' : self.xfn_spouse = true
  when 'xfn_kin' : self.xfn_kin = true
  end
end

def xfn_family
  return 'xfn_child' if self.xfn_child
  return 'xfn_parent' if self.xfn_parent
  return 'xfn_sibling' if self.xfn_sibling
  return 'xfn_spouse' if self.xfn_spouse
  return 'xfn_kin' if self.xfn_kin
  false
end
end
end

```

Figure 5 - 12: The Friendship Model Snippet

Also, XFN allows the embedding of information about relationships into the `rel` attribute of an HTML or XHTML anchor tag, adding a human element into the link rather than just a pointer to an URL. For instance, if we want to link to user Rachel with id 9 on *miniOSN*, is a friend of hers, and have met her, we could specify the `rel` attribute as:

```
<a href="http://localhost:3000/users/9" rel="friend met">Rachel</a>
```

This information can then be displayed alongside the links using CSS. Since the markup is very simple and easily understood by both humans and applications, it

is very simple for new applications to be developed using this information. Within the `rel` attribute, a number of types of relationships can be specified, separated by spaces.

Because the `rel` attribute is just part of the normal anchor tag, it is very simple to implement and is transparent to the user.

5.4 *miniOSN* Architecture

As Figure 5-13 shows below, *miniOSN* uses the MVC architecture powered by Rails, handles incoming requests from the client by sending HTTP requests to the Mongrel server, which then forwards the request to the router, the router then finds the appropriate controller that will interact with the model, the model then sends queries to the MySQL database and receives data/error from the database before responding back to the controller, which then invokes the view, telling the view to prepare XML, XHTML and CSS files for the data, and finally, the view sends back the representation of the data to the browser.

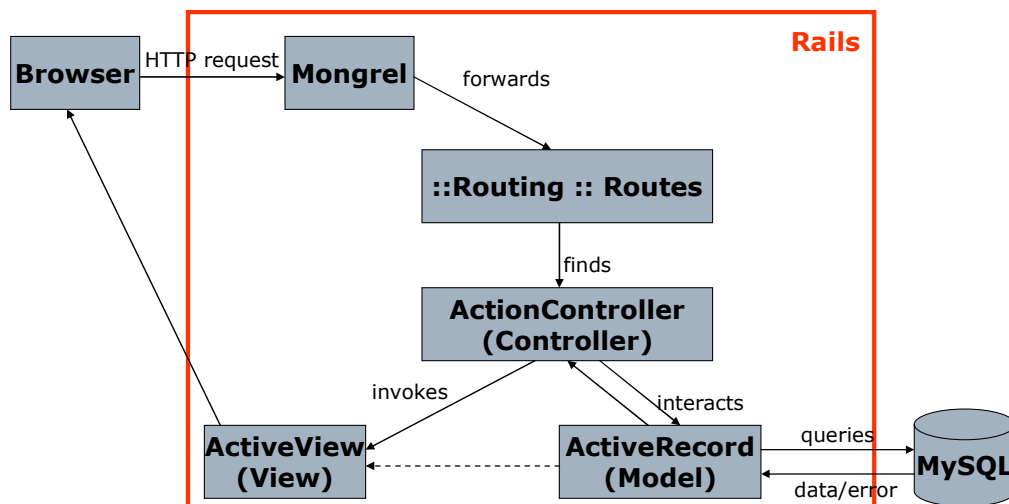


Figure 5 - 13: *miniOSN* Architecture

Mongrel is a fast HTTP library and server for Ruby that is intended for hosting Ruby web applications of any kind using plain HTTP rather than FastCGI or SCGI.

Incoming requests are first sent to a router, which works out where in the application the request should be sent, and how the request itself should be parsed. Ultimately, this phase identifies a particular method, i.e., “action”, somewhere in the controller code. The action might look at data in the request itself, might interact with the model, or it might cause other actions to be invoked. Eventually, the action prepares information for the view, which renders something back to the user.

5.5 Implementation Difficulties

Malfunctioned Ajax and CSS star rating system

Originally, we planned to implement a Rails based Ajax and CSS star rating system [Naffis, 2006], the rating system reuses an elegant CSS only star rating system [Rogie, 2006] and Chris Ingrassia’s `acts_as_rateable` plugin [act_as_rateable plug_in]. So that when a user hovers the mouse over the stars, they can see changes in colours and by clicking on the stars, a user can rate a friend’s trust levels as well as setting trust rating requirements before distributing resources in their online profiles in *miniOSN*.

However, several problems were encountered during the implementation of such a rating system. First of all, the system was limited to one rating per page, secondly, it was limited to one type of object, and most of all, it did not work with objects that were subclassed.

A similar tutorial [Ryan, 2003] from *Midnight Oil* [Midnight Oil] has been investigated carefully, unfortunately, no solutions have been developed to tailor the needs of *miniOSN*. After countless attempts, the idea of integrating such a rating system in the application was abandoned.

5.6 Summary

We have provided several reasons for the decision of using Ruby on Rails for the development of *miniOSN* in this chapter, as well as discussing various technologies used in detail. In addition, we have presented the architecture of the website, and difficulties encountered during the implementation stage.

CHAPTER SIX: EVALUATION

The best way to predict the future is to invent it.

~ Alan Kay

6.1 Introduction

To evaluate the trust mechanism employed in *miniOSN*, we created two scenarios, and compared the performances of *miniOSN* to that of a popular online social network in the created situations, as well as interviewing OSN users on the trust management approach developed. Background information of the scenarios is first provided in the next section, and then the performances of *miniOSN* and *Bebo* are compared in section 6.3. Evaluation interview results and analysis are then presented and discussed in section 6.4. And finally, section 6.5 is a summary of this chapter.

6.2 Scenario Description

In order to find out how well the proposed solution satisfies users' need to express trust subjectively, two situations have been created where trust needs to be modelled asymmetrically, transitively, personally and context-dependently in *miniOSN*.

As mentioned earlier, we have created fictional characters from the situational comedy *Friends* as registered users of *miniOSN*. Ross, Joey and Chandler are three friends connected to each other in *miniOSN*, besides Ross and Joey, Chandler also is connected to a third person Monica.

The scenarios are set with the following background: Ross got three tickets to a New York Rangers game and wanted Joey and Chandler to go along, however Chandler noticed that if they did go to the game, they would not make it back in time for the Thanksgiving dinner which Monica hosts every year, therefore he

tried to persuade the other two not to go themselves either. Although Ross and Joey agreed that it would be a bad idea to go, they secretly went to the game anyway.

6.2.1 Scenario One

Joey took a picture at the game and decides to upload it to his online profile. With two connected friends Ross and Chandler to his profile, Joey knows that if he does not set trust rating requirements for the photo, Chandler would find out that they had gone to the game, however, the picture should not be a secret from Ross. Also, Joey can decide whether other friends of Chandler whom Joey is not connected to should be able to see this photo.

In this Scenario, trust needs to be expressed asymmetrically as well as context-dependently, although Chandler has default trust ratings for Joey, Joey does not feel the same way about Chandler in return in this given situation. Trust is also personalised here, since Ross and Joey would have different trust ratings for their mutual friend Chandler. And finally, trust can be expressed transitively when Joey decides whether Chandler's other friends should see the photo.

6.2.2 Scenario Two

Uploading a photo and posting a blog in *miniOSN* works the very same way, but what happens when a comment is left in someone else's profile?

In the second scenario, Ross posted a blog talking about the Rangers game days before the event, and Joey wants to leave a comment for Ross which concerns the meeting up time, that should certainly be viewable by Ross since it is going to be left in Ross' profile. However, considering Ross, Joey and Chandler are all connected to each other, Joey may suspect that Ross has the default trust ratings for Chandler which would enable Chandler to see everything in Ross' profile. To prevent the comment from being viewable to Chandler, as the owner of that comment, Joey can then set trust rating requirements for this particular resource and stop Chandler from reading the comment.

6.3 Comparison of *miniOSN* and *Bebo* in the Given Scenarios

We are going to take *Bebo* as the representation of notable OSNs due to its popularity with participants who took part in our initial survey, discussed in Chapter Three (see Figure 3-2), and compare performances of *miniOSN* to *Bebo* in the given scenarios. As we are interested in modelling various degrees of trust subjectively among friends of a user, we say that Joey has set his profile “private” in *Bebo*, meaning that only people who are connected to him can see his profile.

In scenario one, once Joey uploads the Rangers game photo in *Bebo*, all of his connected friends would be able to see it. Joey is therefore, forced to grant Ross and Chandler with the same trust level, even though in this situation, Joey does not trust Chandler to view the picture. Hence, trust in *Bebo* can not be expressed asymmetrically, nor can it be tailored to a personalised view depending on the context. And finally, since Joey has set his profile private, Chandler’s friend Monica therefore can not see Joey’s uploaded picture, which means that trust is not transitive.

In scenario two, once a comment is left in Ross’s profile, all of Ross’s connected friends would be able to see it, meaning that both the owner of the comment: Joey, as well as Ross’s connected friend Chandler can view the comment. In *Bebo*, Joey has no way to prevent that from happening.

The two scenarios show that in *Bebo*, users can not express their subjective views of trust among their connected friends. Trust is assumed to be symmetric and non-transitive by the system, there is no such notion of context-specific, let alone any personalisation of expressing the trustworthiness of a particular friend.

With the first scenario in *miniOSN*, trust can be expressed asymmetrically depending on the context, where Chandler may have default trust ratings for Joey, Joey, on the other hand, does not trust his friend the same way in return. He could degrade Chandler’s reliability rating for example and let the trust rating requirements for the photo remain the default values in order to restrict Chandler’s

access to it, but not Ross. When designing our application, we have taken into account with the human feeling factor, and since Chandler does not know the existence of such a picture, he would not have been hurt by it.

As well as expressing trust asymmetrically depending on the context, in *miniOSN*, we could also portrait a personalised view of the trustworthiness of the same person, as the second scenario shows. Ross trusts Chandler with all things in his profile, but Joey thinks otherwise and does not trust Chandler with the comment he left in Ross's profile, by decreasing Chandler's trust ratings, he then prevents Chandler from reading the comment. In this situation, the system obeys Joey's trust requirements for the comment, not Ross's.

And finally, in *miniOSN*, we do not presume that trust is not transitive, we give our users the freedom to express their views on the subject. Whether trust can be transitive or not is user's personal opinion, Joey can choose to let all other Chandler's friends have the same set of trust ratings, meaning that trust is transitive; or he can choose the option "No" when asked "Would you like to apply these values to Chandler's other friends?", by which, he can express that trust is non-transitive if he likes.

6.4 Evaluation Interviews

In order to find out what users think of the design and functionalities of *miniOSN* in relation to expressing various subjective views on trust, we chose to interview several OSN users and gather their opinions on the proposed solution.

We wanted to find out:

- whether there is a desire to express various degrees of trust among connected friends in OSNs in general;
- how well can users of *miniOSN* express the trust characteristics found in the literature review;
- is the proposed rating feature helpful in gaining better control of user profiles and the resources in them;
- can such a rating system hurt friendships;
- and what needs to be refined in our application.

The interviews were held on a one-to-one basis, all questions were open-ended with regards to answers. There are three parts to the interview questions, the first part aimed to find out do users feel the need to express their various levels of trust among their connected friends. The second part of the interview questions concentrated on the given scenarios and gathered volunteer opinions on how well can users in *miniOSN* express their subjective views of trust asymmetrically, transitively, personally and context-dependently. And finally, we asked participants how they felt about the proposed trust management solution that is integrated in *miniOSN* and possible refinement of its features.

The complete evaluation interview questions are listed in Appendix II.

6.4.1 Volunteer Background

A total of nine volunteers took part in our evaluation interview, all of whom are familiar with the concept of OSNs. Four candidates are from M.Sc. in Computer Science – Networks and Distributed Systems programme, three others are from the Knowledge and Data Engineering Ph.D. research group, with the remaining two participants from non-technical background.

6.4.2 Results

From the first part of the questionnaire, seven interviewees stated that they did not trust their connected friends equally, hence, they felt the need to express their various levels of trust among these friends in OSNs. However, two other participants found such a feature was not really necessary for them since they only used OSNs irregularly, so that they could keep in touch with people they hardly have any time for, and most of all, they did not maintain their profiles with many resources and therefore, felt that there really was no need to distinguish one friend from another since nothing was meant to be a secret from anybody.

When asked whether they could relate to, or imagine situations where they wished they could have had a way to decide whomever friend to see whatever resources in their profiles whenever they wanted these friends to, except one person, eight

candidates stated that such situations were inevitable in OSNs they had experienced.

All of the interviewees felt that in the given scenarios, users could express trust asymmetrically and personally. However, in the second part of the questionnaire, we found that when asked whether participants felt that they could express trust transitively depending on the context, candidates had contradictive views on it.

Although three people felt that users in *miniOSN* could express whether they thought trust was transitive or not, six others felt that this arguable characteristic of trust was not modeled well. At the moment, users in *miniOSN* can only choose one of the two given options when asked whether they would like to apply a same set of trust ratings for all other friends of a connected friend, in other words, they can only state either they want to set the exact same trust level to every friend of the connected friend, or have no trust at all towards these people. This is a very clear-cut approach, people who believe that trust level decreases as links between connected nodes grow longer can not express such a view in *miniOSN*.

And finally, can trust be expressed context-dependently in *miniOSN*? Seven candidates felt that this is indeed the case. However, two participants disagreed, stating that the notion of trust is context-specific was not captured well. For example, imagine that Rachel has a picture that she only wants her partner Ross to see, and not by her family members, however, she happens to have the same set of trust ratings for both Ross and her mother Sandra. So in this case, if Rachel wants to restrict Sandra's access to that certain picture, she will have to degrade her mother's trust ratings, which would then trigger a chain of events, as a result, Sandra would no longer have access to all other previous resources in Rachel's profile.

In the third part of the interview, we concentrated on the trust management mechanism used in *miniOSN*, namely, the trust rating feature, and gathered participants' attitude towards such a proposed solution.

Seven interviewees felt that such a feature was not inhumane since the owner of the profile is the only person to see it. And interestingly, one person mentioned that even if such ratings are visible to others, still he would not feel it is inhumane in any way, stating that it is possible to hurt one's feelings, but this should not stand in the way of new technologies.

We found that the eight trust attributes, reliability, honesty, credibility, reputation, competency, belief, faith and confidence were very confusing to many. Several people mentioned that they found it hard to understand the concepts, as well as distinguishing them from each other when “not only the attributes overlap each other, the differences among them are so subtle”. For example, three people mentioned that they simply could not tell faith and belief apart, and two people felt that “competency” seemed out of place in an OSN environment, since it is very business like. Also, we found in our interviews that the number of trust attributes were overwhelming to many. Candidates suggested that it is possible to express trust with just three or four of those attributes instead of having all eight of them, which would in turn help them see the differences in the concepts and make full use of their understanding of them.

How did our candidates feel about associating numbers with the eight trust attributes then, was it easy for them to do? Again, we had a similar finding here, where four interviewees felt that it was easy for them, the majority of our participants felt that since understanding the concepts were so hard for them, consequently, they found it difficult to give numbers for them. Also, two people suggested using visual aid such as sliding bars for the rating system instead of using numbers.

Overall, seven volunteers felt that the trust mechanism implemented in *miniOSN* did help users express various degrees of trust, and it also helped users to gain a better control over their resources in online profiles. However, it was mentioned that the rating system in *miniOSN* seemed to be over-complicating the situation, especially when it is difficult to understand the attributes for a start, several candidates felt that it would be just as efficient and effective if users of a profile can simply specify which friend should see what resources without having to go through such work load by giving rating matrix for every one.

6.4.3 Analysis

From the evaluation interviews, we have found that most people would like to express their subjective views of trust among connected friends depending on the

context in the OSN environment, and many felt that the proposed solution would help users gain a better control over the resources in online profiles. However, refinement is necessary on the design and user interface.

From a design point of view, we need to consider issues such as manageability. At the moment, in *miniOSN*, we are relying on users to keep track of all trust ratings of their connected friends, which works well on a one-to-one basis, however, when one has to manage a large number of friends, it becomes difficult for the user to keep track of various sets of numbers.

Also, user interface needs attention. We could implement features where once a set of trust rating requirements has been set for a certain resource, users should then be notified with a list of connected friends who do have access to the resource, in order to avoid mistakes caused by human error. Besides such a function, it would also be convenient to let users clearly see a list of all their connected friends and their given trust ratings, for easy comparison and readjusting.

The proposed solution addresses the problem of a lack of personalisation when modeling trust in OSNs, however, a common view that trust level decreases as the link between nodes grow longer is not being captured well at the moment. Also, as mentioned in the previous section, what happens when a person's trust rating is readjusted? Currently, such an event would result in restricted access to previous resources, which clearly, is a major problem that needs to be solved.

Finally, we need to take into account of the limitations of our interviews, since most candidates are from the discipline as the author as well as Quinn who is the owner of the multi-faceted model of trust. Due to this background, such close associations may not be well positioned to provide objective views on the trust management approach applied in *miniOSN*.

6.5 Summary

In this chapter, we have evaluated the trust mechanism used in *miniOSN* by comparing its performance to a popular online social network in the given

scenarios as well as collecting user opinions from one-to-one interviews. We have found that the proposed solution has a better control over user profiles and resources in them, however, it also has several limitations, therefore, refinement of the design and functionalities require further study. In addition, we have shown that the proposed solution has received mixed opinions regarding its trust management approach.

CHAPTER SEVEN: CONCLUSIONS

A journey of a thousand miles begins with a single step.

~ Lao-tzu

7.1 Project Summary

The initial goal of this research project was to find out whether users are happy with current trust mechanisms employed in OSNs, and whether a multi-faceted model of trust that is personalisable and specialisable would be welcomed in OSNs, and most of all, would the proposed solution address issues such as expressing trust asymmetrically, personally, transitively and context-dependently.

From the state of the art studies, we have found that various notable OSNs employ simple, “one-size-fits-all” access control methods that ignore important characteristics of trust, and we suspect that these current trust mechanisms can not cater for the online community’s needs of expressing trust freely and subjectively. Hence, we designed our initial survey to find out if our suspicion was reasonable.

From the initial survey, we have gathered evidence that current trust mechanisms have not achieved user satisfaction, and there is strong support for a better trust model. However, a rating system has received mixed opinions, where some supports such a feature in OSNs, others strongly disagree. Since the findings of many questions were contradictive, we concluded that such an outcome maybe due to the limitations of the questionnaire since by their very nature, survey questions are open to interpretation, and it would be too early to abandon the proposed solution.

Therefore, in order to determine how the proposed solution would be truly received by the online community, we designed and implemented *miniOSN*, by which, we aim to solve several problems that have not been addressed by current trust mechanisms in OSNs, as well as gathering user opinions on the proposed solution.

From the evaluation of the trust management system in *miniOSN*, we have found that users have mixed opinions about the rating feature. Most people find the eight trust attributes confusing and difficult to understand, several refinements have been pointed out in relation to the site design as well as the user interface. But overall, the majority of people felt the rating system provides users with a better and flexible control over resources in their online profiles.

7.2 Contribution

During this study, several valuable findings have been discovered.

First of all, our initial survey has gathered a large amount of data regarding OSN user behaviours, their opinions of current trust mechanisms deployed in notable OSNs, and most importantly, there were strong indications that OSN users are not satisfied with current trust management systems and a better trust model that enables users to express trust subjectively is welcomed.

Secondly, with support for a refined model of trust as we have found in the initial survey, and influences from Quinn's multi-faceted model of trust that is personalisable and specialisable, we have designed and implemented a small scale OSN with a trust rating system that operated well as a proof of concept to gain potential user feedback. In addition, this real world application of the model sets foundation for further studies in the topic area.

Thirdly, our study of the integration of the proposed multi-faceted model of trust in an OSN environment has received mixed reviews. Several limitations of the design as well as refinement on the model have been noted in our research. We suspect that such a proposed model would work well in an e-market environment, where users do not have previous relationships offline and are building trust for each other from scratch; however, in an OSN environment, the situation is different, most users have already built friendships and trust for each other, although most of our evaluation candidates feel the rating system indeed helps users gain a better control over resources in a profile, many felt the trust attributes

confusing, which poses further research question of how could we refine the proposed model of trust for better user interactions.

And finally, we have also noted limitations of the proposed solution as listed below:

Unsolved chain-reaction problem

Once one's trust ratings have been brought down to a set of lower scores, this person therefore, can no longer access resources that were uploaded previously with higher trust rating requirements. Currently, the only way to let them have access to previous resources is by readjusting trust rating requirements for the resources, which, as a consequence, would then result in having restricted access for all other people who did have access to some of these resources previously. And to solve the same problem for all these other people, the same circle repeats. Therefore, to eventually have everything organised, becomes hectic work and trouble for the average user.

Management issue

As mentioned before in section 6.4.3, at the moment, the system is relying on the users to keep track of the sets of ratings they may have given to their friends, which can become overwhelming if one has many friends.

Limitation when expressing trust transitively

From the literature review, we have found that it is arguable to state that trust is transitive. Some say that trust is non-transitive, others believe that as the link between the connected nodes grow longer, trust level decreases. In our application, users can choose whether or not they would like to apply a same set of trust values for all other friends of a connected friend, and this is the only way to model this arguable characteristic of trust. We have not implemented features to let users to express a degrading trust level in our design.

7.3 Future Work

To continue further studies of the proposed solution, attention needs to be paid to refining features in *miniOSN* as discussed in Chapter Six, as well as addressing the limitations of the model as mentioned in the previous section.

Besides these areas, several other possibilities are discussed here:

Reduce the number of trust attributes

As our findings in the evaluation interviews suggest, many feel that the number of trust attributes are overwhelming. Instead of having eight attributes, could we reduce them to a total of four possibly? As we recall from the initial survey (see Figure 3-11), honesty was most valued, followed by credibility, reliability and reputation. Are just these four attributes enough to define the broad and subjective views of trust? The answer to this question lies with the users, more surveys and experiments need to be carried out.

Choose collections of trust attributes and assign priorities to them

By reducing the number of trust attributes, we risk restricting ourselves to a limited design right from the start. Hence, instead of cutting down on the number of trust attributes, another alternative is to let users decide what trust means individually, so that each user can have their very own model of trust. For example, trust, in one person's opinion, can mean a combination of credibility, honesty, reputation and confidence, while for another, trust may stand for competency and reliability. As well as letting users to choose collections from the given eight trust attributes, we can also let users to decide which ones are more important than others, by assigning priorities to them, which means, that users would have the freedom of expressing trust in a certain individual depending on the context. For instance, a friend's reliability is more important than his/her honesty in certain scenarios, while at other times, competency is valued more than reputation.

By having such features, the application would achieve better personalisation when modelling trust in the OSN environment, where users can tailor their needs by creating different models of trust in the same individual for different situations,

and their various levels of trust as well as different kinds of trust among connected friends.

Make trust ratings visible to others

Our application has been hugely influenced by the findings from the initial survey, where a strong disliking of the rating feature was presented, mainly due to worries of hurting friendships and concerns of the possibility of encouraging online bullying behaviour. Therefore, in our design, we have decided to let ratings for others be viewable by the owner only.

However, at the stage when we carried out our initial survey, participants had very little idea of the functionalities of such a rating system, and without any knowledge on the proposed solution, they have developed a sense of insecurity. As we recall from the evaluation interviews, one person stated that even if trust ratings are visible to others, he still thought it would do no harm and had expressed a strong interest in using such features in OSNs if they were to be deployed.

Therefore, we foresee experiments and the application of an OSN where ratings are visible to others. And we believe that comparison of the two websites could bring us to some very interesting findings and conclusions.

Security issues

Database design requires care, a secure system where confidentiality, integrity and availability concerning trust ratings must be built as we develop our application. If ratings were to stay private, it is possible that malicious users would be interested to find out what ratings one has given another for example. We must prevent them from hacking into the database, either tempering with the values in the database or distributing such information to third parties.

Find out user requirements

Since OSNs are user-centric by their very nature, going back to our customers for guidance is always a good idea. Further studies on trust management in OSNs can look into the broad population and find out what users really want when expressing trust in OSNs.

Real world deployment and case studies

Finally, it would be very interesting to see user reactions to a refined model of trust. Hence, development of a real world application would further this study and we can then gather a much larger audience for the evaluation of the proposed solution.

There is certainly, a huge scope for the continuation of research in the area of trust management in OSNs. Case studies could be designed to tackle various issues discussed in this thesis. Besides experimenting with online communities in the broad Internet context, it would be also very interesting to address issues with OSNs used in organizations and centre around particular user requirements in a work setting.

7.4 Final Remarks

On the 6th of September, 2007, BBC News reported that, *Facebook* has added a public-facing search which would “initially allow anyone who is not registered with the site to search for a specific person. More controversially, in a month's time, the feature will also allow people to track down *Facebook* members via search engines such as Google” [BBC News, 2007]. According to BBC, this public search listing will show the thumbnail picture of a *Facebook* member as well as links allowing others to interact with them. However, in order to add someone as a friend or send them a message, the person must be a registered member. Users who want to restrict what information is available to the public or opt out of this listing feature can change their privacy settings in the coming month.

Critics have expressed concerns and disappointment at the move. Technology writer Om Malik commented that the move “transforms *Facebook* from being a social network to being a quasi-White Pages of the web” in his blog GigaOm [GigaOm]. Mr Malik, and others, are concerned about the data trail that people are routinely leaving behind them on social networking and other sites.

The author believes that such a move highlights the importance of having complete control of online profiles in OSNs, especially the resources presented in

them. Once such a public listing feature is in place, *Fackbook* would then assume that once such a function is enabled in a profile, the registered user therefore would trust all random strangers equally to access the information specified.

However, this is not the case, as we have found in our literature review. Since trust is always closely associated with a broad context, it is difficult to say that such a feature with little flexibility would lead to user satisfaction in relation to expressing a diverse range of views on trust depending on the context.

The author's vision for the future of managing trust in online social networks is the deployment of innovative trust management approaches such as a multi-faceted model of trust that is personalisable and specialisable, where users can express their opinions and understandings of trust and annotate trust accordingly.

Finally, it is the opinion of the author that modelling trust will remain one of the most pressing issues in computer science, which poses significant challenges of capturing the elusive notion of trust in computer systems, especially in the new and relatively unstructured online social network environment. Trust mechanisms that are able to cater diverse user needs in online social networking websites require focused attention from computer scientists.

BIBLIOGRAPHY

- [37signals] 37signals website, <http://www.37signals.com>
- [43Things] 43Things website, <http://www.43things.com>
- [Abdul-Rahman, 2004] Abdul-Rahman, A., "A Framework for Decentralised Trust Reasoning", Ph.D. Thesis, University of London, UK. 2004.
- [acts_as_rateable plugin] acts_as_rateable plugin project website, <http://rubyforge.org/projects/rateableplugin>
- [Amazon] Amazon website, <http://www.amazon.com>
- [aSmallWorld] aSmallWorld website, <http://www.asmallworld.net>
- [attachment_fu plugin] attachment_fu plugin project website, retrieved from http://svn.techno-weenie.net/projects/plugins/attachment_fu
- [Barnes, 1967] Barnes, J. A., "Politics in a Changing Society: a political history of the fort Jameson Ngoni", Manchester: Manchester University Press, 1967.
- [Basecamp] Basecamp website, <http://www.basecamphq.com>
- [BBC News, 2007] "Facebook opens profiles to public", September, 2007, retrieved from <http://news.bbc.co.uk/2/hi/technology/6980454.stm>
- [Bebo] Bebo website, <http://www.bebo.com>
- [Bradburne, 2007] Bradburne, A., 2007, "Practical Rails Social Networking Sites", Apress.
- [CarDomain] CarDomain website, <http://www.cardomain.com>
- [Chu et al, 1997] Chu, Y., Feigenbaum, J., LaMacchia, B., Resnick, P., and Strauss, Ma., 'REFeree: Trust Management for Web Applications.', *The World Wide Web Journal*, 1997, 2(3), pp. 127-139.
- [Dey, 2001] Dey, A., 2001, "Understanding and Using Context", *Personal and Ubiquitous Computing* 5(1): 4-7.
- [DCU] Dublin City University website, <http://www.dcu.ie>
- [Doostang] Doostang website, <http://www.doostang.com>
- [DIT] Dublin Institute of Technology website, <http://www.dit.ie/DIT/Homepage>
- [Dumbill et al, 2002] Dumbill, E., 'XML Watch: Finding friends with XML and RDF.', *IBM Developer Works*, June 2002, retrieved from <http://www-106.ibm.com/developerworks/xml/library/xfoaf.html>
- [eBay] eBay website, <http://www.ebay.com>
- [Ecademy] Ecademy website, <http://www.ecademy.com>

- [Facebook] Facebook website, <http://www.facebook.com>
- [Friends] Situational Comedy Friends website, Warner Bros,
<http://www2.warnerbros.com/friendstv>
- [Friends Reunited] Friends Reunited website, <http://www.friendsreunited.com>
- [Friendster] Friendster website, <http://www.friendster.com>
- [GigaOm] GigaOm website, <http://gigaom.com>
- [Gil et al, 2002] Gil, Y., Ratnakar, V., ‘Trusting Information Sources One Citizen at a Time’, Proceedings of the First International Semantic Web Conference (ISWC), Sardinia, Italy, June 2002.
- [Golbeck, 2005] Golbeck, J. A., 2005, “Computing and Applying Trust in Web-Based Social Networks”, Ph.D. thesis, University of Maryland.
- [Graduates] Graduates website, <http://graduates.com>
- [Grandison, 2003] Grandison, T., “Trust Management for Internet Applications”, Ph.D. thesis, University of London, UK. 2003.
- [Grandison et al, 2001] Grandison, T., Sloman, M., ‘SULTAN - A Language for Trust Specification and Analysis’, Proceedings of the 8th Annual Workshop HP Open View University Association (HP-OVUA), Berlin, Germany, June 24-27, 2001.
- [Grandison & Sloman, 2000] Grandison, T., and Sloman, M., 2000, “A survey of trust in internet applications”, IEEE Communications Surveys and Tutorials, 4(4):2–16.
- [Gray, 2006] Gray, E. L., 2006, “A Trust-Based Management System”, Ph.D. thesis, Department of Computer Science and Statistics, Trinity College, Dublin.
- [Hauben & Hauben, 2004] Hauben M., and Hauben R., on-line Netbook, “Netizens: On the History and Impact of Usenet and the Internet”, retrieved from <http://www.columbia.edu/~hauben/netbook>
- [Hi5] Hi5 website, <http://www.hi5.com>
- [Hitwise] Hitwise website, <http://www.hitwise.com>
- [Hitwise Data Centre, 2006] “Hitwise: One in 20 Web Visits Go to Social-Networking Sites”, 2006, retrieved from http://www.marketingvox.com/archives/2006/11/09/hitwise_one_in_20_web_visits_go_to_socialnetworking_sites
- [Hitwise Data Centre, 2007] “Top 20 Websites - August, 2007”, retrieved from <http://www.hitwise.com/datacenter/rankings.php>
- [Information Today database, 2007] Information Today, 2007, retrieved from http://www.infotoday.com/searcher/jul07/Reid_Grey_WebsitesTable.pdf
- [ImageMagick] ImageMagick Project website, <http://www.imagemagick.org>
- [Instant Rails] Instant Rails homepage, <http://instantrails.rubyforge.org>

- [Jøsang A., 1996] Jøsang A., 1996, “The right type of trust for distributed systems”, Proceedings of the 1996 workshop on new security paradigms. Lake Arrowhead, California, United States, ACM Press.
- [Kawamoto & Sandoval] Kawamoto D., and Sandoval G., “MySpace growth continues amid criticism”, CNET News.com, March 31, 2006, retrieved from http://m.news.com/MySpace+growth+continues+amid+criticism/2163-1025_3-6056580.html
- [KDEG] Knowledge and Data Engineering Group website, <http://kdeg.cs.tcd.ie>
- [LinkedIn] LinkedIn website, <http://www.linkedin.com>
- [Marsh, 1994] Marsh S., 1994, “Formalising Trust as a Computational Concept”, Ph.D. thesis, Department of Mathematics and Computer Science, University of Stirling.
- [MOG] MOG website, <http://mog.com>
- [Microformats] Microformats project website, <http://microformats.org>
- [Midnight Oil] Midnight Oil website, <http://blog.aisleten.com>
- [Mui et al., 2002] Mui, L., Mohtashemi, M., and Halberstadt, A., 2002, “A computational model of trust and reputation”, In Proceedings of the 35th International Conference on System Science, pages 280–287.
- [MySpace] MySpace website, <http://www.myspace.com>
- [Naffis, 2006] Naffis, D., 2006, “Rails based Ajax and CSS star rating system”, retrieved from <http://www.naffis.com/2006/8/31/rails-ajax-star-rating-system>
- [NUI, Galway] National University of Ireland, Galway website, <http://www.nuigalway.ie>
- [Olmedilla et al., 2005] Olmedilla, D., Rana, O., Matthews, B., and Nejdil, W., 2005, “Security and trust issues in semantic grids”, In Proceedings of the Dagstuhl Seminar, Semantic Grid: The Convergence of Technologies, volume 05271.
- [Plaxo] Plaxo website, <http://www.plaxo.com>
- [Quinn, 2006] Quinn K., 2006, “A Multi-faceted Model of Trust that is Personalisable and Specialisable”, Ph.D. thesis, Department of Computer Science and Statistics, Trinity College, Dublin.
- [Rails] Ruby on Rails website, <http://www.rubyonrails.org>
- [Ralph, Alessandro et al. 2005] Ralph, Alessandro et al., 2005, “Information revelation and privacy in online social networks”, Proceedings of the 2005 ACM workshop on Privacy in the electronic society. Alexandria, VA, USA, ACM Press.
- [Reid & Gray, 2007] Reid M., and Gray C., “Online Social Networks, Virtual Communities, Enterprises, and Information Professionals - Part 1. Past and Present”, August, 2007, retrieved from http://www.infoday.com/searcher/jul07/Reid_Grey.shtml
- [Rmagick] Rmagick Project website, <http://rmagick.rubyforge.org>

- [Rogie, 2006] Rogie, 2006, "CSS Star Rating Part Deux", retrieved from <http://komodome.com/blog/index.php/2006/01/09/css-star-rating-part-deux>
- [RoR, 2007] Ruby on Rails wiki website, <http://wiki.rubyonrails.org/rails/pages/Plugins>
- [Ruby] Ruby Programming Language website, <http://www.ruby-lang.org>
- [Ryan, 2003] Ryan, 2003, "Ajax CSS Star Rating with acts_as_rateable", retrieved from http://blog.aisleten.com/2007/05/03/ajax-css-star-rating-with-acts_as_rateable
- [Shneiderman, 2000] Shneiderman B., "Designing trust into online experiences". Commun. ACM 43(12): 57-59.
- [SurveyMonkey] Survey Monkey website, <http://www.surveymonkey.com>
- [Teten & Allen, 2005] Teten D., and Allen S., "The Virtual Handshake: Opening Doors and Closing Deals Online", AMACOM/American Management Association
- [TCD] Trinity College Dublin website, <http://www.tcd.ie>
- [UCC] University College Cork website, <http://www.ucc.ie>
- [UCD] University College Dublin website, <http://www.ucd.ie>
- [USA Today, 2006] USA Today. "YouTube serves up 100 million videos a day online", 2006, retrieved from http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm
- [USENET] USENET website, <http://www.usenet.com>
- [Vannevar, 1996] Vannevar, B., 1996, "As we may think." interactions 3(2): 35-46.
- [XFN] XHTML Friends Network microformat project website, <http://www.gmpg.org/xfn>
- [XING] XING website, <http://www.xing.com>
- [Yahoo!360] Yahoo!360 website, <http://360.yahoo.com>
- [YouTube] YouTube website, <http://youtube.com>
- [Zimmermann, 1994] Zimmermann, P., "PGP(tm) User's Guide", October 1994.
- [Zimmerman, 1995] Zimmerman, P.R., "The Official PGP Users Guide", MIT Press, Cambridge, MA, USA, 1995.

APPENDICES

APPENDIX I – A Survey of Online Social Networks

The anonymously collected data sets for this survey can be found on the CD-ROM that accompanies this thesis, in the folder named “Initial Survey”.

Section 1. Introduction

Thank you for taking 3-5 minutes of your time to complete the following questionnaire. In recent years, we have seen dramatic increase of Online Social Networks (for example, Bebo; MySpace; etc.) that offer users the freedom to create their very own profiles with the intention of socializing with friends; making connections; finding jobs; entertainment and much more. This survey aims to research user opinions of Online Social Networks (OSNs from here after) and in particular current methods of controlling access to user profiles.

The information you provide is greatly appreciated and completely confidential; data gathered will not be shared with third parties and shall be destroyed in November 2007. Click "Next" to get started with the survey. If you'd like to leave the survey at any time, just click "Exit this survey". Your answers will be saved.

If you have any questions in relation to this survey, please do not hesitate to contact bofu@cs.tcd.ie

Section 2. Participant Background

Please select the option that best describes your status.

1. Please tell us your age:

under 20

21-25

26-30

31-40

over 40

rather not say

2. Please specify your gender:

male

female

rather not say

3. Please tell us your standing in college:

undergraduate

postgraduate

staff

rather not say

4. Please tell us your technical background:

Studying for/have a degree in engineering/computer science/information technology related field

Have interest and some knowledge in engineering/computer science/information technology related field

Have little knowledge in engineering/computer science/information technology related field

5. Do you have any experience in using OSNs?

Yes, I am currently using OSNs

Yes, I have in the past, but I am no longer active with OSNs

No, I have never used OSNs

Section 3. Have Used OSNs In The Past - But No Longer Active

Please select ALL options that describe your opinions.

6. Why have you stopped using OSNs?

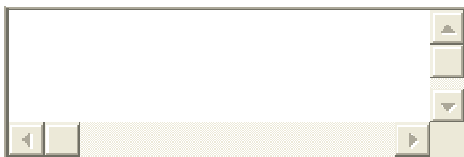
Don't have time for them

Lost interest in them

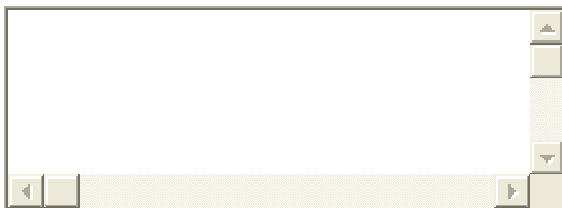
Don't like having personal information on the Internet

Not happy with their services

Other (please specify)



7. If you have chosen "not happy with their services" in the above question, please specify what features weren't you happy with:



Section 4. Multi Choice Questions - People Currently Using/Have Used OSNs

Please select ALL options that describe your opinions.

8. Please select all OSNs you have had experience with:

Bebo

Classmates.com

Facebook

Flickr

Friendster

Hi5

LinkedIn

LiveJournal

MySpace

Orkut

Windows Live Spaces (formerly MSN Spaces)

XING

Yahoo! 360

YouTube

Other (please specify)

9. Based on your experience, what features would you like to see in OSNs that are not in place at the moment?

To rate movies for example

To rate credibility of people/their opinions for example

To rate reliability of people/information they give for example

To rate reputation of someone in specific situations for example

To rate competency of someone for example

To find out the ratings others might have given me/my opinions for example

To find out the ratings others might have given to someone I know of for example

Other (please specify)

10. How did you become a member of the OSNs you are currently using or have used?

Joined voluntarily

Invited by friends

Peer pressure

Other (please specify)

11. Why do/did you use OSNs?

Socialise with people you know offline

Make new friends online

Educational purposes

Business oriented

Find jobs

Dating

Killing time

Peer pressure

Other (please specify)

Section 5. Single Choice Questions- People Currently Using/Have Used OSNs

Please select the option that best describes your status.

12. Which OSN do/did you visit most often?

- Bebo
- Classmates.com
- Facebook
- Flickr
- Friendster
- Hi5
- LinkedIn
- LiveJournal
- MySpace
- Orkut
- Windows Live Spaces (formerly MSN Spaces)
- XING
- Yahoo! 360°
- YouTube
- Other (please specify)

13. How often do/did you visit OSNs?

- Every day
- Several times a week
- Several times a month
- Hardly ever

14. Is/was your profile viewable to:

- People directly linked with you
- Only some of your directly linked friends
- Other friends of your directly linked friends
- Anyone

15. Among OSNs you are using/have used, are you happy with current ways of controlling access to your profile/certain contents in your profile?

- Yes
- No
- Don't care

16. If you have chosen "No" in the above question, please tell us what would you like to do:

Section 6. People Who Have Never Used OSNs

Please select ALL options that describe your opinions.

17. If you have never used OSNs, please tell us why:

- Have never heard of OSNs
- Not interested in using OSNs
- Don't have time
- Don't want to put personal things on the internet
- Other (please specify)

18. If you were to use OSNs, what would be your reasons?

- Socialise with people you know offline
- Make new friends online
- Educational purposes
- Business oriented
- Find jobs
- Dating
- Killing time
- Peer pressure
- Other (please specify)

Section 7. Opinions of Trust in Online Social Networks - PART ONE

Please select the option that best describes your opinion.

19. Do you think you will continue using/start to use OSNs in the future?

- Yes
- No

20. When using OSNs, would you like to be able to set your profile viewable to:

- Anyone
- Only the trusted ones of my directly linked friends
- All my directly linked friends
- Any other friends of my directly linked friends
- Doesn't bother me whichever way

21. Do/would you consider people you have met in OSNs as your friends?

- Yes, some of them
- No, it is difficult to trust them

22. Do/would you trust your directly linked friends to view all parts of your online profile?

- Yes, all these friends
- Yes, but only some of these friends

No

Don't care

23. Do you think it is necessary that only certain people can view certain parts of your profile?

Yes

No

Don't care

24. Do/would you trust other friends of your directly linked friends to view your online profile?

Yes, all of these other friends of my directly linked friends

Yes, but only some of these other friends of my directly linked friends

No

Don't care

25. Do/would you trust random strangers to view your online profile?

Yes

No

Don't care

Section 8. Opinions of Trust in Online Social Networks - PART TWO

26. Would you feel comfortable that your ratings of a certain friend are visible to this particular person?

Yes

No

Don't care

27. Would you feel comfortable that your ratings of a certain friend are visible to others but NOT to this certain friend?

Yes

No

Don't care

28. Would you like to see the ratings your friends have given you in OSNs?

Yes

No

Don't care

29. Would you like to see the ratings given to other friends of your directly linked friends whom you have not met?

Yes

No

Don't care

30. Do you think controlling access to your profile/certain content of your profile is necessary in OSNs?

Yes

No

- Don't care
31. Would you like to rate how much you trust your friends?
- Yes, that would be helpful
- No, that is not necessary
- Don't care
- Undecided
32. If you could rate the following of a person in OSNs, which one/ones are most important in your opinion?
- Credibility
- Honesty
- Reliability
- Reputation
- Competency
- Belief
- Faith
- Confidence
33. If you could rate your friends in OSNs, would you be bothered doing so?
- Yes, I would
- No, I wouldn't
- Don't care

Section 9. Follow-up Questionnaires

34. Would you be interested in taking part in follow-up questionnaires of OSNs like this one in the next two months?

- Yes
- No

35. If you have chosen "yes" in the above question, please tell us your email address:

APPENDIX II – Evaluation Questionnaire

Introduction

In most notable online social networks (OSNs), users can group their connected friends into categories and grant specified access to these identified categories, however, users can not differentiate various levels of trust among members of these categories. *MiniOSN* takes a flexible and personalisable approach, and allows users grant access to their resources however they desire and whomever they would like to based on ratings of credibility, reliability, reputation, honesty, competency, belief, faith and confidence trust attributes of a connected friend.

In this questionnaire, we aim to find out does the rating feature implemented in *MiniOSN* satisfy user expectations in relation to personalising trust.

Part One – General information

1. Do you know what services the likes of *Bebo/MySpace/Facebook* provide?

2. Do you trust all your connected friends equally?
3. Do you feel the need to define various trust levels among connected friends since some of them are simply people whom you do not dislike?
4. Have you encountered situations, or could you imagine situations where you wish you could have had a way to decide whomever to see whatever resources in your profile whenever you want them to depending on the situation?

Part Two – Evaluation of the Scenarios

5. In the given scenarios, do you feel that you can express trust asymmetrically in *MiniOSN*?
6. In the given scenarios, do you feel that you can express trust transitively in *MiniOSN*?
7. In the given scenarios, do you feel that you can express your personalised view of trust subjectively?
8. In the given scenarios, do you feel that you can differentiate trusting a person and a person's actions in *MiniOSN*?

Part Three – About MiniOSN

9. Do you feel the rating feature is inhumane in *MiniOSN*?
10. Are the eight trust attributes: credibility, reliability, reputation, competency, honesty, faith, belief and confidence confusing?
11. Which of these trust attribute(s) is/are unclear to you?
12. Is it easy for you to quantify the eight trust attributes?
13. In *MiniOSN*, you can express various degrees of trust among connected friends, do you find this feature helpful in terms of gaining control over resources in your profile?
14. If answered “no” in the above question, what are your reasons?
15. Any suggestions in relation to expressing trust in *MiniOSN*:
16. Other comments: