VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belgaum-590014

A PROJECT REPORT ON

"MULTIPARTY ACCESS CONTROL FOR ONLINE SOCIAL NETWORKS: MODEL AND MECHANISMS"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING
IN

COMPUTER SCIENCE & ENGINEERING

BY

PREMPRAKASH KASHYAP

(1CD09CS067)

UNDER THE GUIDANCE OF

Mrs. SHILPA B

Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CAMBRIDGE INSTITUTE OF TECHNOLOGY, BANGALORE-560036

2012-2013

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K. R. PURAM, BANGALORE-560036

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Project work entitled **"MULTIPARTY ACCESS CONTROL FOR ONLINE SOCIAL NETWORKS: MODEL AND MECHANISMS"** is a bonafied work carried out by **PREMPRAKASH KASHYAP (1CD09CS067),** in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering under **Visvesvaraya Technological University**, **Belgaum** during the year 2012-13. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the Report. The Project Report has been approved as it satisfies the academic requirements with respect of Project Work prescribed for the said Degree.

Dr. B V Ravishankar	**Dr. Suresh L**	**Mrs. Shilpa B**
Principal	Prof & Head, Vice Principal	Asst. Prof,
CiTech	Dept. of CSE,CiTech	Dept.CSE,CiTech

Name of the Examiners **Signature with Date**

1)

2)

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K. R. PURAM, BANGALORE-560036.

DECLARATION OF THE CANDIDATE

I, **PREMPRAKASH KASHYAP (1CD09CS067),** hereby declare that the Project Work entitled **"MULTIPARTY ACCESS CONTROL FOR ONLINE SOCIAL NETWORKS: MODEL AND MECHANISMS"** has been carried out under the supervision of our Guide **Mrs. SHILPA B, Asst.Prof., Department of CSE, CiTech** in partial fulfillment of requirements for the award of degree of B.E.(CSE) submitted in the Department of **Computer science and Engineering** of **CAMBRIDGE INSTITUTE OF TECHNOLOGY** under **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, KARNATAKA** is a compiled work carried out during a period from February 2013 to May 2013.

Signature of the Student

Premprakash Kashyap

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

_____ _____

Signature of the GUIDE
Mrs. SHILPA B

Signature of the Project
Co-ordinator

The Project Review was held on __ /__/ 2013 and is accepted.

ACKNOWLEDGEMENT

I would like to express my deepest sense of gratitude to **Shri D. K. Mohan** Chairman, Cambridge Group of Institution, Bangalore, India for providing excellent Infrastructure and Academic Environment at CITECH without which this work would not have been possible.

I am extremely thankful to **Dr. B. V. Ravi Shankar**, Principal, CITECH, Bangalore, for providing me Academic ambience and Laboratory facilities to work in, and everlasting motivation to carry out this work and shaping our careers.

I express my sincere gratitude to **Dr. Suresh L**, HOD, Dept. of Computer Science and Engineering, CITECH, Bangalore, India, for his stimulating guidance, continuous encouragement and motivation throughout the course of project work.

I express my sincere gratitude to Prof. **Dr. Shashikumar D.R.** HOD, Dept. of Information Science and Engineering, CITECH, Bangalore, India, for his stimulating guidance, continuous encouragement and motivation throughout the course of project work.

I also wish to extend my thanks to our project guide **Mrs. Shilpa B,** Assistant Professor, Dept. of Computer Science and Engineering, CITECH, Bangalore, India, for her critical, insightful comments, guidance and constructive suggestions to improve the quality of this project work.

I like to thank all the faculties and staffs of Department of CSE.

I take this opportunity to thank all my friends, classmates who always stood by me in difficult situations, also helped me in some technical aspects and last but not the least I wish to express deepest sense of gratitude to my parents who were a constant source of encouragement and stood by me as pillar of strength for completing this work and course successfully.

Premprakash Kashyap

ABSTRACT

Online social networks (OSNs) have experienced tremendous growth in recent years and become a *de facto* portal for hundreds of millions of Internet users. These OSNs offer attractive means for digital social interactions and information sharing, but also raise a number of security and privacy issues. While OSNs allow users to restrict access to shared data, they currently do not provide any mechanism to enforce privacy concerns over data associated with multiple users.

To this end, they proposed an approach to enable the protection of shared data associated with multiple users in OSNs. An access control model is formulated to capture the essence of multiparty authorization requirements, along with a multiparty policy specification scheme and a policy enforcement mechanism.

Besides, a logical representation of access control model is presented which allows us to leverage the features of existing logic solvers to perform various analysis tasks on this model. Also a proof-of-concept prototype of this approach is discussed as part of an application in Facebook and provide usability study and system evaluation of specified method.

A systematic solution to facilitate collaborative management of shared data in OSNs is proposed in this work. This is done by examining how the lack of multiparty access control for data sharing in OSNs can undermine the protection of user data.

Some typical data sharing patterns with respect to multiparty authorization in OSNs are also identified. Based on these sharing patterns, a multiparty access control (MPAC) model is formulated to capture the core features of multiparty authorization requirements which have not been accommodated so far by existing access control systems and models for OSNs.

This model also contains a multiparty policy specification scheme. Meanwhile, since conflicts are inevitable in multiparty authorization enforcement, a voting mechanism is further provided to deal with authorization and privacy conflicts in their model.

CONTENTS

CHAPTERS **PAGE NO**

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

ONLINE social networks (OSNs) such as Facebook ,Google+, and Twitter are inherently designed to enable people to share personal and public information and make social connections with friends, co-workers , colleagues ,family and even with strangers. In recent years, we have seen unprecedented growth in the application of OSNs. For example, Facebook, one of representative social network sites, claims that it has more than 800 million active users and over 30 billion pieces of content (web links ,news stories, blog posts, notes, photo albums, etc.) shared each month. To protect user data, access control has become a central feature of OSNs[3],[4].

A typical OSN provides each user with a virtual space containing profile information, a list of the user's friends, and web pages, such as *wall* in Facebook, where users and friends can post content and leave messages. A user profile usually includes information with respect to the user's birthday, gender, interests, education and work history, and contact information. In addition, users can not only upload a content into their own or others' spaces but also to other users who appear in the content.

Each tag is an explicit reference that links to a user's space. For the protection of user data, current OSNs indirectly require users to be system and policy administrators for regulating their data, where users can restrict data sharing to a specific set of trusted users. OSNs often use *user relationship* and *group membership* to distinguish between trusted and untrusted users. For example, in Facebook, users can allow *friends*, *friends of friends*, *groups* or *public* to access their data, depending on their personal authorization and privacy requirements.

Although OSNs currently provide simple access control mechanisms allowing users to govern access to information contained in their own spaces, users, unfortunately, have no control over data residing *outside* their spaces. For instance, if a user posts a comment in a friend's space, s/he cannot specify which users can view the comment.

In another case, when a user uploads a photo and tags friends who appear in the photo, the tagged friends cannot restrict who can see this photo, even though the tagged friends may have different privacy concerns about the photo. To address such a critical issue, preliminary protection mechanisms have been offered by existing OSNs. For example, Facebook allows tagged users to remove the tags linked to their profiles or report violations asking Facebook managers to remove the contents that they do not want to share with the public. However, these simple protection mechanisms suffer from several limitations.

On one hand, removing a tag from a photo can only prevent other members from seeing a user's profile by means of the association link, but the user's image is still contained in the photo. Since original access control policies cannot be changed, the user's image continues to be revealed to all authorized users. On the other hand, reporting to OSNs only allows us to either keep or delete the content. Such a binary decision from OSN managers is either too loose or too restrictive, relying on the OSN's administration and requiring several people to report their request on the same content. Hence, it is essential to develop an effective and flexible access control mechanism for OSNs, accommodating the special authorization requirements coming from multiple associated users for managing the shared data collaboratively.

In this work, a systematic solution to facilitate collaborative management of shared data in OSNs is proposed. This is done by examining how the lack of multiparty access control for data sharing in OSNs can undermine the protection of user data. Some typical data sharing patterns with respect to multiparty authorization in OSNs are also identified.

Based on these sharing patterns, a multiparty access control (MPAC) model is formulated to capture the core features of multiparty authorization requirements which have not been accommodated so far by existing access control systems and models for OSNs. This model also contains a multiparty policy specification scheme. Meanwhile, since conflicts are inevitable in multiparty authorization enforcement, a voting mechanism is further provided to deal with authorization and privacy conflicts in their model.

Another compelling feature of the solution is the support of analysis on multiparty access control model and systems. The correctness of implementation of an access control model is based on the premise that the access control model is valid. Moreover, while the use of multiparty access control mechanism can greatly enhance the flexibility for regulating data sharing in OSNs, it may potentially reduce the certainty of system authorization consequences due to the reason that authorization and privacy conflicts need to be resolved elegantly.

Assessing the implications of access control mechanisms traditionally relies on the security analysis technique, which has been applied in several domains(e.g., operating systems, trust management, and role-based access control). In this approach, a method to represent and reason about the model in a logic program is introduced. In addition to this a prototype implementation of the authorization mechanism in the context of Facebook is also provided. The experimental results demonstrate the feasibility and usability of the approach.

*****

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

2.1 WHAT IS SOCIAL NETWORKS

The term social network was originally coined to distinguish between networks that were used for business purposes and those that were used for socializing. The term has been extended to include interactive websites with message boards, chat rooms or the ability to leave comments and have a discussion with other people. Examples of websites that are primarily used for socializing include Facebook and MySpace.

Many companies have their own Facebook pages and use them to promote products. Consumers get the feeling they are actually interacting with these companies, which is of course an illusion.

The websites are supposed to be fun; that was the original purpose. But many concerns have arisen as they have become more popular. There have been instances of bullying. A young gay man was outed on Facebook and later committed suicide. A woman used the site to harass a female teenager who also committed suicide.

There are also privacy concerns. Your information can be viewed not only by your friends, but also by companies, spammers, thieves and perhaps your potential boss. There are ways to reduce your risks and still use the social networking sites but you have to be careful.

The good news is that there are hundreds of social networking and informational sites created for the softer side of humanity. Networking sites where peace and love are promoted; helping the jobless find work, activism in food industry, alchemy, ascension, and metaphysics.

While there are risks inherent in online social networking, there are also many potential benefits. Social networking can provide opportunities for new relationships as well as strengthening existing relationships, whether your kids' friends are close to home or across the world. It's important to be vigilant when your kids are getting involved in online social networking, but it's also good to encourage positive relationships through various avenues, including the Internet.

Social networking platforms may allow organizations to improve communication and productivity by disseminating information among different groups of employees in a more efficient manner, resulting in increased productivity. While it is not meant to be all-inclusive, the list below outlines some of the possible advantages and disadvantages.

- **Possible advantages:**

- Facilitates open communication, leading to enhanced information discovery and delivery.
- Allows employees to discuss ideas, post news, ask questions and share links.
- Provides an opportunity to widen business contacts.
- Targets a wide audience, making it a useful and effective recruitment tool.
- Improves business reputation and client base with minimal use of advertising.
- Expands market research, implements marketing campaigns, delivers communications and directs interested people to specific web sites.

- **Possible disadvantages:**

- Opens up the possibility for hackers to commit fraud and launch spam and virus attacks.
- Increases the risk of people falling prey to online scams that seem genuine, resulting in data or identity theft.
- May result in negative comments from employees about the company or potential legal consequences if employees use these sites to view objectionable, illicit or offensive material.
- Potentially results in lost productivity, especially if employees are busy updating profiles, etc.

Employers do have the right to simply ban all computer activity that is not work-related, but this approach may not yield optimal results. If employees are to be allowed access to social networking platforms, then a comprehensive and well-defined policy should be established to prevent abuse.

What may be the most concerning aspect of social networking platforms is that they encourage people to share personal information. Even the most cautious and well-meaning individuals can give away information they should not; the same applies to what is posted on company-approved social networking platforms.

Employees may not be aware of how our actions online may compromise company security. Educate employees as to how a simple click on a received link or a downloaded application can result in a virus infecting our computer and the network. Advise them not to click on suspicious links and to pay careful attention when providing personal information online. Remember that just because employees may have an online profile, it doesn't necessarily mean they have a high level of security awareness.

Present OSNs

Online social networks (OSNs) have experienced tremendous growth in recent years and become a *de facto* portal for hundreds of millions of Internet users. These OSNs offer attractive means for digital social interactions and information sharing. but also raise a number of security and privacy issues. Online social networks (OSNs) such as Facebook, Google+, and Twitter are inherently designed to enable people to share personal and public information and make social connections with friends, coworkers, colleagues, family and even with strangers.

Disadvantages of existing system

While OSNs allow users to restrict access to shared data, they currently do not provide any mechanism to enforce privacy concerns over data associated with multiple users.

Although OSNs currently provide simple access control mechanisms allowing users to govern access to information contained in their own spaces, users, unfortunately, have no control over data residing *outside* their spaces. For instance, if a user posts a comment in a friend's space, s/he cannot specify which users can view the comment.

In another case, when a user uploads a photo and tags friends who appear in the photo, the tagged friends cannot restrict who can see this photo, even though the tagged friends may have different privacy concerns about the photo. To address such a critical issue, preliminary protection mechanisms have been offered by existing OSNs. For example, Facebook allows tagged users to remove the tags linked to their profiles or report violations asking Facebook managers to remove the contents that they do not want to share with the public.

Advantatages of proposed system over existing system

However, above mentioned simple protection mechanisms suffer from several limitations. On one hand, removing a tag from a photo can only prevent other members from seeing a user's profile by means of the association link, but the user's image is still contained in the photo. Since original access control policies cannot be changed, the user's imag continues to be revealed to all authorized users. On the other hand, reporting to OSNs only allows us to either

keep or delete the content. Such a binary decision from OSN managers is either too loose or too restrictive, relying on the OSN's administration and requiring several people to report their request on the same content. Hence, it is essential to develop an effective and flexible access control mechanism for OSNs, accommodating the special authorization requirements coming from multiple associated users for managing the shared data collaboratively.

So a multiparty access control for online social networks:model and mechanisms is proposed for solving the problems related to photo sharing of the existing system.It involves certain models which define their own policies for the proper functioning of the required problems.

2.2 System Usability and Performance Evaluation

2.2.1 Participants and Procedure

MController is a functional proof-of-concept implementation of collaborative privacy management. To measure the practicality and usability of our mechanism, we conducted a survey study (n=35) to explore the factors surrounding users' desires for privacy and discover how we might improve those implemented in *MController*.

Specifically, we were interested in users' perspectives on the current Facebook privacy system and their desires for more control over photos they do not own. We recruited participants through university mailing lists and through Facebook itself using Facebook's built-in sharing API. Users were given the opportunity to share our application and play with their friends. While this is not a random sampling, recruiting using the natural dissemination features of Facebook arguably gives an accurate profile of the ecosystem.

Participants were first asked to answer some questions about their usage and perception of Facebook's privacy controls, then were invited to watch a video (http://bit.ly/MController) describing the concept behind *MController*. Users were then instructed to install the application using their Facebook profiles and complete the following actions: set privacy settings for a photo

they do not own but are tagged in, set privacy settings for a photo they own, set privacy settings for a photo they contributed, and set privacy settings for a photo they disseminated. As users completed these actions, they answered questions on the usability of the controls in *MController*. Afterward, they were asked to answer further questions and compare their experience with *MController* to that in Facebook.

2.2.2 User Study of MController

For evaluation purposes, questions were split into three areas: *likeability*, *simplicity*, and *control*. *Likeability* is a measure of a user's satisfaction with a system (e.g. "I like the idea of being able to control photos in which I am tagged"). *Simplicity* is a measure how intuitive and useful the system is (e.g. "Setting my privacy settings for a photo in *MController* is Complicated (1) to Simple (5)" with a 5-point scale).

Control is a measure of the user's perceived control of their personal data (e.g. "If Facebook implemented controls like *MController*'s to control photo privacy, my photos would be better protected"). Questions were either True/False or measured on a 5-pointlikert scale, and all responses were scaled from 0 to 1 for numerical analysis. In the measurement, a higher number indicates a positive perception or opinion of the system while a lower number indicates a negative one.

To analyze the average user perception of the system, we used a 95% confidence interval for the users' answers. This assumes the population to be mostly normal. *Before Using MController*. Prior to using *MController*, users were asked a few questions about their usage of Facebook to determine the user's perceived usability of the current Facebook privacy controls.

Since we were interested in the maximum average perception of Facebook, we looked at the upper bound of the confidence interval. An average user asserts at most 25% positively about the *likability* and *control* of Facebook's privacy management mechanism, and at most 44% on Facebook's *simplicity* as shown in Table 1.

This demonstrates an average negative opinion of the Facebook's privacy controls that users currently must use. *After Using MController*. Users were then asked to perform a few tasks in *MController*. Since we were interested in the average minimum opinion of *MController*, we

looked at the lower bound of the confidence interval. An average user asserts at least 80% positively about the *likability* and *control*, and at least 67% positively on *MController*'s *simplicity* as shown in Table 1. This demonstrates an average positive opinion of the controls and ideas presented to users in *MController*. *Improvement*. Besides viewing user opinions and analyzing the usability of our proof-of-concept application, we also briefly investigated the potential improvement of our application.

The lowest score for *MController* is in the area of *simplicity*. On average, users found setting privacy settings for a photo more complicated in *MController* than Facebook. This means that while users appreciated the privacy controls of *MController* presented to them, it would be desirable to further simplify the process implemented in *MController*. This could be achieved by adopting learnbased generation of privacy settings for OSNs.

2.2.3 Performance Evaluation

To evaluate the performance of the policy evaluation mechanism in *MController*, we changed the number of the controllers of a shared photo from 1 to 20, and assigned each controller with the average number of friends, 130, which is claimed by Facebook statistics. Also, we considered two cases for our evaluation. In the first case, each controller allows "friends" to access the shared photo. In the second case, controllers specify "friends of friends" as the accessors instead of "friends". In our experiments, we performed 1,000 independent trials and measured the performance of each trial.

Since the system performance depends on other processes running at the time of measurement, we had initial discrepancies in our performance. To minimize such an impact, we performed 10 independent trials (a total of 10,000 calculations for each number of controllers).

For both cases, the experimental results showed that the policy evaluation time increases linearly with the increase of the number of controllers. With the simplest implementation of our mechanism, where n is the number of controllers of a shared photo, a series of operations essentially takes place n times. There are O(n) MySQL calls and data fetching operations and O(1) for additional operations. Moreover,we could observe there was no significant overhead when we run *MController* in Facebook.

Matric	**Facebook**		**Mcontroller**	
	Average	Upper bound on 95% confidence interval	Average	Lower bound on 95% confidence interval
Likability	0.20	0.25	0.83	0.80
Simplicity	0.38	0.44	0.72	0.64
Control	0.20	0.25	0.83	0.80

Table 2.1 Usability Evaluation for Facebook and *MController* Privacy Controls.

*****

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 SYSTEM CONFIGURATION:

3.1.1 HARDWARE REQUIREMENTS:

Hardware	-	Pentium4
Speed	-	1.1 GHz
RAM	-	1GB
Hard Disk	-	20 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

3.1.2 SOFTWARE REQUIREMENTS:

Operating System	: Windows
Technology	: Java and J2EE
Web Technologies	: Html, JavaScript, CSS
IDE	: My Eclipse
Web Server	: Tomcat
Tool kit	: Android Phone
Database	: My SQL
Java Version	: J2SDK1.5

*****

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

The existing work could model and analyze access control requirements with respect to collaborative authorization management of shared data in OSNs. The need of joint management for data sharing, especially photo sharing, in OSNs has been recognized by the recent work provided a solution for collective privacy management in OSNs. Their work considered access control policies of a content that is co-owned by multiple users in an OSN, such that each co-owner may separately specify her/his own privacy preference for the shared content.

4.2 PROPOSED SYSTEM

In Proposed System they implemented a proof-of-concept Facebook application for the collaborative management of shared data, called *MController*. Their prototype application enables multiple associated users to specify their authorization policies and privacy preferences to co-control a shared data item. It is worth noting that their current implementation was restricted to handle photo sharing in OSNs. Obversely, their approach can be generalized to deal with other kinds of data sharing and comments, in OSNs as long as the stakeholder of shared data are identified with effective methods like tagging or searching. The proposed system shows a novel solution for collaborative management of shared data in OSNs.

A multiparty access control model was formulated, along with a multiparty policy specification scheme and corresponding policy evaluation mechanism. In addition, they have introduced an approach for representing and reasoning about their proposed model. A proof-of-concept implementation of their solution called *MController* has been discussed as well,

followed by the usability study and system evaluation of their method. Indeed, a flexible access control mechanism in a multi-user environment like OSNs should allow multiple controllers, who are associated with the shared data, to specify access control policies. As they identified previously in the sharing patterns in addition to the *owner* of data, other controllers, including the *contributor*, *stakeholder* and *disseminator* of data, need to regulate the access of the shared data as well. In their multiparty access control system, a group of users could collude with one another so as to manipulate the final access control decision.

4.3 SYSTEM STUDY

4.3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

A. ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

B. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

C. SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

*****

CHAPTER 5

SYSTEM DESIGN

5.1 MULTIPARTY ACCESS CONTROL FOR PROFILE AND RELATIONSHIP SHARING

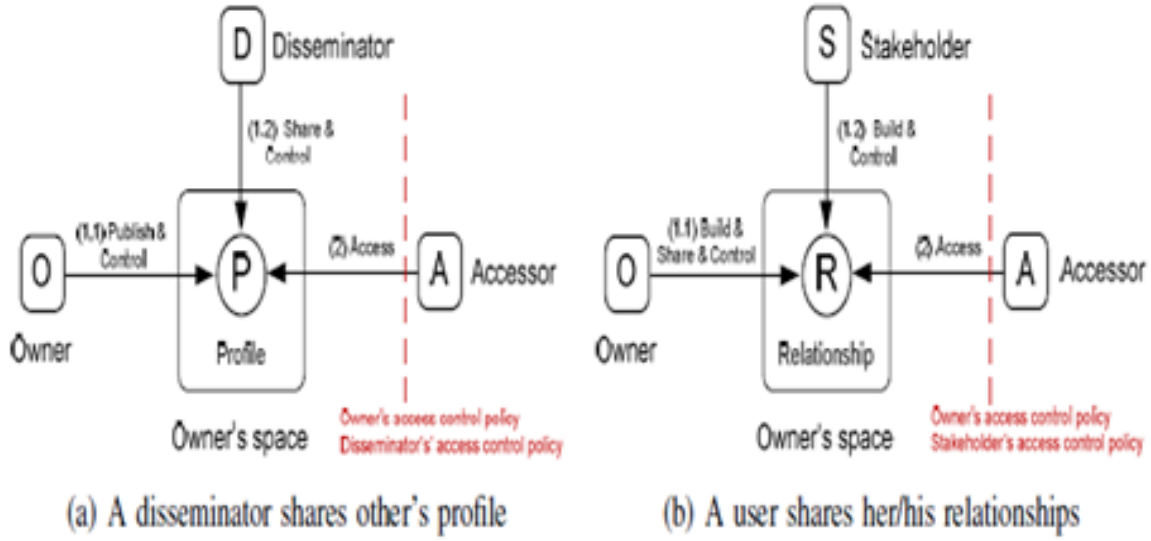
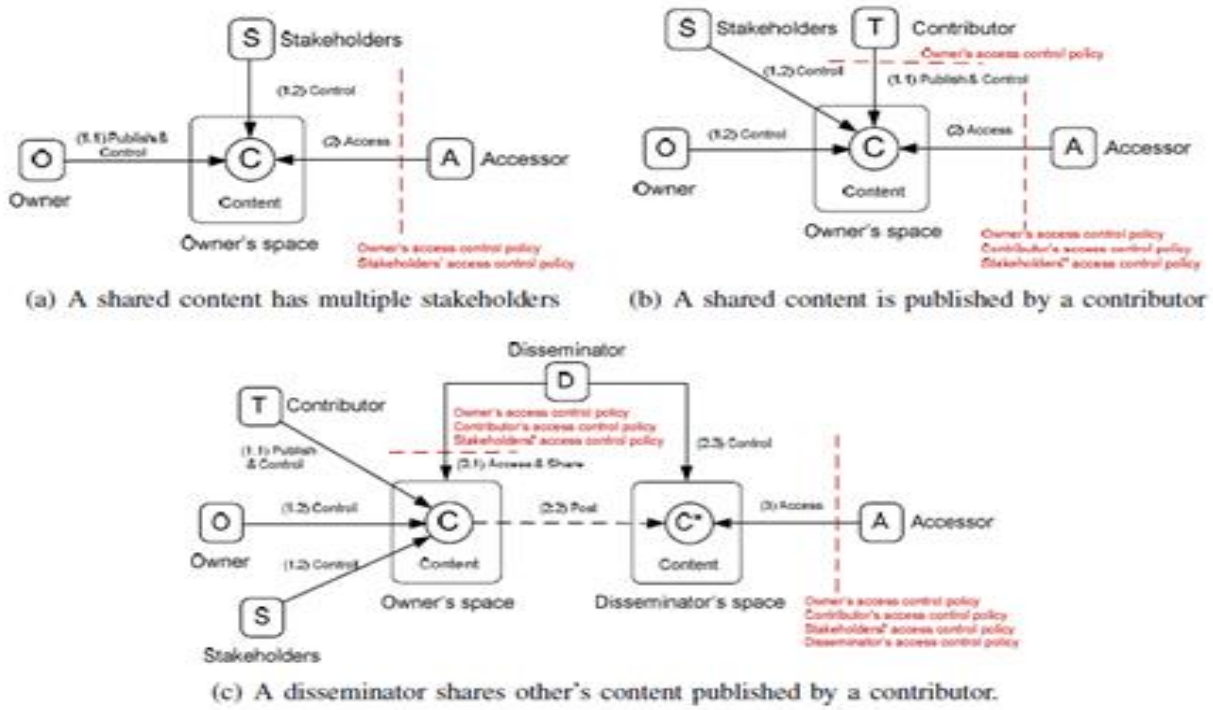


Fig 5.1: Multiparty Access Control For Profile and Relationship Sharing

5.2 MULTIPARTY ACCESS CONTROL FOR CONTENT SHARING



(a) A shared content has multiple stakeholders (b) A shared content is published by a contributor

(c) A disseminator shares other's content published by a contributor.

Fig 5.2: Multiparty Access Control for Content Sharing

5.3 MULTIPARTY SOCIAL NETWORK REPRESENTATION

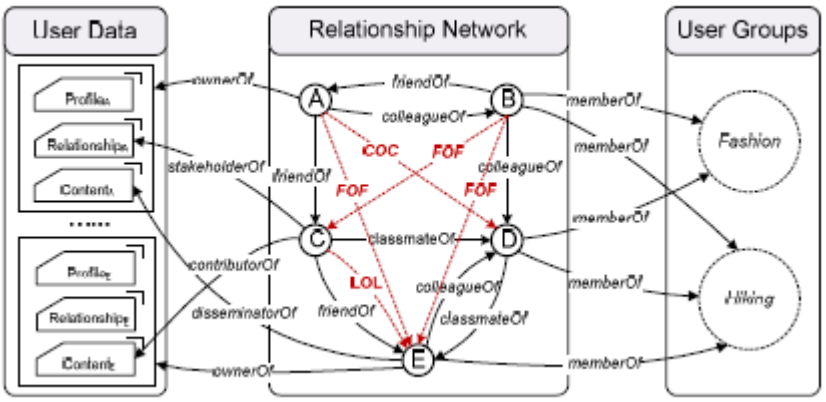


Fig 5.3: An Example of Multiparty Social Network Representation

5.4 MULTIPARY POLICY EVALUATION PROCESS

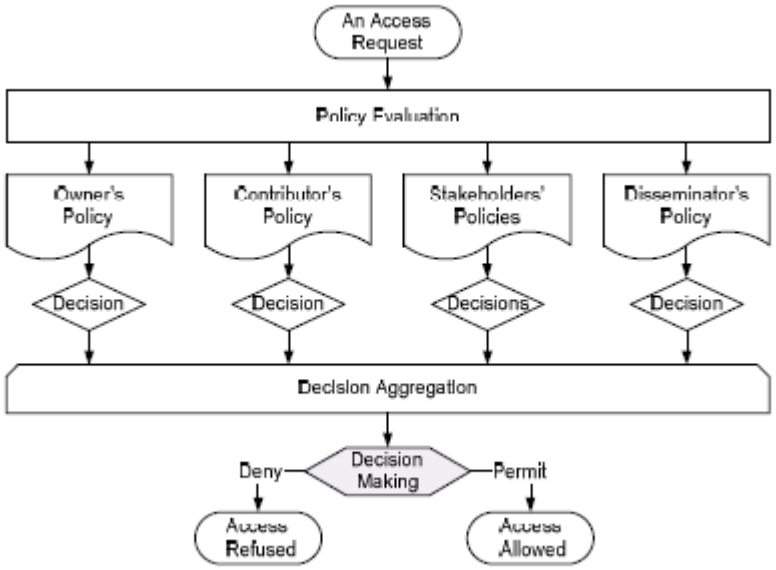


Fig 5.4: Multiparty Policy Evaluation Process

5.5 OVERALL ARCHITECTURE OF MCONTROLLER APPLICATION

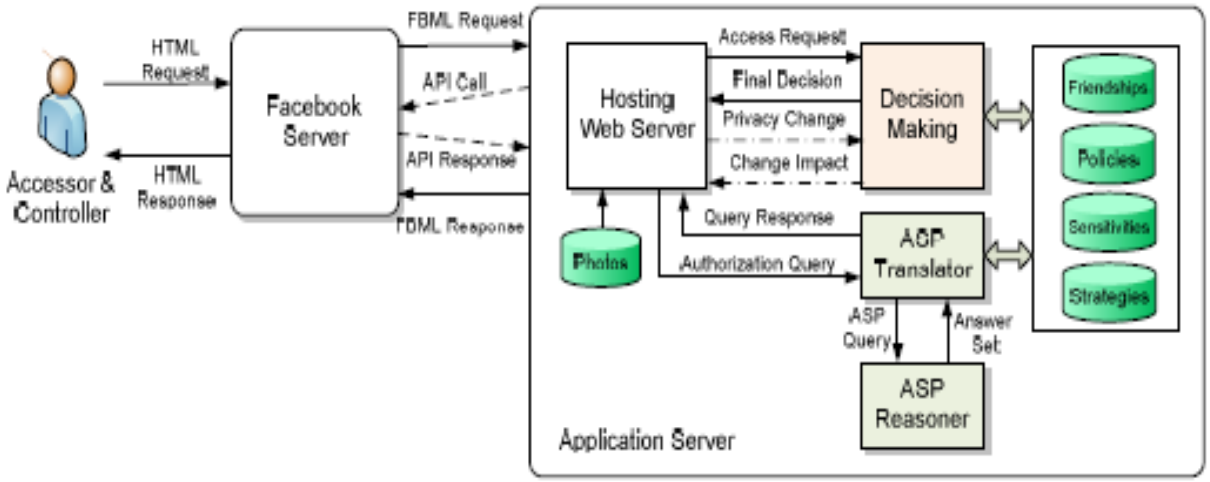


Fig 5.5: Overall Architecture of MController Application

5.6 SYSTEM ARCHITECTURE OF DECISION MAKING IN MCONTROLLER

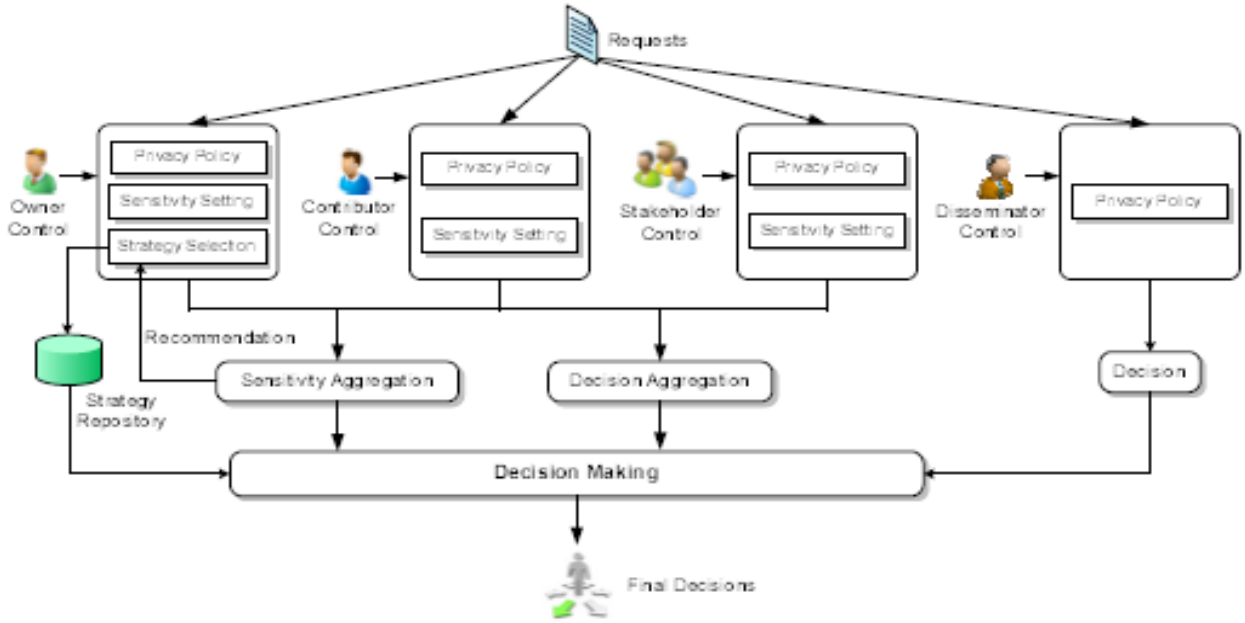


Fig 5.6: System Architecture of Decision Making in MController

5.7 DATA FLOW DIAGRAM / USE CASE DIAGRAM/ FLOW DIAGRAM

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

5.7.1 DATA FLOW DIAGRAM

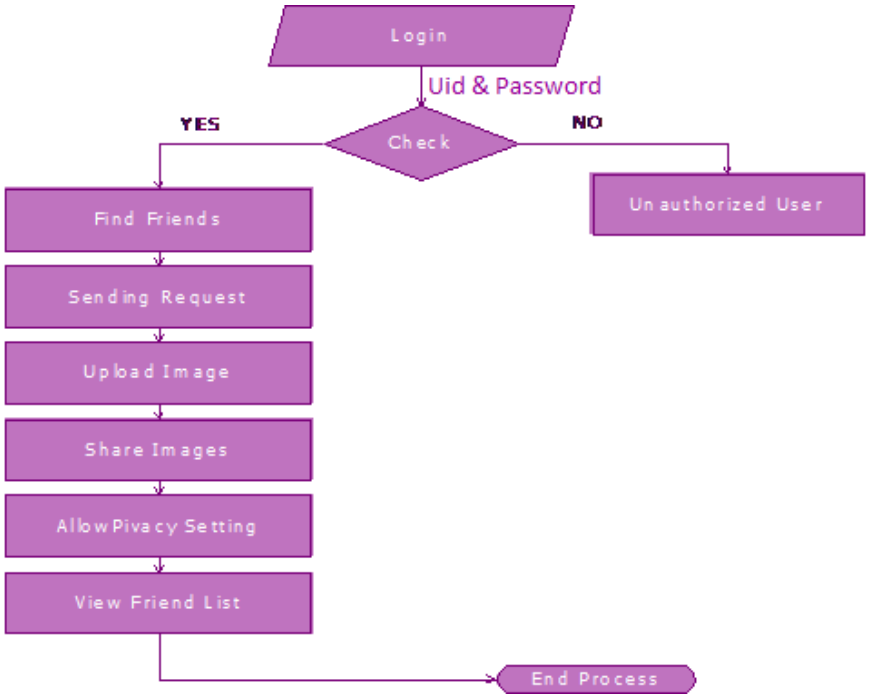


Fig 5.7: Data Flow Diagram

5.7.2 USE CASE DIAGRAM

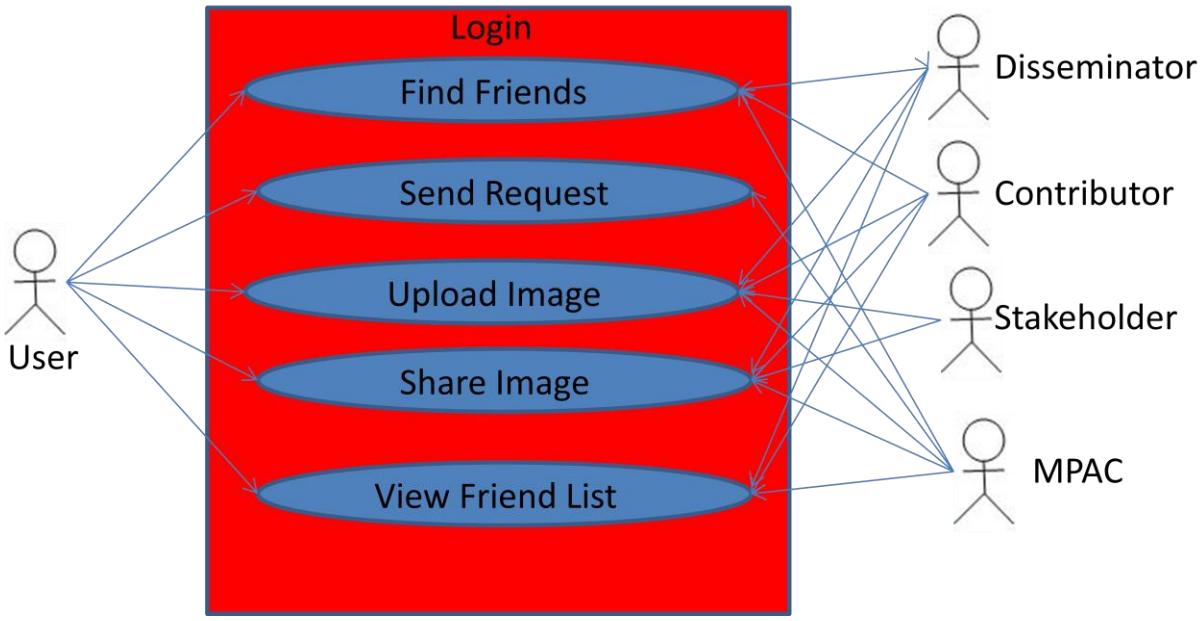


Fig 5.8: Use Case Diagram

5.7.3 ACTIVITY DIAGRAM

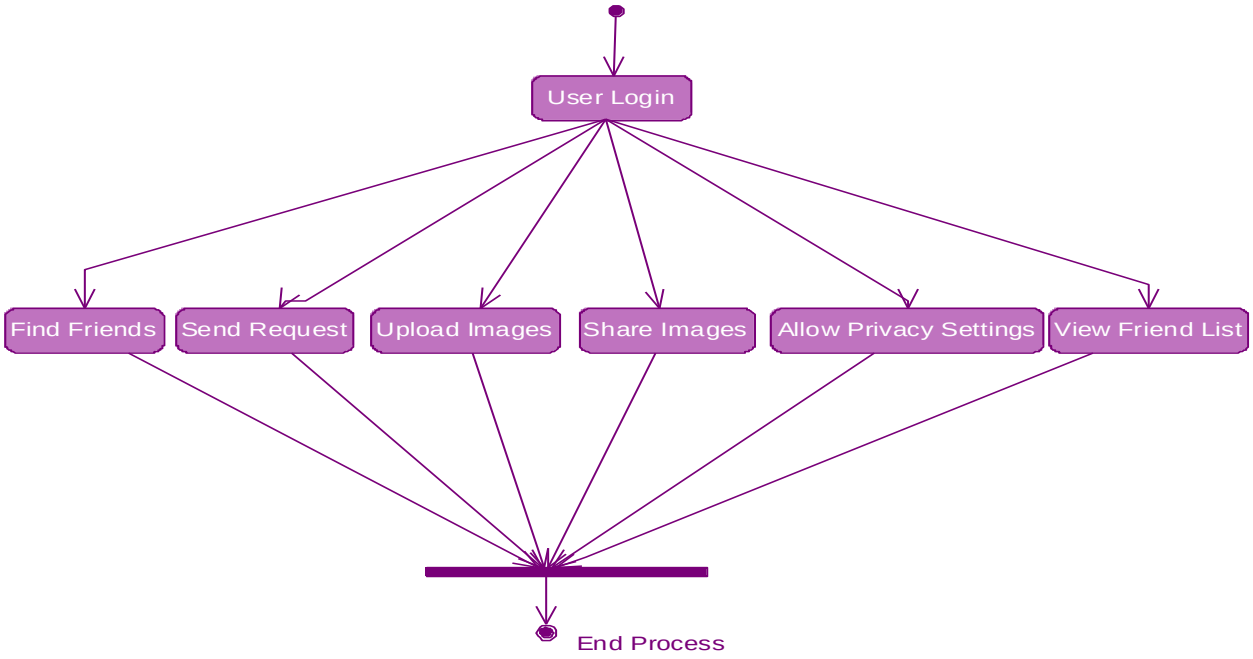


Fig 5.9: Activity Diagram

5.7.4 SEQUENCE DIAGRAM

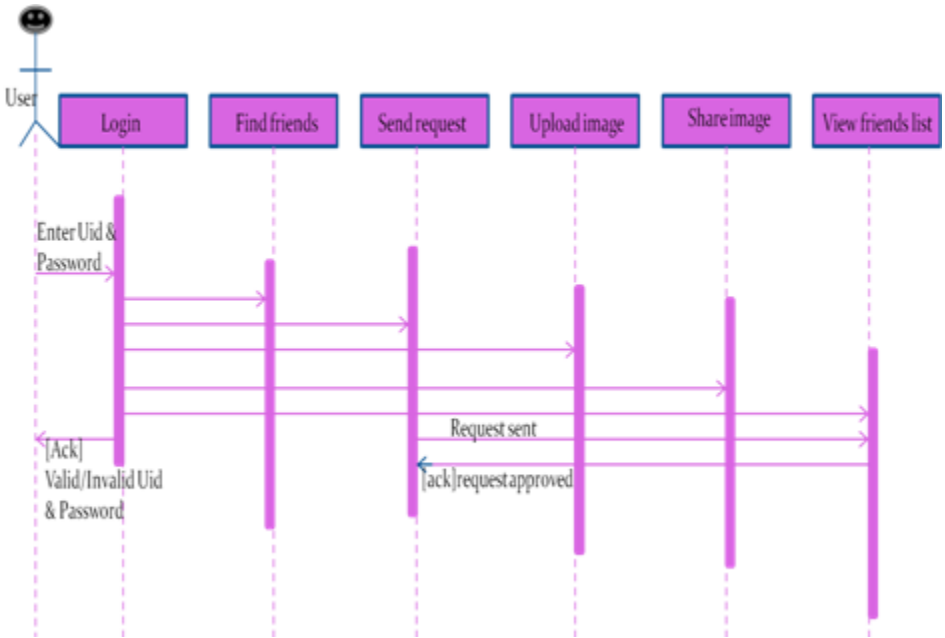


Fig 5.10: Sequence Diagram

CHAPTER 6

SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

6.1 MODULE DESCRIPTION

After careful analysis the system has been identified to have the following modules:

1. **OWNER MODULE**
2. **CONTRIBUTOR MODULE**
3. **STAKEHOLDER MODULE**
4. **DISSEMINATOR MODULE**
5. **MPAC MODULE**

6.1.1 OWNER MODULE

In Owner module let *d* be a data item in the space m of a user *u* in the social network. The user *u* is called the owner of *d*. The user *u* is called the contributor of *d*. We specifically analyze three scenarios—profile sharing, relationship sharing and content sharing—to understand the risks posted by the lack of collaborative control in OSNs. In this the owner and the disseminator can specify access control policies to restrict the sharing of profile attributes.

Thus, it enables the owner to discover potential malicious activities in collaborative control. The detection of collusion behaviors in collaborative systems has been addressed by the recent work.

6.1.2 CONTRIBUTOR MODULE

In Contributor module let d be a data item published by a user u in someone else's space in the social network. The contributor publishes content to other's space and the content may also have multiple stakeholders (e.g., tagged users). The memory space for the user will be allotted according to user request for content sharing. A shared content is published by a contributor.

6.1.3 STAKEHOLDER MODULE

In Stakeholder module let *d* be a data item in the space of a user in the social network. Let *T* be the set of tagged users associated with *d*. A user *u* is called a stakeholder of *d*, if *u 2 T* who has a relationship with another user called *stakeholder*, shares the relationship with an *accessor*. In this scenario, authorization requirements from both the owner and the stakeholder should be considered. Otherwise, the stakeholder's privacy concern may be violated. A shared content has multiple stakeholders.

6.1.4 DISSEMINATOR MODULE

In Disseminator module let *d* be a data item shared by a user *u* from someone else's space to his/her space in the social network. The user *u* is called a disseminator of *d*. A content sharing pattern where the sharing starts with an *originator* (*owner* or *contributor* who uploads the content) publishing the content, and then a disseminator views and shares the content. All access control policies defined by associated users should be enforced to regulate access of the content in disseminator's space. For a more complicated case, the disseminated content may be further *re-disseminated* by disseminator's friends, where effective access control mechanisms should be applied in each procedure to regulate *sharing* behaviors. Especially, regardless of how many

steps the content has been re-disseminated, the original access control policies should be always enforced to protect further dissemination of the content.

6.1.5 MPAC MODULE

MPAC is used to prove if their proposed access control model is valid. To enable a collaborative authorization management of data sharing in OSNs, it is essential for multiparty access control policies to be in place to regulate access over shared data, representing authorization requirements from multiple associated users. Their policy specification scheme is built upon the proposed MPAC model. *Accessor Specification*: Accessors are a set of users who are granted to access the shared data. Accessors can be represented with a set of user names, asset of relationship names or a set of group names in OSNs.

6.2 SOFTWARE ENVIRONMENT

6.2.1 JAVA

Java is a high-level, third generation programming language, like C, Fortran, Smalltalk, Perl and many others. You can use Java to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of our things computer softwarecan do. Compared to the other programming languages, Java is most similar to C. However though Java shares much of C's syntax. It is not C. Knowing how to program in C or better yet, C++ will certainly help you to lead Java more quickly, but you don't need to know C to learn Java. Unlicke C++ Java is not a superset of C. A Java compiler won't compile C code and most large C programs need to be changed substantially before they can become Java programs.

6.2.2 JAVA TECHNOLOGY

Java technology is both a programming language and a platform.

6.2.2.1 The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

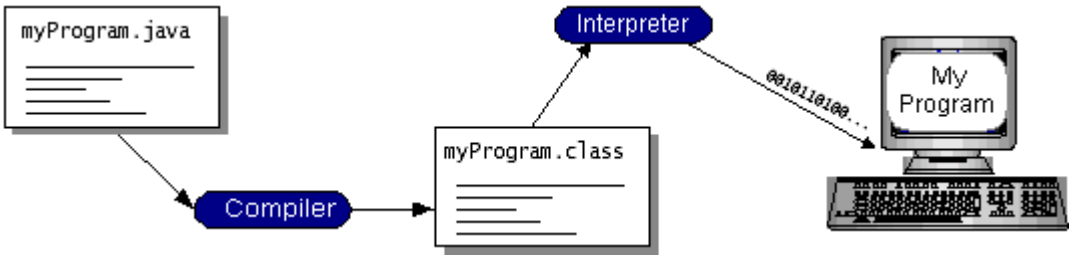


Fig6.1 Java byte code

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

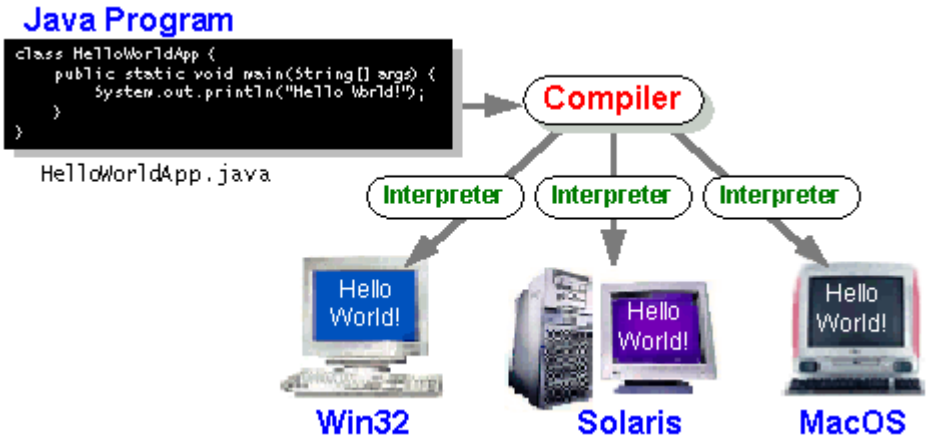


Fig 6.2: Execution of Java Programming Language

6.2.2.2 The Java Plateform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

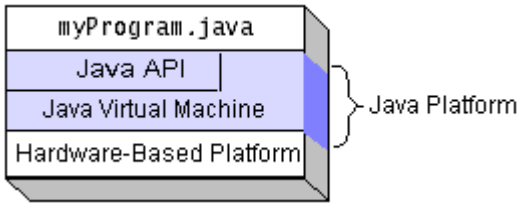


Fig6.3: Program Running on Java Platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

- **What Can Java Technology Do?**

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

- **Software components**: Known as JavaBeans™, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

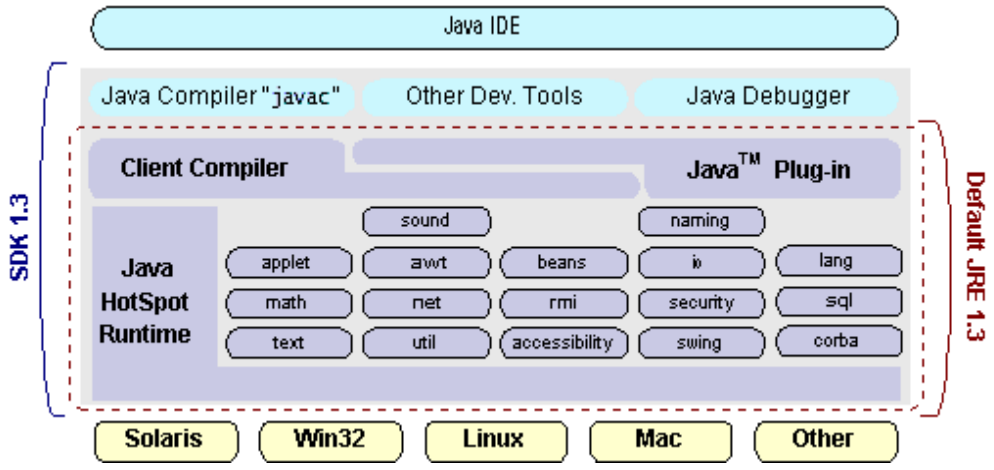


Fig6.4: Java 2 SDK

- **How Will Java Technology Change My Life?**

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

- **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.

- **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
- **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.
- **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
- **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

6.2.3 ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port our program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly

language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

6.2.4 JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

6.2.4.1 JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. **SQL Level API**

The designers felt that our main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to "generate" JDBC code and to hide many of JDBC's complexities from the end user.

2. **SQL Conformance**

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

3. **JDBC must be implemental on top of common database interfaces**

The JDBC SQL API must "sit" on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

4. **Provide a Java interface that is consistent with the rest of the Java system**

Because of Java's acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. **Keep it simple**

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

6. **Use strong, static typing wherever possible**

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

7. **Keep the common cases simple**

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking. And for dynamically updating the cache table we go for MS Access database. Java has two things: a programming language and a platform. Java is a high-level programming language that is all of the following:

Simple	Architecture-neutral
Object-oriented	Portable
Distributed	High-performance
Interpreted	multithreaded
Robust	Dynamic
Secure	

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

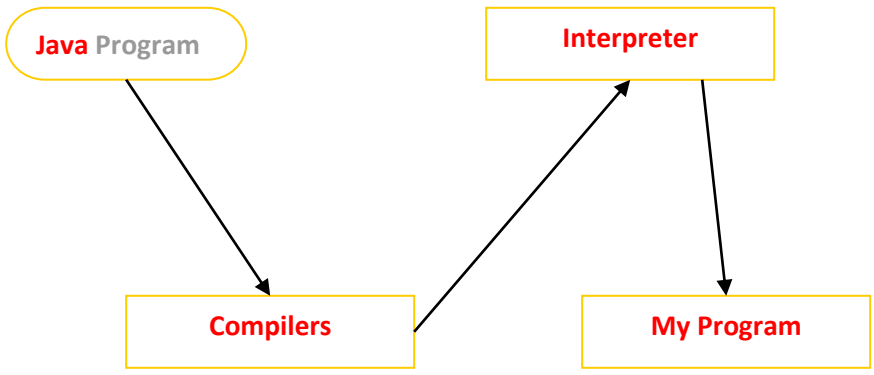


Fig6.5: Working of Java

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

6.2.5 NETWORKING TCP/IP STACK:

The TCP/IP stack is shorter than the OSI one:

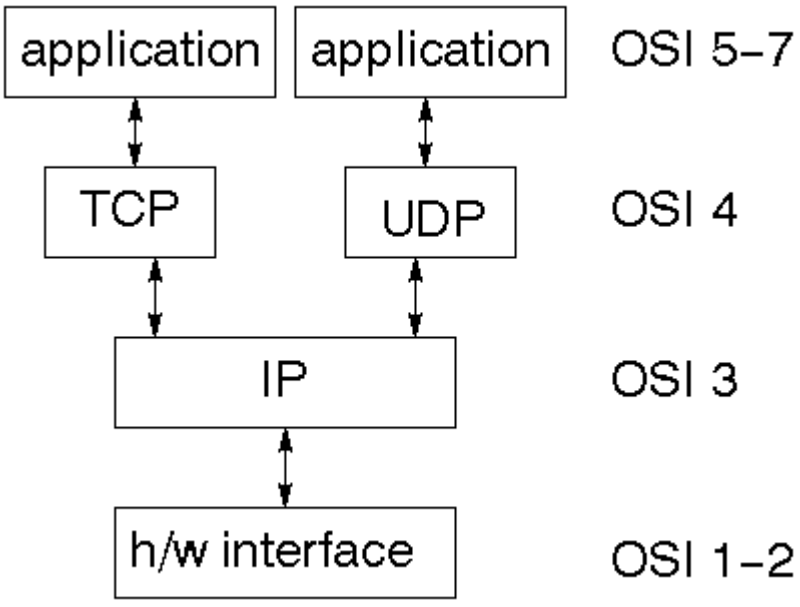


Fig6.6: Networking TCP/IP Stack

TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

- **IP datagram's:**

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

- **UDP:**

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

- **TCP:**

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

- **Internet addresses**

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

- **Network address:**

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

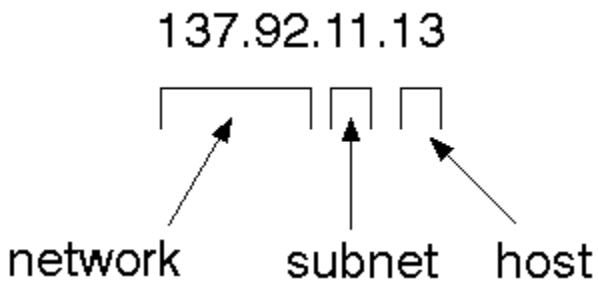
- **Subnet address:**

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

- **Host address:**

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

- **Total address:**



The 32 bit address is usually written as 4 integers separated by dots.

- **Port addresses**

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

- **Sockets:**

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(int family, int type, int protocol);
```

Here "family" will be AF_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a

network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

- **JFree Chart:**

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in our applications. JFreeChart's extensive feature set includes:

- A consistent and well-documented API, supporting a wide range of chart types;
- A flexible design that is easy to extend, and targets both server-side and client-side applications;
- Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
- JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

1. **Map Visualizations**

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

2. **Time Series Chart Interactivity**

Implement a new (to JFreeChart) feature for interactive time series charts - to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. **Dashboards**

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

4. **Property Editors**

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

- **Tomcat 6.0 web server**

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process.

Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs Weblogic, is one of the popular application server).To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc to run your application.

Fig6.7: Tomcat Web Server

*****

CHAPTER 7

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

7.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.4 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.7 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

*****

CHAPTER 8

INTERPRETATION OF RESULTS

8.1 SNAP SHOTS

8.1.1 SIGNUP PAGE

Fig 8.1 Signup page

The page is created for new user signup.

8.1.2 FIND FRIENDS

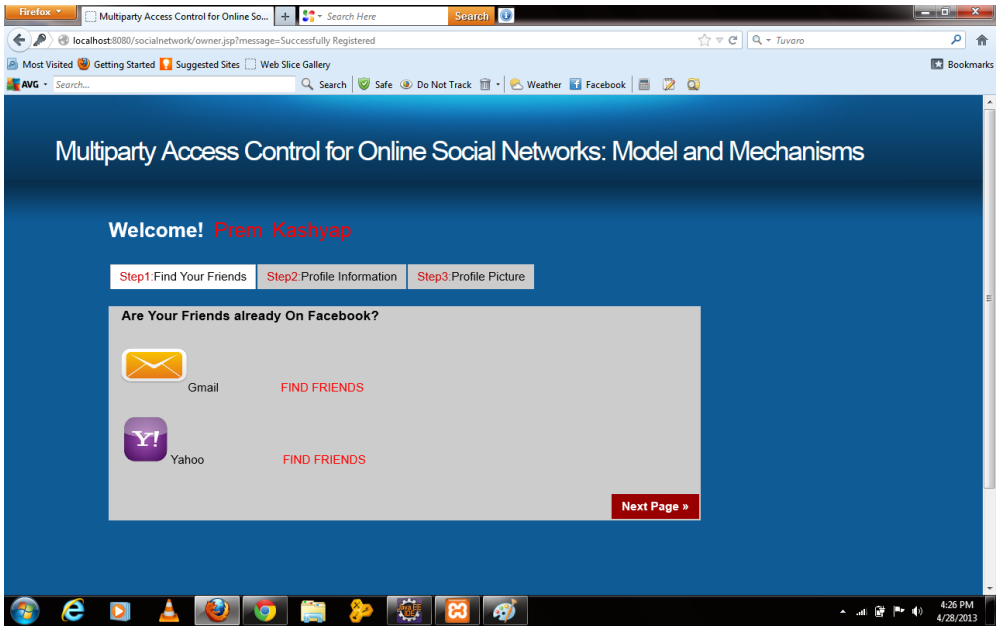


Fig 8.2 Find Friends

This page is created for finding your friends in OSNs.

8.1.3 PROFILE INFORMATION

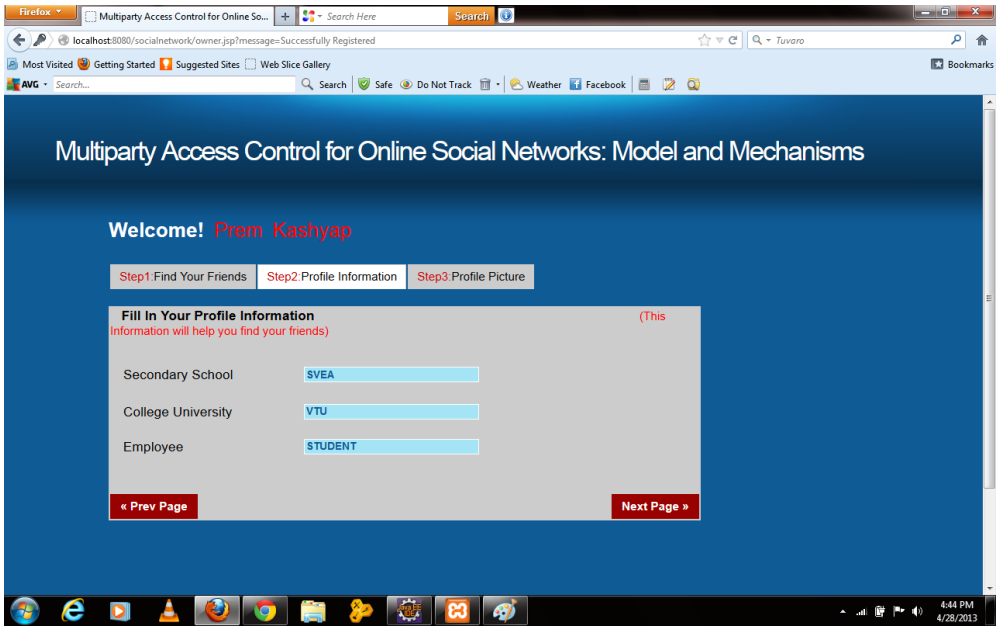


Fig 8.3 Profile Information

This page is created for filling profile information for the new user.

8.1.4 PROFILE PICTURE

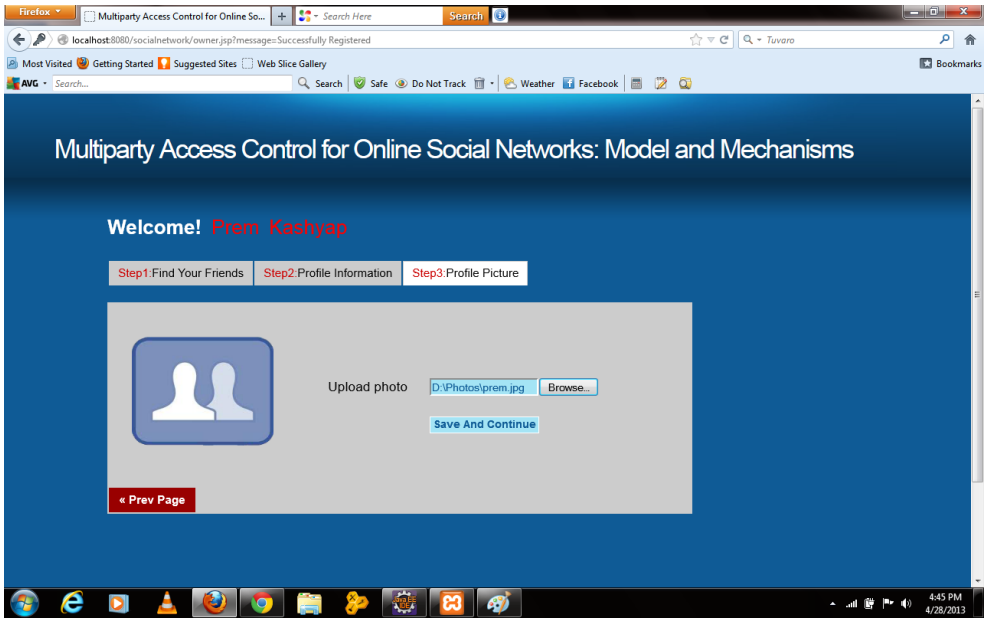


Fig 8.4 Profile Picture

This page is created for updating profile picture.

8.1.5 LOGIN

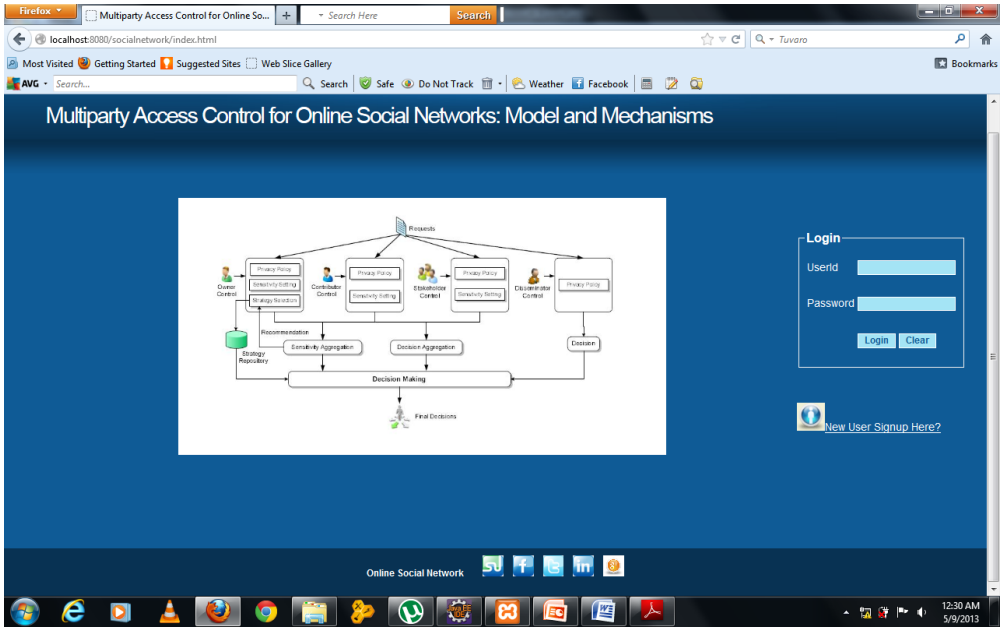


Fig 8.5 Logine Page

This page is created for User Login.

8.1.6 USER LOGIN PAGE

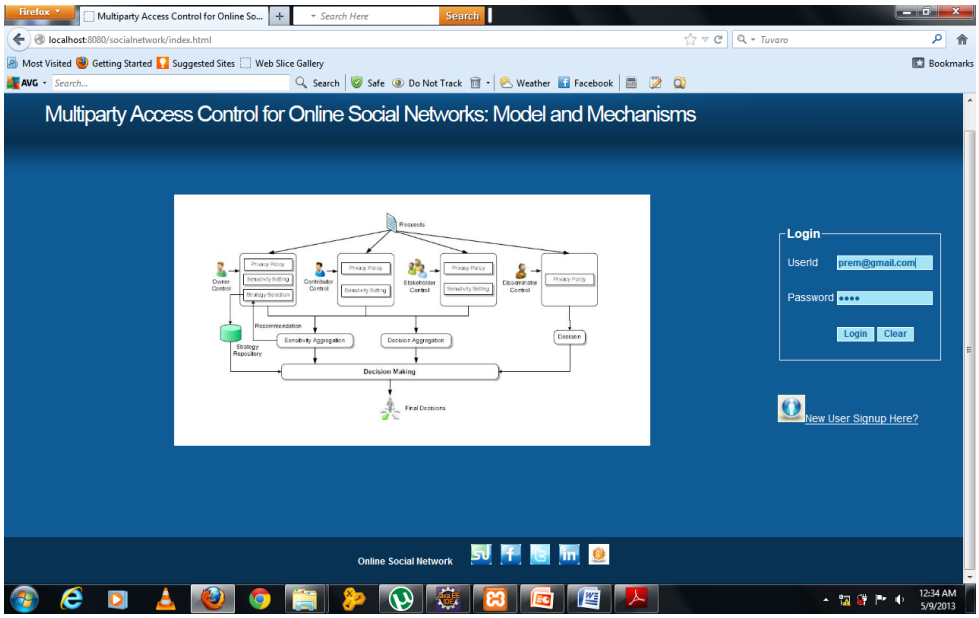


Fig 8.6 User Login Page

This page is created for user login, Here user will enter his Userid and Password.

8.1.7 OWNER'S PROFILE

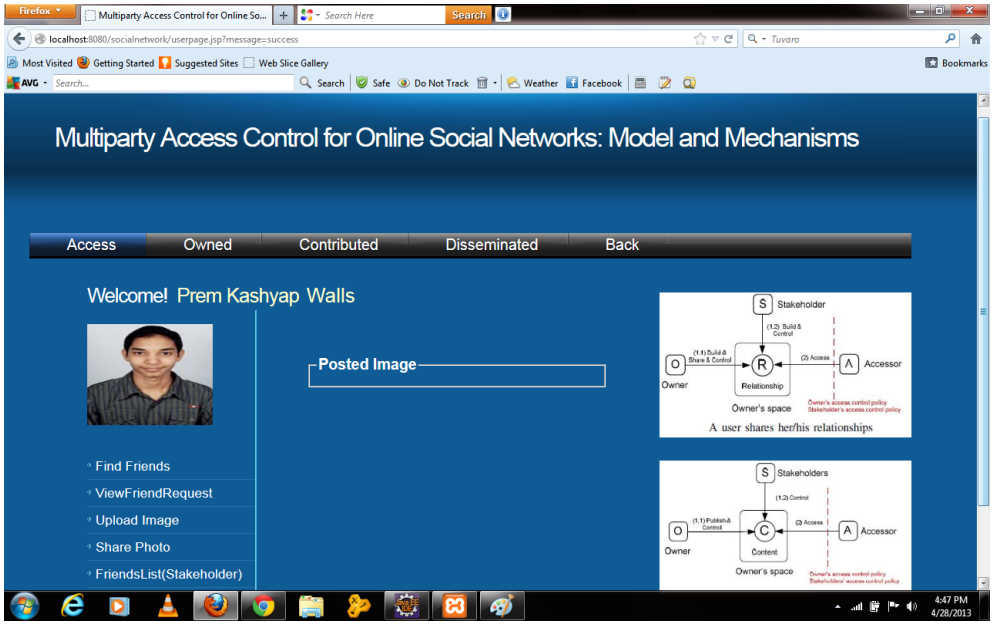


Fig 8.7 Owner's Profile

This page shows the user's profile welcome.

8.1.8 UPLOAD IMAGE

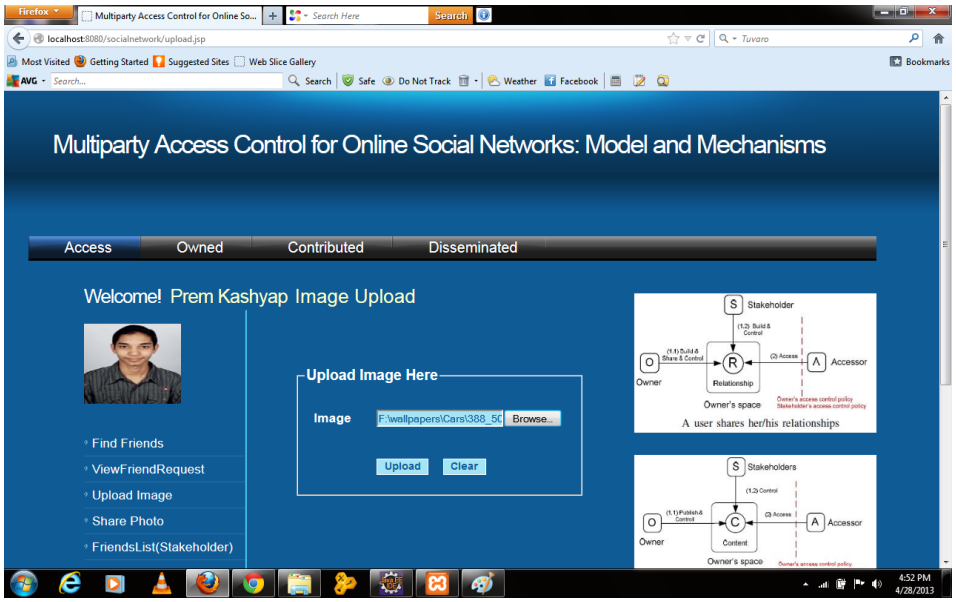


Fig 8.8 Upload Image

This page is created for uploading the image for user.

8.1.9 UPLOADING IMAGE

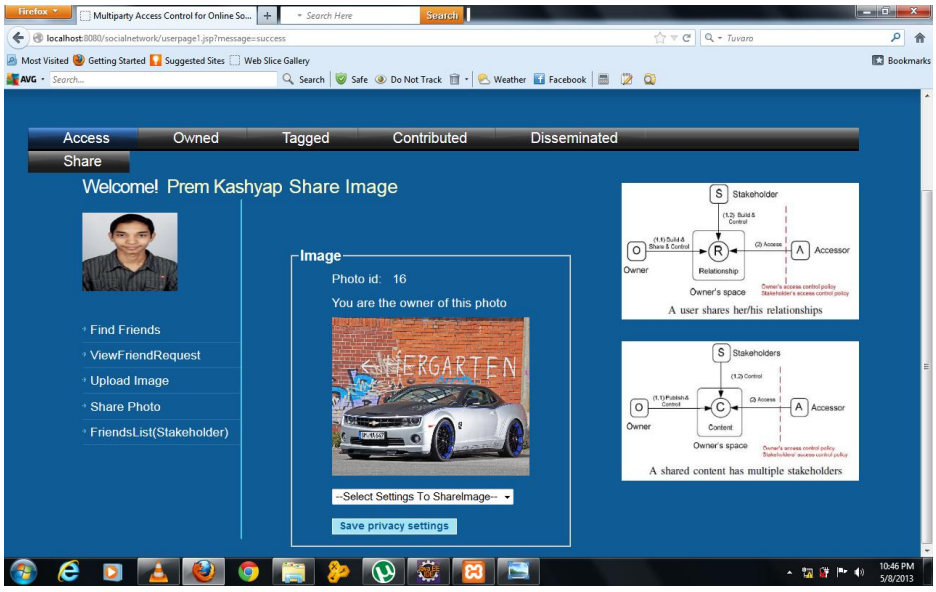


Fig 8.9 Uploading Image

This shows that user is uploading the image and he can set his own privacy for share image.

8.1.10 SAVE PRIVACY

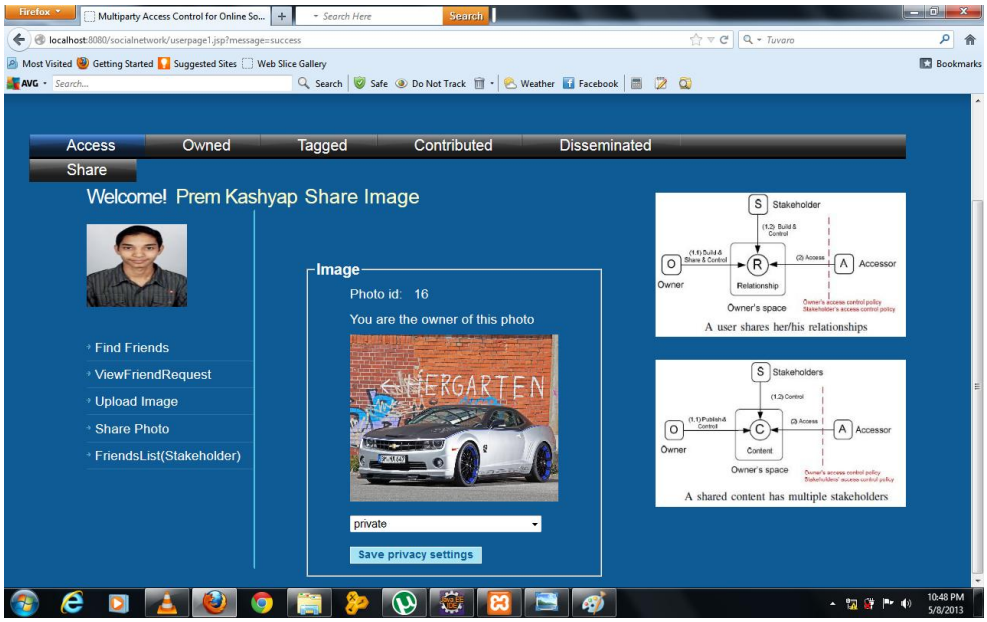


Fig 8.10 Save Privacy

This page shows the user is set his own privacy either public or private and save privacy setting.

8.1.11 PRIVACY SETTINGS

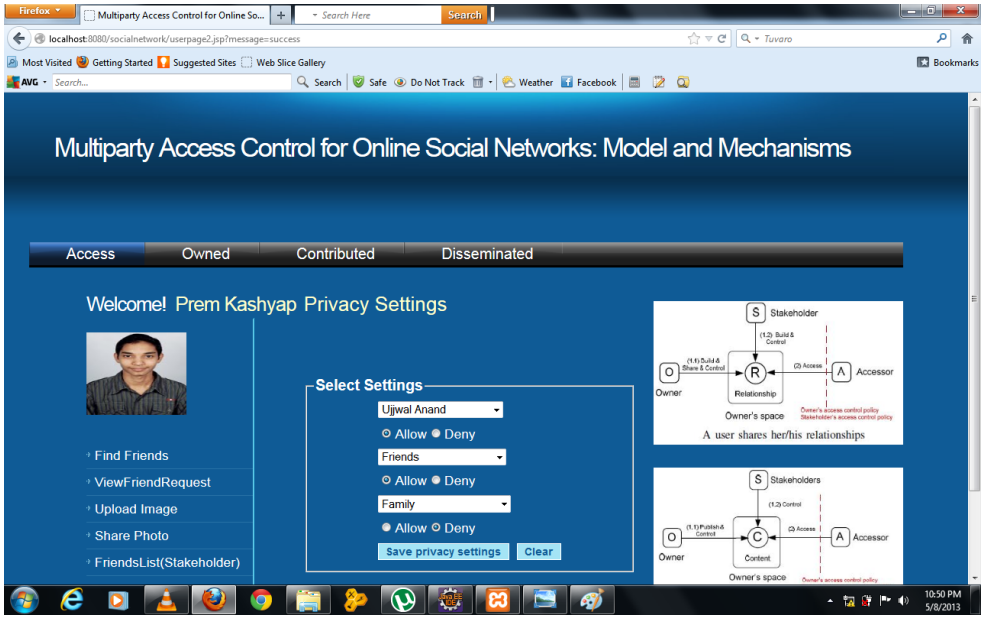


Fig 8.11 Privacy Setting

This page shows the user is selecting his own privacy either allow or deny.

8.1.12 IMAGE UPLOADED

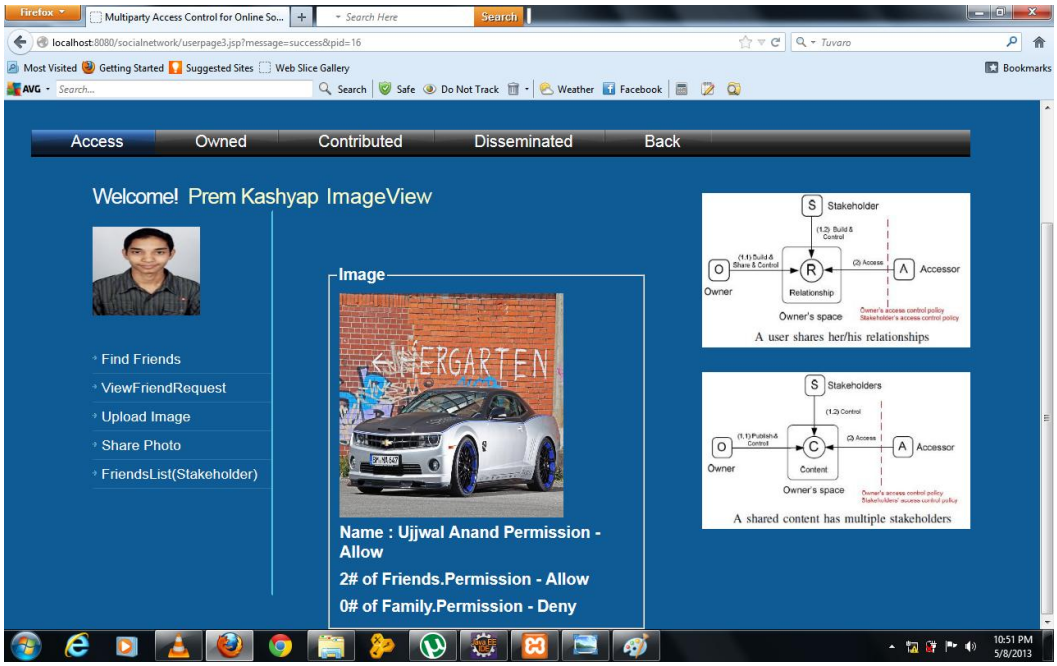


Fig 8.12 Image Uploaded

This page shows the image is uploaded and shows the selected privacy.

8.1.13 SEND FRIEND REQUEST

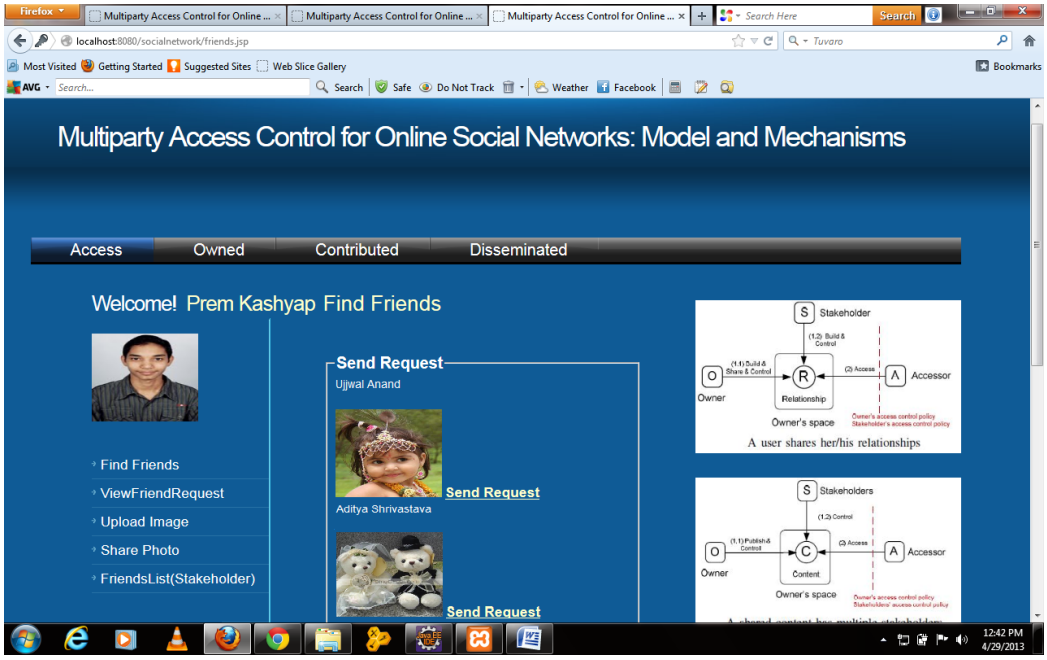


Fig 8.13 Send Friend Request

8.1.14 SELECT RELATIONSHIP

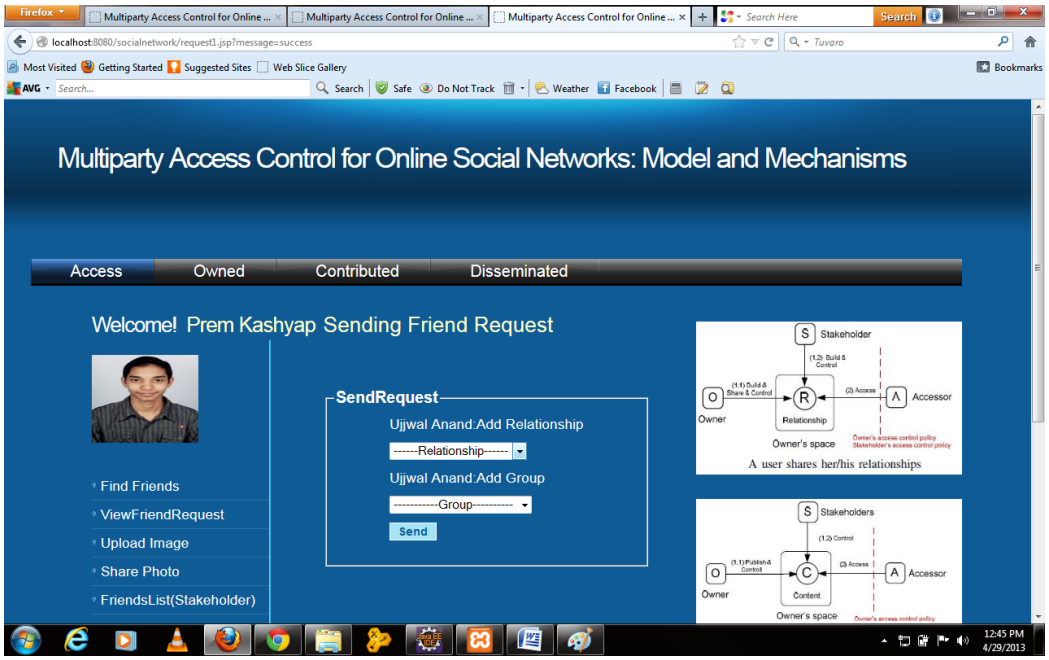


Fig 8.14 Select Relationship

8.1.15 SELECTING RELATIONSHIP

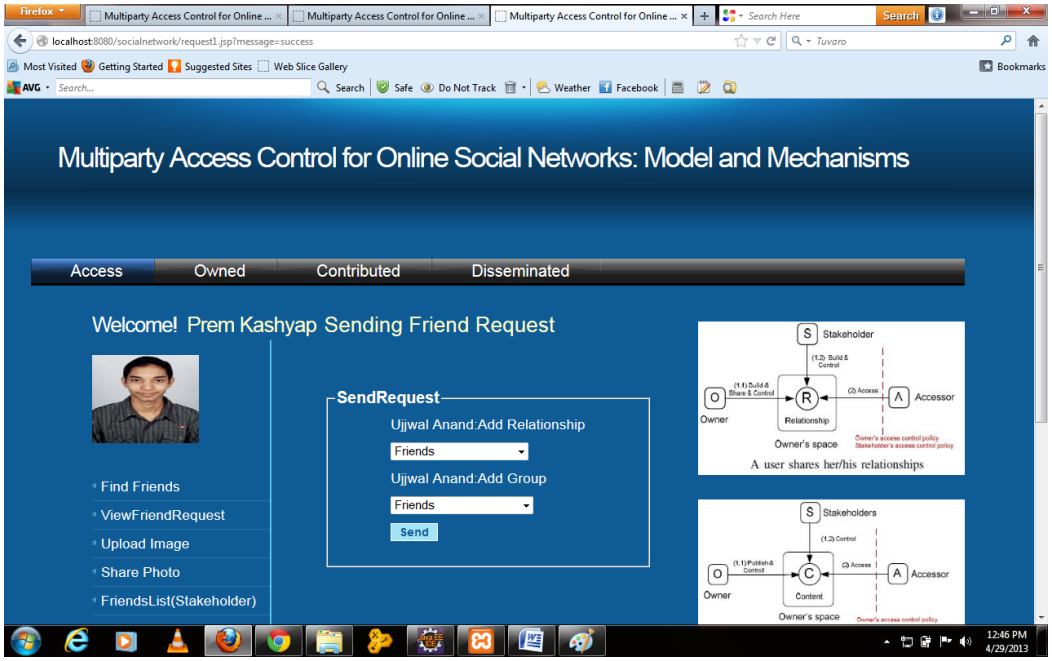


Fig 8.15 Selecting Relationship for Friend Request

8.1.16 RESPOND TO THE REQUEST

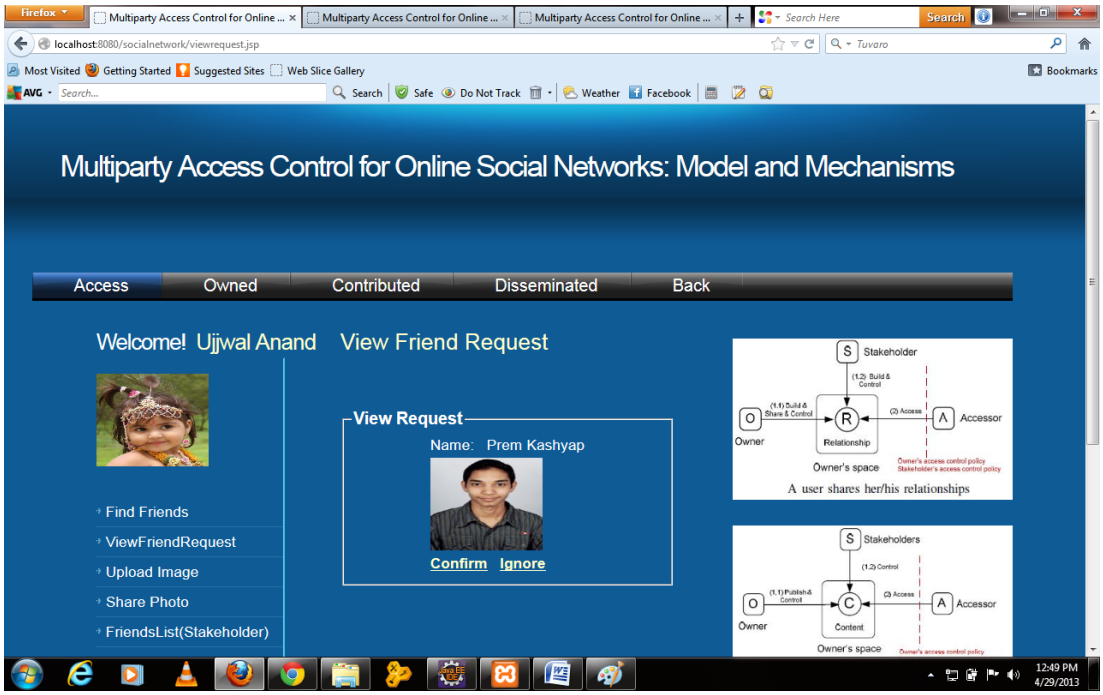


Fig 8.16 Respond To The Request

8.1.17 VIEW FRIEND REQUEST

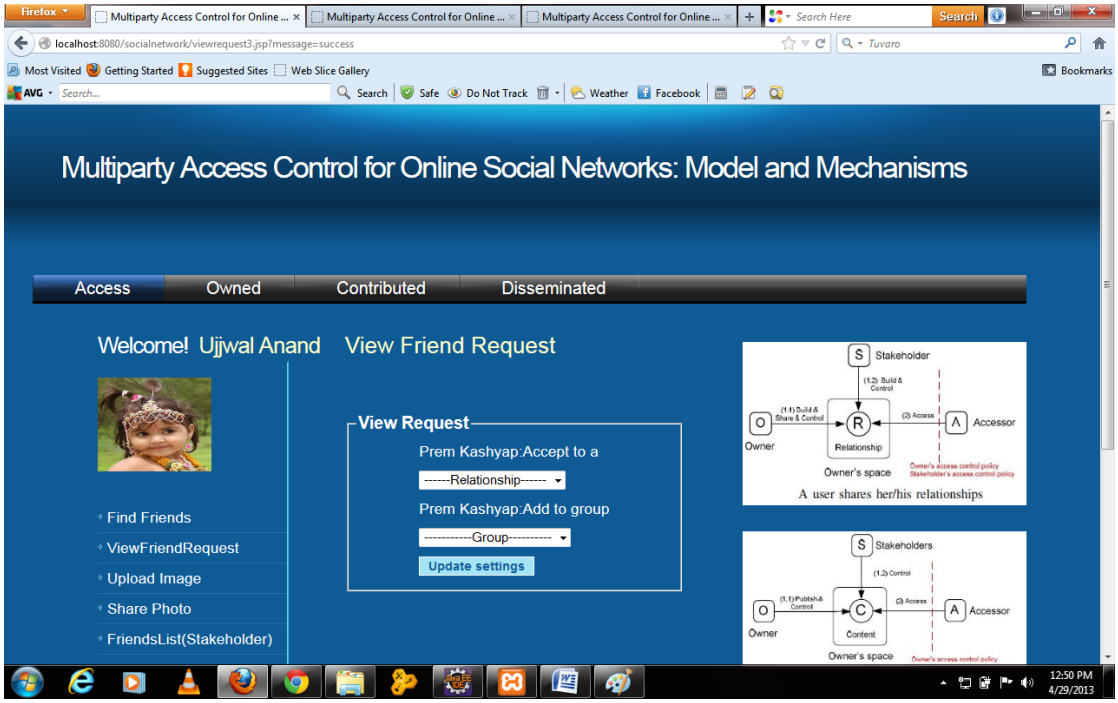


Fig 8.17 View Friend Request

8.1.18 RESPONDING REQUEST

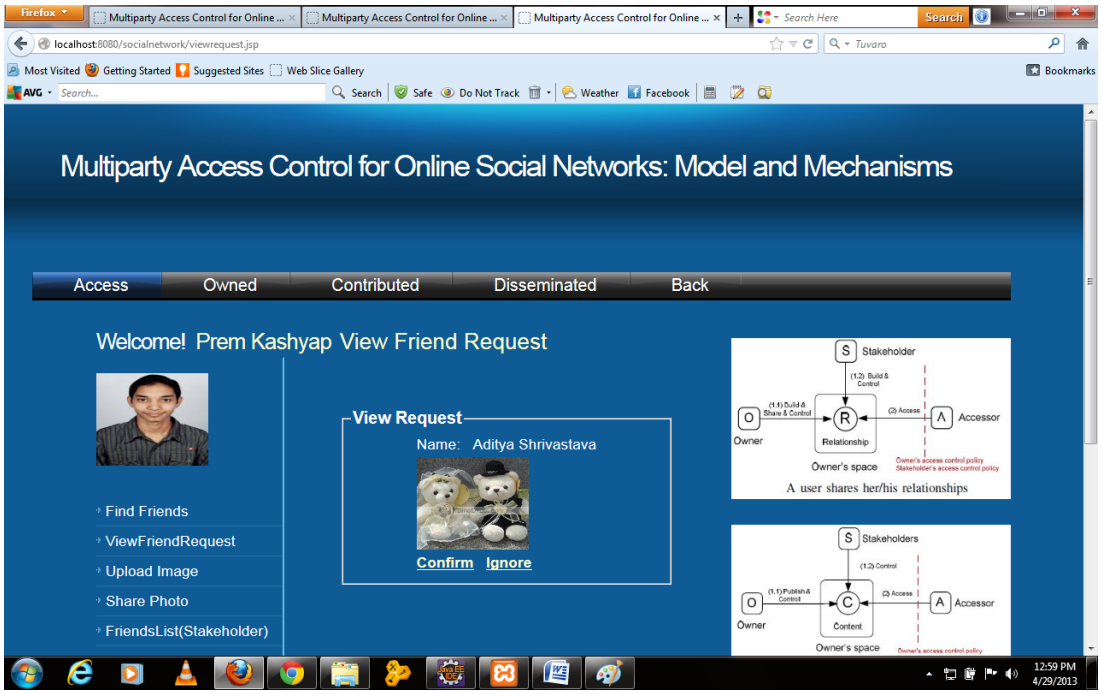


Fig 8.18 Responding Request

8.1.19 VIEW FRIEND LIST

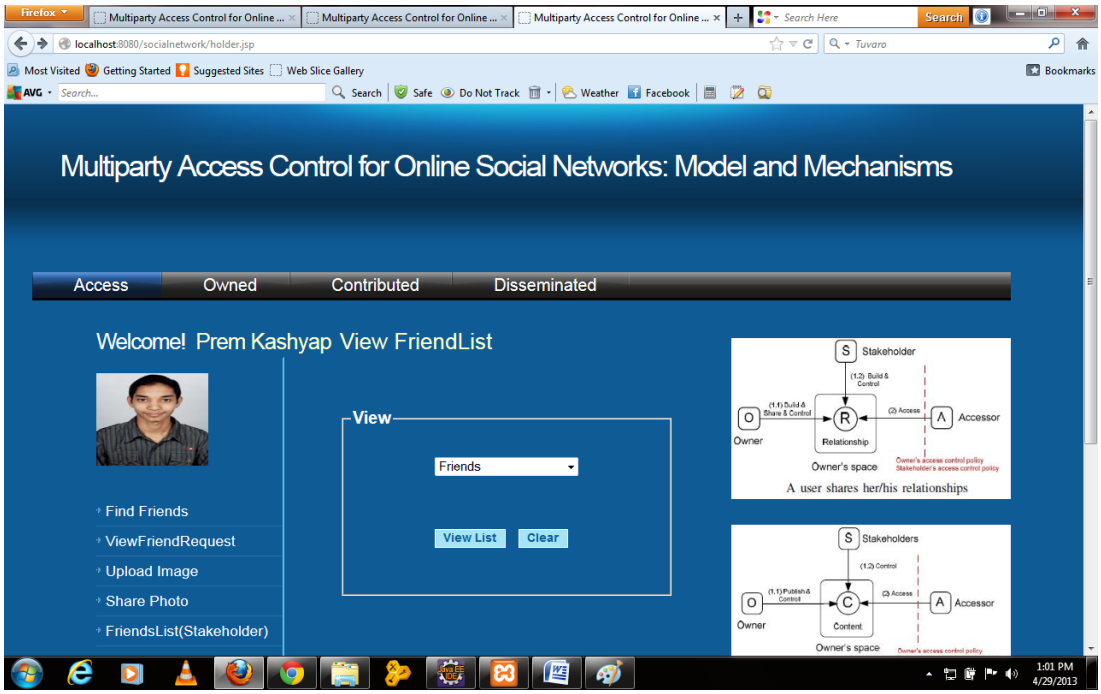


Fig 8.19 View Friend List

8.1.20 POSTED IMAGE

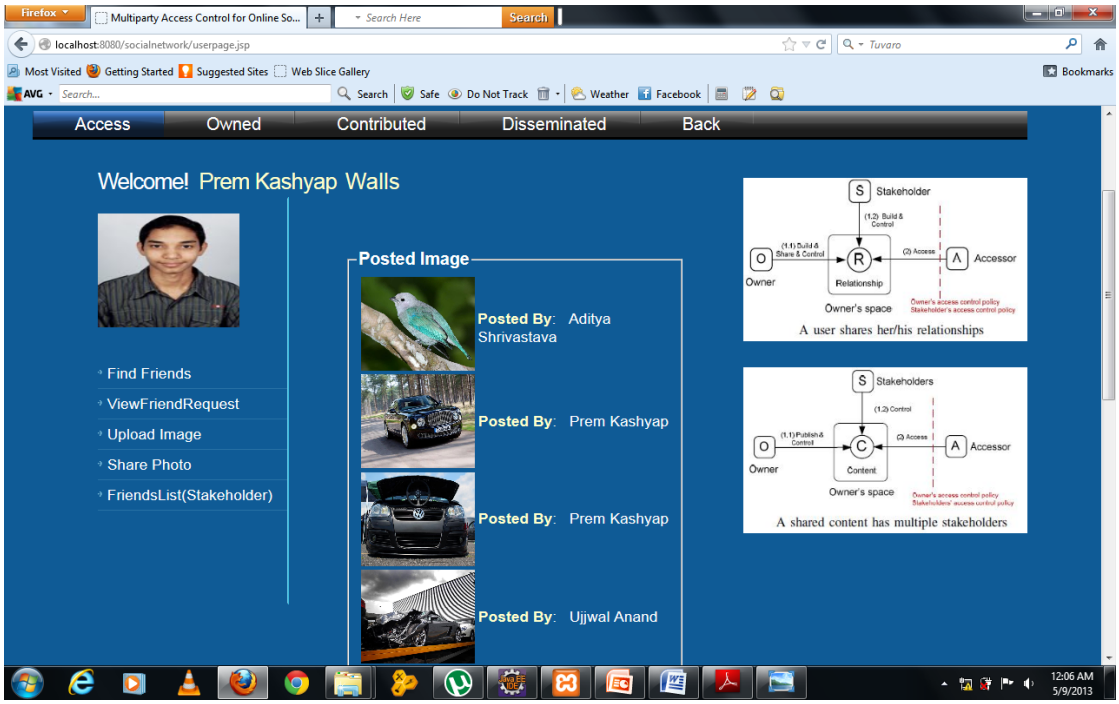


Fig 8.20 Postes Image

8.1.21 OWNED POSTED IMAGE

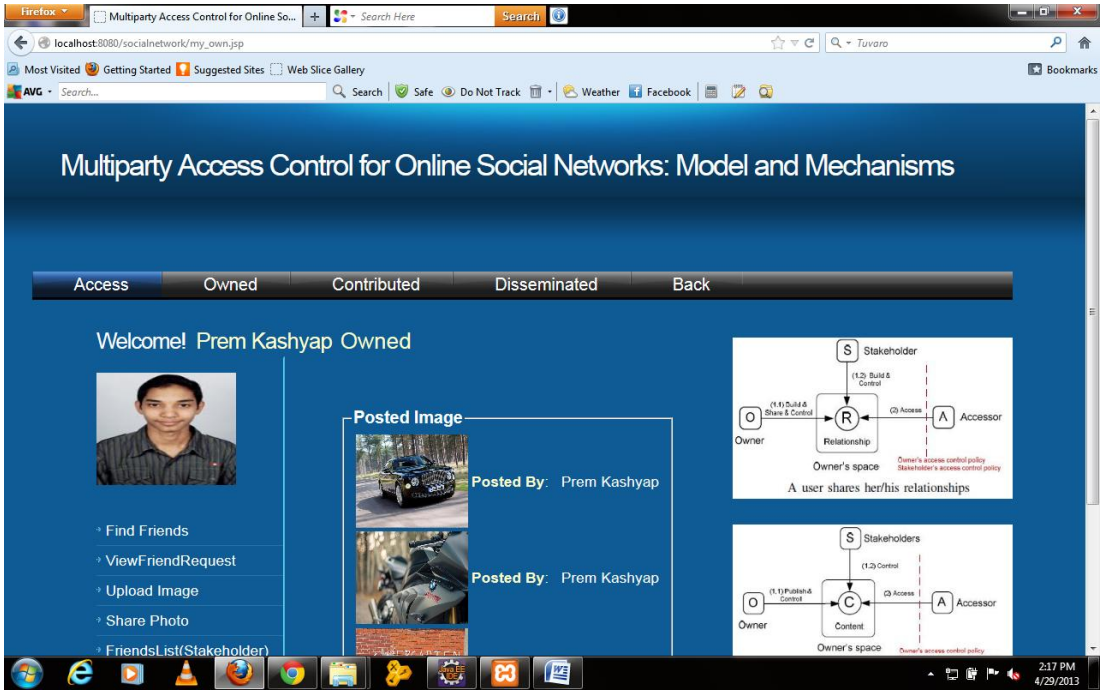


Fig 8.21 Owned Posted Image

8.1.22 DISSEMINATOR

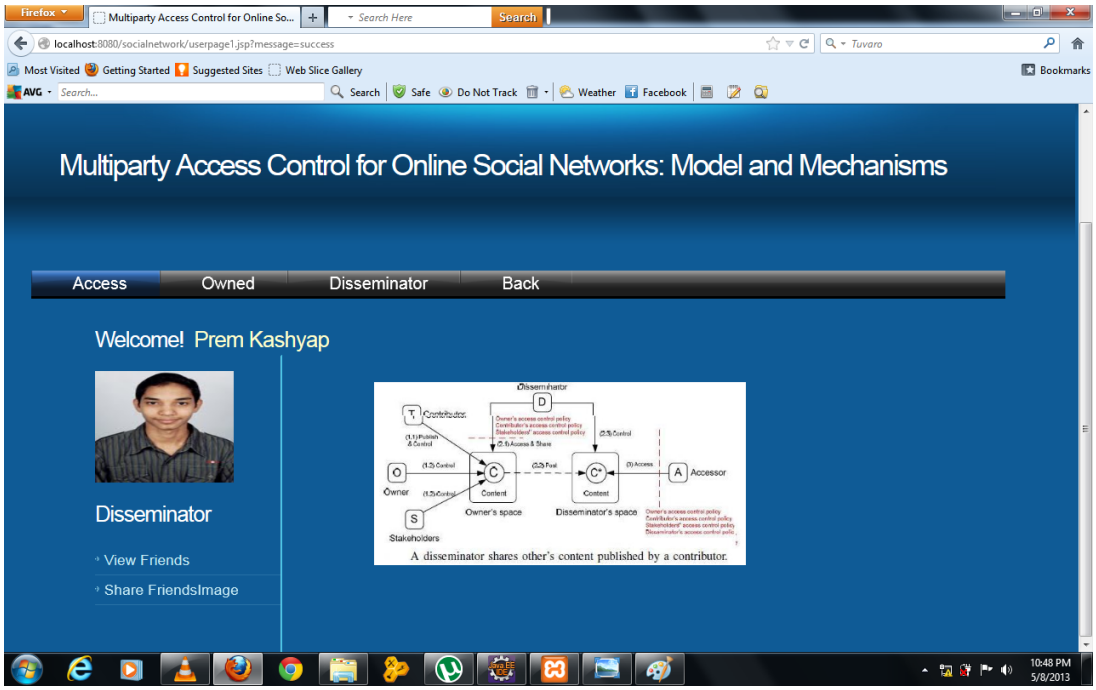


Fig 8.22 Disseminator

8.1.23 CONTRIBUTOR

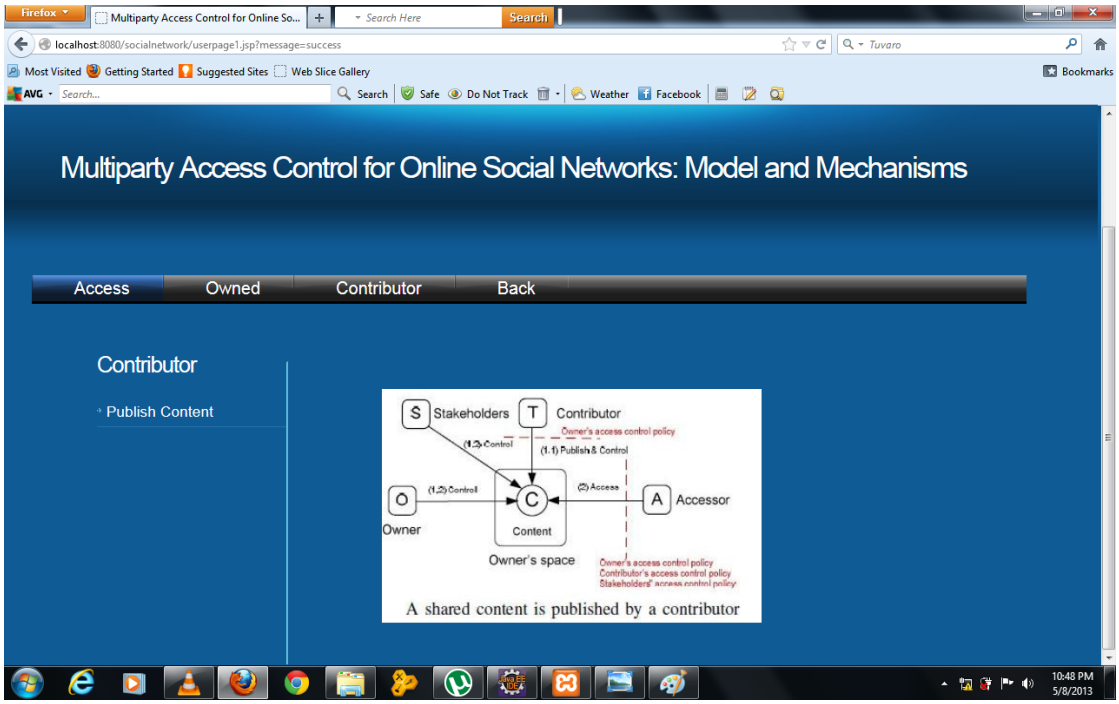


Fig 8.23 Contributor

CHAPTER 9

CONCLUSION

Muliparty access control for online social network has proposed a novel solution for collaborative management of shared data in OSNs. A multiparty access control model was formulated, along with a multiparty policy specification scheme and corresponding policy evaluation mechanism. In addition, they have introduced an approach for representing and reasoning about their proposed model. A proof-of-concept implementation of their solution called *MController* has been discussed as well, followed by the usability study and system evaluation of their method.

*****

CHAPTER 10

FUTURE ENHANCEMENT

As part of future work, they are planning to investigate more comprehensive privacy conflict resolution approach and analysis services for collaborative management of shared data in OSNs. Also, they would explore more criteria to evaluate the features of their proposed MPAC model. For example, one of their recent work has evaluated the effectiveness of MPAC conflict resolution approach based on the tradeoff of privacy risk and sharing loss.

In addition, users may be involved in the control of a larger number of shared photos and the configurations of the privacy preferences may become time-consuming and tedious tasks. Therefore, they would study inference-based techniques for automatically configure privacy preferences in MPAC. Besides, we plan to systematically integrate the notion of trust and reputation into their MPAC model and investigate a comprehensive solution to cope with collusion attacks for providing a robust MPAC servce in OSNs.

Their future work would integrate an effective collusion detection technique into MPAC. To prevent collusion activities, their current prototype has implemented a function for owner control, where the photo owner can disable any controller, who is suspected to be malicious, from participating in collaborative control of the photo. In addition, they would further investigate how users' reputations–based on their collaboration activities– can be applied to prevent and detect malicious activities in their future work.

*****

BIBLIOGRAPHY

Good Teachers are worth more than thousand books, we have them in Our Department.

ABBREVIATIONS:

OSNs : Online Social Networks

MPAC : Multiparty Access Control

PHP : Personal Home Page

MySQL : My Structure Query Language

JSP : Java Servelet Pages

ReBAC : Relationship Based Access Control

FoAP : Friend of a Friend

NSF : National Science Foundation

REFERENCES

[1]. Hongxin Hu, Gail-Joon Ahn. Multiparty Access Control for Online Social Network: Model and Mechanisms.ACM-2012

[2]. A. Besmer and H. Richter Lipford. Moving beyond untagging: Photo privacy in a tagged world. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1563–1572. ACM , 2010.

[3]. L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.

[4]. B. Carminati and E. Ferrari. Collaborative access control in online social networks. In *Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 231–240. IEEE, 2011.

[5]. B. Carminati, E. Ferrari, and A. Perego. Rule-based access control for social networks. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 1734–1744. Springer, 2006.

[6]. B. Carminati, E. Ferrari, and A. Perego. Enforcing access control in web-based social networks. ACM Transactions on Information and System Security (TISSEC), 13(1):1–38, 2009.

[7]. E. Carrie. Access Control Requirements for Web 2.0 Security and Privacy. In Proc. Of Workshop on Web 2.0 Security & Privacy (W2SP). Citeseer, 2007.

[8]. A. Squicciarini, M. Shehab, and F. Paci. Collective privacy management in social networks. In Proceedings of the 18th international conference on World wide web, pages 521–530. ACM, 2009.

[9]. H. Hu and G. Ahn. Multiparty authorization framework for data sharing in online social networks. In Proceedings of the 25th annual IFIP WG 11.3 conference on Data and applications security and privacy, pages 29–43. Springer-Verlag, 2011.

[10]. H. Hu, G. Ahn, and K. Kulkarni. Anomaly discovery and resolution in web access control policies. In Proceedings of the 16th ACM symposium on Access control models and technologies, pages 165–174. ACM, 2011.

[11]. E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In Proceedings of the 18th international conference on World wide web, pages 531–540. ACM, 2009.

[12]. A. Squicciarini, M. Shehab, and F. Paci. Collective privacy management in social networks. In Proceedings of the 18th international conference on World wide web, pages 521–530. ACM, 2009.

[13]. N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin. Access control policy combining: theory meets practice. In Proceedings of the 14th ACM symposium on Access control models and technologies, pages 135–144. ACM, 2009.

WEB LINKS:

[1]. Facebook Developers. http://developers.facebook.com/.

[2]. Facebook Privacy Policy. http://www.facebook.com/policy.php/.

[3]. Facebook Statistics. http://www.facebook.com/press/info.php?statistics.

[4]. Google+ Privacy Policy. http://http://www.google.com/intl/en/+/policy/.

[5]. Open Social Framework. http://code.google.com/p/opensocial-resources/.

[6]. The Google+ Project. https://plus.google.com.

[7]. http://java.sun.com.

[8]. http://www.roseindia.com/.

[9]. http://www.java2s.com/