#### A PROJECT REPORT ON

**Detection and Resolution of Conflict in Multi-Party Privacy Management for Social**

**Media**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY , PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

### BACHELOR OF ENGINEERING

**(Computer Engineering)**

**BY**

Student Name:- Shubham Vasant KAmble Exam No:- B150424267

Student Name:- Shreyas Atmaram Dargude Exam No:-B150424377

Student Name:- Mahendra Papan Bitake Exam No:-B150424234

Student Name:- Dattatray Bansilal Maske Exam No:-150424287

### Under The Guidance of

Prof. V.N. Dhawas

### DEPARTMENT OF COMPUTER ENGINEERING

**Sinhagad Institute Of Technology**

Lonavala

### SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE 2018 - 19

collegelogo.png

**CERTIFICATE**

This is to certify that the Project Entitled

### Detection and Resolution of Conflict in Multi-Party Privacy Management for Social Media

Submitted by

Student Name:- Shubham Vasant Kamble Exam No:B150424267

Student Name:- Shreyas Atmaram Dargude Exam No:B150424377

Student Name:-Mahendra Papan Bitake Exam No:B150424234

Student Name:-Dattatray Bansilal Maske Exam No:B150424287

is a bonafide work carried out by them under the supervision of V.N.Dhawas and it is approved for the partial fulfillment of the requirement of Savtribai Phule Pune university, Pune for the award of the degree of Bachelor of Engineering (Computer Engineering).

Prof. V.N.Dhawas Dr. S.D.Babar

Guide H.O.D

Dept. of Computer Engg. Dept. of Computer Engg.

Dr. M.S.Gaikwad

Principal

Sinhagad Institute Of Technology Lonavala

#### PROJECT APPROVAL SHEET

A Project Report Titled as

## Detection and Resolution of Conflict in Multi-Party Privacy Management for Social Media

Is verified for its originality in documentation, problem statement, proposed work and implementation successfully completed by

Student Name:-S.V. Kamble Exam No:B150424267

Student Name:-S.A. Dargude Exam No:B150424377

Student Name:-M.P. Bitake Exam No:B150424234

Student Name:-D.B. Maske Exam No:-B15024287

DEPARTMENT OF COMPUTER ENGINEERING SINHAGAD INSTITUTE OF TECHNOLOGY SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE ACADEMIC YEAR 2018-2019

Prof. V.N. Dhawas Dr. S.D. Babar

Guide H.O.D

### Abstract

Billions of items loaded on social media are basically owned by multiple users. However only the user who uploads the item has the authority to adjust its privacy settings (i.e., accessibility to the items.) Adjusting one’s privacy is a serious issue which is to be dealt efficiently, as user’s privacy preferences may collide. This might result into serious consequences e.g. losing jobs, cyber stalking. Items contain images that involve many people, comments which include various people etc. Multi-party privacy management is of great im- portance for users to properly preserve their privacy on social media. The proposed system includes first computational mechanism to resolve conflicts for multi-party management which is capable of operating in different situ- ations by modeling the concessions made by user to reach a solution to the conflict. In the proposed system the mediator uses various concession rules to resolve conflicts for multi-party privacy management .A tag line is given to the original sender to overcome the no concession rule .In this system friends are recommended on the basis of current user’s interest.

### Acknowledgments

Please Write here Acknowledgment.Example given as

It gives us great pleasure in presenting the preliminary project report on **‘Detection and Resolution of Conflict in Multi-Party Privacy Man- agement for Social Media’**.

I would like to take this opportunity to thank my internal guide **Prof. V.N.Dhawas** for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.

I am also grateful to **Prof. S. D. Babar**, Head of Computer Engineering Department, SINHAGAD INSTITUTE OF TECHNOLOGY for his indispensable support, suggestions.

In the end our special thanks to **JIO** for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Project.

Shubham Kamble

Shreyas Dargude

Mahendra Bitake

Dattatray Maske

(B.E. Computer Engg.)

**Contents**

1. [Synopsis](#_bookmark0) 1
   1. [Project Title](#_bookmark1) 2
   2. [Project Option](#_bookmark2) 2
   3. [Internal Guide](#_bookmark3) 2
   4. [Sponsorship and External Guide](#_bookmark4) 2
   5. [Technical Keywords (As per ACM Keywords)](#_bookmark5) 2
   6. [Problem Statement](#_bookmark6) 2
   7. [Abstract](#_bookmark7) 3
   8. [Goals and Objectives](#_bookmark8) 3
   9. [Relevant mathematics associated with the Project](#_bookmark9) 3
   10. [Names of Conferences / Journals where papers can be published](#_bookmark10) 6
   11. [Review of Conference/Journal Papers supporting Project idea](#_bookmark11) 6
   12. [Plan of Project Execution](#_bookmark12) 9
2. [Technical Keywords](#_bookmark13) 11
   1. [Area of Project](#_bookmark14) 12
   2. [Technical Keywords](#_bookmark15) 12
3. [Introduction](#_bookmark16) 14
   1. [Project Idea](#_bookmark17) 15
   2. [Motivation of the Project](#_bookmark18) 15
   3. [Literature Survey](#_bookmark19) 15
4. [Problem Definition and scope](#_bookmark20) 20
   1. [Problem Statement](#_bookmark21) 21
      1. [Goals and objectives](#_bookmark22) 21
      2. [Statement of scope](#_bookmark23) 21
   2. [Software context](#_bookmark24) 21
   3. [Major Constraints](#_bookmark25) 21
   4. [Methodologies of Problem solving and efficiency issues](#_bookmark26) 22
   5. [Scenario in which multi-core, Embedded and Distributed Com- puting used](#_bookmark27) 23
   6. [Outcome](#_bookmark28) 23
   7. [Applications](#_bookmark29) 23
   8. [Hardware Resources Required](#_bookmark30) 23
   9. [Software Resources Required](#_bookmark31) 24
5. [Project Plan](#_bookmark32) 25
   1. [Project Estimates](#_bookmark33) 26
      1. [Reconciled Estimates](#_bookmark35) 27
      2. [Project Resources](#_bookmark36) 27
   2. [Risk Management w.r.t. NP Hard analysis](#_bookmark37) 28
      1. [Risk Identification](#_bookmark38) 30
      2. [Risk Analysis](#_bookmark39) 30
      3. [Overview of Risk Mitigation, Monitoring, Management](#_bookmark43) 31
   3. [Project Schedule](#_bookmark44) 31
      1. [Project task set](#_bookmark45) 31
      2. [Task network](#_bookmark48) 33
      3. [Timeline Chart](#_bookmark49) 33
   4. [Team Organization](#_bookmark51) 34
      1. [Team structure](#_bookmark52) 34
      2. [Management reporting and communication](#_bookmark53) 34
6. [Software requirement specification](#_bookmark55) 36
   1. [Introduction](#_bookmark56) 37
      1. [Purpose and Scope of Document](#_bookmark57) 37
      2. [Overview of responsibilities of Developer](#_bookmark58) 37
   2. [Usage Scenario](#_bookmark59) 37
      1. [User profiles](#_bookmark60) 38
      2. [Use-cases](#_bookmark61) 38
      3. [Use Case View](#_bookmark63) 40
   3. [Data Model and Description](#_bookmark65) 41
      1. [Data Description](#_bookmark66) 41
      2. [Data objects and Relationships](#_bookmark67) 42
   4. [Functional Model and Description](#_bookmark69) 43
      1. [Data Flow Diagram](#_bookmark70) 43
      2. [Description of functions](#_bookmark73) 45
      3. [Activity Diagram:](#_bookmark74) 45
      4. [Non Functional Requirements:](#_bookmark76) 48
      5. [State Diagram:](#_bookmark77) 48
      6. [Design Constraints](#_bookmark79) 50
      7. [Software Interface Description](#_bookmark80) 50
7. [Detailed Design Document using Annexure A and B](#_bookmark81) 51
   1. [Introduction](#_bookmark82) 52
   2. [Architectural Design](#_bookmark83) 52
   3. [Data design (using Appendices A and B)](#_bookmark85) 54
      1. [Internal software data structure](#_bookmark86) 54
      2. [Global data structure](#_bookmark87) 55
      3. [Temporary data structure](#_bookmark88) 55
      4. [Database description](#_bookmark89) 55
   4. [Component Design](#_bookmark90) 56
      1. [Component Diagram](#_bookmark91) 56
      2. [Class Diagram](#_bookmark93) 56
      3. [Sequence Diagram](#_bookmark95) 59
8. [Project Implementation](#_bookmark97) 61
   1. [Introduction](#_bookmark98) 62
   2. [Tools and Technologies Used](#_bookmark99) 62
      1. [Software Resources Required](#_bookmark100) 62
      2. [Hardware Resources Required](#_bookmark101) 62
   3. [Methodologies/Algorithm Details](#_bookmark102) 63
      1. [Conflict Detection:](#_bookmark103) 63
      2. [Conflict Resolution:](#_bookmark104) 64
   4. [Verification and Validation for Acceptance](#_bookmark105) 64
9. [Software Testing](#_bookmark107) 66
   1. [Type of Testing Used](#_bookmark108) 67
   2. [Test Cases and Test Results](#_bookmark109) 68
10. [Results](#_bookmark112) 73
    1. [Screen shots](#_bookmark113) 74
    2. [Outputs](#_bookmark114) 74
11. [Deployment and Maintenance](#_bookmark117) 76
    1. [Installation and un-installation](#_bookmark118) 77
    2. [User help](#_bookmark119) 79
12. [Conclusion and Future Scope](#_bookmark120) 81
    1. [Conclusion](#_bookmark121) 82
    2. [Future Scope](#_bookmark122) 82

[References](#_bookmark122) 82

#### [Annexure A Laboratory assignments on Project Analysis of Al-](#_bookmark123)

|  |  |  |
| --- | --- | --- |
| [**gorithmic Design**](#_bookmark123) |  | **86** |
| [**Annexure B Laboratory assignments on Project Quality**](#_bookmark125)[**Reliability Testing of Project Design**](#_bookmark125) | [**and**](#_bookmark125) | **88** |
| [**Annexure C Project Planner**](#_bookmark126) |  | **90** |
| [**Annexure D Reviewers Comments of Paper Submitted**](#_bookmark127) |  | **92** |
| [**Annexure E Plagiarism Report**](#_bookmark128) |  | **94** |
| [**Annexure F Term-II Project Laboratory Assignments**](#_bookmark129) |  | **96** |
| [**Annexure G Information of Project Group Members**](#_bookmark130) |  | **98** |

**List of Figures**

[5.1 Software Plan](#_bookmark34) 26

* 1. [Use Case Diagram](#_bookmark64) 41
  2. [E-R Diagram](#_bookmark68) 43
  3. [DFD Level-0](#_bookmark71) 44
  4. [DFD Level-1](#_bookmark72) 45
  5. [Activity Diagram](#_bookmark75) 47
  6. [State Diagram](#_bookmark78) 49
  7. [System Architecture](#_bookmark84) 53
  8. [Component Diagram](#_bookmark92) 56
  9. [Class Diagram](#_bookmark94) 58
  10. [Sequence Diagram](#_bookmark96) 60

[10.1 Month wise conflict detection count](#_bookmark116) 75

**List of Tables**

|  |  |  |
| --- | --- | --- |
| [5.1](#_bookmark40) | [Risk Table](#_bookmark40) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 30 |
| [5.2](#_bookmark41) | [Risk Probability Definitions](#_bookmark41) . . . . . . . . . . . . . . . . . . . | 30 |
| [5.3](#_bookmark42) | [Risk Impact Definitions](#_bookmark42) . . . . . . . . . . . . . . . . . . . . . | 31 |
| [5.4](#_bookmark46) | [Risk 1 Overview](#_bookmark46) . . . . . . . . . . . . . . . . . . . . . . . . . . | 32 |
| [5.5](#_bookmark47) | [Risk 2 Overview](#_bookmark47) . . . . . . . . . . . . . . . . . . . . . . . . . . | 32 |
| [5.6](#_bookmark50) | [Timeline Chart](#_bookmark50) . . . . . . . . . . . . . . . . . . . . . . . . . . | 34 |
| [5.7](#_bookmark54) | [Management plan](#_bookmark54) . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
| [6.1](#_bookmark62) | [Use cases](#_bookmark62) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 39 |
| [8.1](#_bookmark106) | [Validation testing](#_bookmark106) . . . . . . . . . . . . . . . . . . . . . . . . . | 65 |
| [9.1](#_bookmark110) | [Test Cases](#_bookmark110) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 71 |
| [9.2](#_bookmark111) | [Test Cases](#_bookmark111) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 72 |
| [10.1](#_bookmark115) | [Detected conflict count per month](#_bookmark115) . . . . . . . . . . . . . . . . | 74 |
| [A.1](#_bookmark124) | [IDEA Matrix](#_bookmark124) . . . . . . . . . . . . . . . . . . . . . . . . . . . | 87 |
|  |  |  |

# CHAPTER 1 SYNOPSIS

SIT Lonavala , Department of Computer Engineering 2017-18

## Project Title

Detection and Resolution of Conflict in Multi-Party Privacy Management for Social Media

## Project Option

Internal Project

## Internal Guide

Prof. Internal Guide Name

## Sponsorship and External Guide

Our project has nothing any sponsorship.

## Technical Keywords (As per ACM Key- words)

* + 1. Social media
    2. Privacy
    3. Conflicts
    4. Multi-Party Privacy
    5. Social Networking Services
    6. Online Social Networks
    7. Friend Recommendation
    8. User Willingness

## Problem Statement

To detect and resolve multi-party privacy conflicts by using Conflict Detec- tion and Conflict Resolution algorithm.

## Abstract

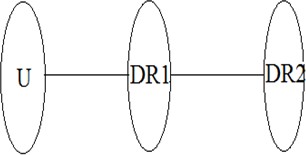
Billions of items loaded on social media are basically owned by multiple users. However only the user who uploads the item has the authority to adjust its privacy settings (i.e., accessibility to the items.) Adjusting one’s privacy is a serious issue which is to be dealt efficiently, as user’s privacy preferences may collide. This might result into serious consequences e.g. losing jobs, cyber stalking. Items contain images that involve many people, comments which include various people etc. Multi-party privacy management is of great im- portance for users to properly preserve their privacy on social media. The proposed system includes first computational mechanism to resolve conflicts for multi-party management which is capable of operating in different situ- ations by modeling the concessions made by user to reach a solution to the conflict. In the proposed system the mediator uses various concession rules to resolve conflicts for multi-party privacy management .A tag line is given to the original sender to overcome the no concession rule .In this system friends are recommended on the basis of current user’s interest.

## Goals and Objectives

* To detect the conflicts occurred using conflict detection algorithm
* To resolve the detected conflicts using conflict resolution algorithm
* To send tag line to original sender

## Relevant mathematics associated with the Project

The following terms shows in detail working of project.



Where,

A1: Share files with privacy policies provided by the user A2: Share files with privacy policies provided by the user

R1: Resulted web snippets provided by the sharing file process A3: Share files with privacy policies provided by the user

R2: Error (Conflict) routine in the accessing file process.

Given a set of negotiating users N = n1*, ., nk who co own an item i.e., there is one uploader* N who uploads the item to social media and the rest in N

∈

{ } −

are users affected by the item; and their individual (possibly conflicting)

privacy policies P*n*1*, ..., Pnk for that item*; *how can the negotiating users agree on with whom, from the set of the target users*

*T* = {*t*1*, , tm*}*, the item should be shared*?

*This problem can be decomposed into* : (1)*GiventhesetofindividualprivacypoliciesPn*1*, ..., Pnk of each negotiating user for the item, how can we identify if at least two policies*

*have contradictory decisions or conflicts*

*about whether or not granting target users T access to the item.* (2) *If conflicts are detected, how can we propose a solution to the conflicts found that respects as much as possible the preferences of negotiating users N.*

*Wedefinedthesetoftargetusersasasubsetoftheuserstoremainasgeneralaspossible*; *i.e., without for cingitto satisfy particular property .However, these to f target users could be further qualified as a particular subset of users satisfying any property without changing the subsequent formalization*; *e.g., these to f target users could be defined as the union of all of the negotiating users online friends.*

**Conflict Detection:** *Given a user n*∈ N, her groups G*n, her individual privacy policy Pn* =*< A, E >, and a user t* T ; we define the action function as:

∈

*act*(*Pn, t*) = 1 *if* ∃G ∈ G*n* : *t* ∈ G ∧ ∈ P*n.A* ∧ t ƒ∈ P*n.E act*(*Pn, t*) = 1 *if* ∃G ∈ G*n* : *t* ∈ G ∧ ƒ∈ P*n.A* ∧ t ∈ P*n.E act*(*Pn, t*) = 0 *otherwise*

Note that the definition of this function will vary according to the access control model used, but it will be defined in a similar way. That is, the idea is to be able to know, given a target user t, whether the privacy policy will grant/deny t access to the item regardless of the access control model being used. In particular, we assume that the available actions are either 0 (denying access) or 1 (granting access).

Given a set of negotiating users N and a set of target users T; a target user t ∈ T is said to be in conflict iff ∃a, b∈N with individual privacy poli- cies P*a and Pb respectively, so Va*[*t*] ƒ= V*b*[*t*]*.*

*Further, we say that the set of users in conflict C* T is the set that contains all the target users that are in conflict.

⊆

#### Input:

N = {negotiating users}

P*n*1*, ..., Pnk* = {*privacy policies*} *T* = {*target users*}

#### Output:

*C* = {*conflict*}

*Recall groups are disjoint. Otherwise, the complexity is* (|*U* |4)*.*

#### Conflict Resolution:

*Given user n*∈ N, her preferred privacy policy P*n, the maximum tie strength value δ*, a conflicting target user c ∈ C, the willingness of user n to accept changing her most preferred action for c is a function: W:NC→ [0,1]

*W* (*n, c*) = 1*/*2*.*((|*δ*-I*n*(*c*)|)*/*(*δ*+I*n*(*c*)) + (|*δ*-S*n*|)*/*(*δ*+S*n*))

#### I Do Not Mind (IDM) Rule

Assuming a negotiating user a N, and a conflicting target user c C, this concession can be formalized as the following fuzzy IF-THEN rule:

∈

∈

IF W(a,c) IS high THEN concede.

Concede means that user would accept changing her initial most preferred action to reach an agreement.

#### I Understand (IU) Rule

Assuming a negotiating user a N, and a conflicting target user c C, this concession can be formalized as the following fuzzy IF-THEN rule:

∈ ∈

IF W(a,c) IS low V*a*[*c*] = 1 b N,W(b,c) IS low V*b*[*c*] = 0 *THEN concede*

∧ ∧ ∃ ∈ ∧

#### No Concession (NC) Rule

*For the other cases in which neither IDM nor IU applies, then the mediator estimates that a negotiating user would not concede and would prefer to stick to her preferred action for the conflicting target user. For completeness, this can be formalized as*

*the following fuzzy IF* − *THEN rule assuming a negotiating user a*∈ N, and a conflicting target user c ∈ C:

IF W(a,c) IS low (V*a*[*c*] = 0 ( b N, W(b,c) IS low V*b*[*c*] = 0))

∧ ∨ ∈ ∧

*THEN do not concede*

#### Input:

N = {negotiating users}

P*n*1*, ..., Pnk* = {*privacy policies*} *T* = {*conflict*}

#### Output:

*O* = {*action vector*}

*The complexity of Algorithm* 2 *is O*(|*C*| × |*N* |2)

## Names of Conferences / Journals where papers can be published

3•.

IJARCSMS(International Journal of Advance Research in Computer Science And Management Studies)

* IJTRA(International Journal of Technical Research and Applications)

## Review of Conference/Journal Papers sup- porting Project idea

B. Carminati and E. Ferrari, proposed Collaborative access control in online social networks [1]. Authors show how topology-based access control can be enhanced by exploiting the collaboration among OSN users, which is the

essence of any OSN. The need of user collaboration during access control en- forcement arises by the fact that, different from traditional settings, in most OSN services users can reference other users in resources (e.g., a user can be tagged to a photo), and therefore it is generally not possible for a user to control the resources published by another user. For this reason, auhors introduce collaborative security policies, that is, access control policies iden- tifying a set of collaborative users that must be involved during access control enforcement. Also discuss how user collaboration can also be exploited for policy administration and we present architecture on support of collaborative policy enforcement.

L. Fang and K. LeFevre, presented Privacy wizards for social networking sites [2]. The intuition for the design comes from the observation that real users conceive their privacy preferences (which friends should be able to see which information) based on an implicit set of rules. Thus, with a lim- ited amount of user input, it is usually possible to build a machine learning model that concisely describes a particular users preferences, and then use this model to configure the users privacy settings automatically. As an in- stance of this general framework, we have built a wizard based on an active learning paradigm called uncertainty sampling. The wizard iteratively asks the user to assign privacy labels to selected (informative) friends, and it uses this input to construct a classifier, which can in turn be used to automatically assign privileges to the rest of the users (unlabeled) friends. To evaluate our approach, we collected detailed privacy preference data from 45 real Facebook users. Authors study revealed two important things. First, real users tend to conceive their privacy preferences in terms of communities, which can easily be extracted from a social network graph using existing techniques. Second, their active learning wizard, using communities as features, is able to rec- ommend high-accuracy privacy settings using less user input than existing policy-specification tools.

A. C. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede, presented A3p: Adaptive policy prediction for shared images over popular content sharing sites [3]. This system helps users compose privacy settings for their images. In particular, authors examine the role of image content and metadata as possible indicators of users’ privacy preferences. They propose a two-level image classification framework to obtain image categories which may be asso- ciated with similar policies. Then, they develop a policy prediction algorithm to automatically generate a policy for each newly uploaded image. Most im- portantly, the generated policy will follow the trend of the user’s privacy concerns evolved with time.

E. Gilbert and K. Karahalios, proposed Predicting tie strength with social media [4]. Authors present a predictive model that maps social media data to tie strength. The model builds on a dataset of over 2,000 social media ties and performs quite well, distinguishing between strong and weak ties with over 85% accuracy. They complement these quantitative findings with interviews that unpack the relationships we could not predict. The paper concludes by illustrating how modeling tie strength can improve social media design elements, including privacy controls, message routing, friend introductions and information prioritization.

A. Acquisti and R. Gross, presented Imagined communities: Awareness, information sharing, and privacy on the Facebook [5]. Authors survey a representative sample of the members of the Facebook (a social network for colleges and high schools) at a US academic institution, and compare the survey data to information retrieved from the network itself. They look for underlying demographic or behavioral differences between the communi- ties of the networks members and non-members; they analyze the impact of privacy concerns on members behavior; they compare members stated at- titudes with actual behavior; and they document the changes in behavior subsequent to privacy-related information exposure. Authors find that an individuals privacy concerns are only a weak predictor of his membership to the network. Also privacy concerned individuals join the network and reveal great amounts of personal information. Some manage their privacy concerns by trusting their ability to control the information they provide and the external access to it. However, they also find evidence of members mis- conceptions about the online communitys actual size and composition, and about the visibility of members profiles.

K. Thomas, C. Grier, and D. M. Nicol, proposed Unfriendly: Multi-party privacy risks in social networks [6]. In this paper, authors examine how the lack of joint privacy controls over content can inadvertently reveal sensitive information about a user including preferences, relationships, conversations, and photos. Specifically, they analyze Facebook to identify scenarios where conflicting privacy settings between friends will reveal information that at least one user intended remain private. By aggregating the information ex- posed in this manner, they demonstrate how a user’s private attributes can be inferred from simply being listed as a friend or mentioned in a story. To mitigate this threat, authors show how Facebook’s privacy model can be adapted to enforce multi-party privacy. They present a proof of concept ap- plication built into Facebook that automatically ensures mutually acceptable

privacy restrictions are enforced on group content.

A. Lampinen, V. Lehtinen, A. Lehmuskallio, and S. Tamminen, presented were in it together: Interpersonal management of disclosure in social net- work services [7]. The workload needed for managing privacy and publicness in current social network services (SNSs) is placed on individuals, yet peo- ple have few means to control what others disclose about them. This paper considers SNS-users concerns in relation to online disclosure and the ways in which they cope with these both individually and collaboratively. Authors present a framework of strategies for boundary regulation that informs both theoretical work and design practice related to management of disclosure in SNSs. The framework considers disclosure as an interpersonal process of boundary regulation, in which people are dependent on what others choose to disclose about them. The paper concludes by proposing design solutions supportive of collaborative and preventive strategies in boundary regulation that facilitate the management of disclosure online.

P. Wisniewski, H. Lipford, and D. Wilson, proposed Fighting for my space: Coping mechanisms for SNS boundary regulation [8]. Sharing information online via social network sites (SNSs) is at an all-time high, yet research shows that users often exhibit a marked dissatisfaction in using such sites. A compelling explanation for this dichotomy is that users are struggling against their SNS environment in an effort to achieve their preferred levels of pri- vacy for regulating social interactions. This research investigates users SNS boundary regulation behavior. This paper presents results from a qualita- tive interview-based study to identify coping mechanisms that users devise outside explicit boundary-regulation interface features in order to manage interpersonal boundaries. There categorization of such mechanisms provides insight into interaction design issues and opportunities for new SNS features.

## Plan of Project Execution

|  |  |  |  |
| --- | --- | --- | --- |
| **Schedule** | | **Date** | **Project Activity** |
| July | 1st week | 01/07/2017 | Formation of Project Group |
| 2nd week | 08/07/2017 | Project Topic Selection |
| 3rd week | 15/07/2017 | Synopsis Submission |
| August | 1st Week | 05/08/2017 | Presentation On Project Ideas |
| 2nd Week | 12/08/2017 | Submission Of Literature Survey |
| 3rd Week | 19/08/2017 | Feasibility Assessment |
| September | 1st Week | 02/09/2017 | Mid Sem Presentation |
| 3rd Week | 16/09/2017 | Design Of Mathematical Model |
| 4th Week | 23/09/2017 | End Sem Presentation. |
| October | 1st Week | 07/10/2017 | Report Preparation And  Submission |
| December | 3rd Week | 19/12/2017 | 1st module presentation |
| 4th Week | 26/12/2017 | Discussion and implementation of  2nd module |
| January | 1st Week | 02/01/2018 | Preparation for ANEC conference |
| 2nd Week | 09/01/2018 | Study of porter stemmer and tf  algorithm. |
| 3rd Week | 16/01/2018 | Discussion about modification to  Improved K-means |
| 4th Week | 23/01/2018 | 1st and 2nd module presentation |
| 5th Week | 30/01/2018 | Discussion on flow of project and  designing new module |
| February | 1st Week | 06/02/2018 | Modification of modules. |
| 2nd Week | 13/02/2018 | Designed test cases for our  module. |
| 3rd Week | 20/02/2018 | Worked on user interface. |
| March | 1st Week | 06/03/2018 | Integration of all modules. |
| 3rd Week | 20/03/2018 | Final Report and presentation. |

**CHAPTER 2 TECHNICAL KEYWORDS**

## Area of Project

Area of project is data mining(DM) has attracted a great deal of attention in the information industry, the healthcare sector and in society as a whole in recent years, due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowl- edge. DM is applied to discover interesting patterns and knowledge from large databases. The knowledge and intelligence gained can be used for a wide range of applications which may include market analysis, web services ranking customer retention, stock and production control, healthcare sys- tems and science exploration. DM and knowledge discovery are commonly seen as intelligent tools that help to accumulate and process data and make use of it. DM bridges many technical areas, including databases, statistics, neural networks, machine learning, web services and human computer inter- action. The set of DM processes used to extract and verify patterns in data is the core of the knowledge discovery process. We design and implement a novel conflict detection and resolutions algorithms to detect and resolve multi conflicts occurs in the social media.

## Technical Keywords

#### Social media

Social media is the collective of online communications channels dedi- cated to community-based input, interaction, content-sharing and col- laboration. Websites and applications dedicated to forums, microblog- ging, social networking, social curation, social bookmarking and wikis are among the different types of social media. Facebook is a popu- lar free social networking website that allows registered users to create profiles, upload photos and video, send messages and keep in touch with friends, family and colleagues. According to statistics from the Nielsen Group, Internet users within the United States spend more time on Facebook than any other website. Twitter is a free microblogging service that allows registered members to broadcast short posts called tweets. Twitter members can broadcast tweets and follow other users’ tweets by using multiple platforms and devices. Google+ (pronounced Google plus) is Google’s social networking project, designed to repli- cate the way people interact offline more closely than is the case in other social networking services. The projects slogan is Real-life shar- ing rethought for the web.

#### Privacy

Privacy concerns with social networking services is a subset of data privacy, involving the right of mandating personal privacy concerning storing, re-purposing, provision to third parties, and displaying of in- formation pertaining to oneself via the Internet.

#### Conflicts

Conflict is serious disagreement and argument about something impor- tant. If two people or groups are in conflict, they have had a serious disagreement or argument and have not yet reached agreement.

#### Social Networking Services

A social networking service (also social networking site’, SNS or social media) is an online platform which people use to build social networks orsocial relations with other people who share similar personal or career interests, activities, backgrounds or real-life connections. The defini- tion of online social networking encompasses networking for business, pleasure, and all points in between. Networks themselves have different purposes, and their online counterparts work in various ways. Loosely speaking, a social network allows people to communicate with friends and acquaintances both old and new.

#### User Willingness

In social networking user willingness deals with user trust.

# CHAPTER 3 INTRODUCTION

## Project Idea

Social media sites have an extensive presence in nowadays society. User can learn a lot of useful information about human behavior and interaction by paying attention to the information and relations of social media users. This information can be open or private. Ensuring the private data of the clients in informal organizations is a genuine concern. It proposes different method to solve these privacy conflicts. Current Social Media infrastructure lacks in multi-party privacy management. It makes users unable to appropriately control to whom these items are actually shared or not. Computational mech- anisms that are able to merge the privacy preferences of multiple users into a single policy for an item can help solve this problem. It propose the com- putational mechanism to solve the conflicts by using different algorithms like conflict detection and conflict resolution for multiple user privacy manage- ment in Social Media. It is able to adapt to different situations by modeling the concessions that users make to reach a solution to the conflicts.

## Motivation of the Project

In this paper, we present the first computational mechanism for social media that, given the individual privacy preferences of each user involved in an item, is able to find and resolve conflicts by applying a different conflict resolution method based on the concessions users may be willing to make in different situations. We also present a user comparing our computational mechanism of conflict resolution and other previous approaches to what users would do themselves manually in a number of situations. The results obtained suggest our proposed mechanism significantly outperformed other previously proposed approaches in terms of the number of times it matched participants behavior.

## Literature Survey

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Author, Title and Journal Name | Advantages | Disadvantage | Refer Points |
| 1 | K. Thomas,   1. Grier, and 2. M. Nicol, Unfriendly: Multi-party privacy risks in social networks, in Proc. 10th Int. Symp. Pri- vacy Enhancing Technol., 2010, pp. 236252. | 1. Privacy must extend beyond single-owner model    * Tags,   links,  mentions can ref- erence multiple users   * + Rely on these existing   features to distinguish who is at risk   1. Allow each user to specify global privacy policy content, regardless page it appears | 1. In absence of mutual friends, safe set of viewers tends towards empty set 2. Assume friends will consent to not sharing with   wider audi- ence other way to distinguish privacy-affected parties   1. Censor-   ship; prevents negative speech | 1. Adapt privacy con- trols:    * Grant users con- trol over all per-   sonal references, regardless where it appears   * + Includes tags, mentions, links   + Allow users to specify global privacy settings   users referenced   * + Determines mu- tually accept- able audience  1. Prototype solution as a Facebook applica- tion |
| 2 | A. Lampinen,  V. Lehtinen, A. Lehmuskallio, and S. Tam- minen, Were in it together: Interpersonal management  of disclosure in social network services, in Proc. SIGCHI Conf. | 1. The effective privacy manage- ment. 2. In collabo- rative strategy, asking another person to delete content 3. Reporting in- appropriate con- tent to service administrators |  | 1. This paper con- siders SNS-users con- cerns in relation to on- line disclosure and the ways in which they cope with these both individually and col- laboratively. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Author, Title and Journal Name | Advantages | Disadvantage | Refer Points |
|  | Human Fac- tors Comput. Syst., 2011, pp.  32173226. | 1. Supporting a non-serious in- terpretation 2. Interpret- ing content to be non-serious |  | 2. A framework of strategies for bound- ary regulation that in- forms both theoretical work and design prac- tice related to man- agement of disclosure in SNSs. |
| 3 | P. Wisniewski,  H. Lipford, and D. Wilson, Fighting for my space: Coping mechanisms for SNS boundary regulation, in Proc. SIGCHI Conf. Human  Factors Com- put. Syst., 2012,  pp. 609618. | 1. Privacy through effective interpersonal boundary reg- ulation serves as a way to improve how individuals con- nect and share with others 2. Improved in- terface design to better support interpersonal boundary reg- ulation could serve to im- prove, instead of prevent, higher levels of social interaction. | 1. Interpersonal boundary reg- ulation within online social networks as a means to align interactional privacy needs. | 1. This paper, in- vestigates users SNS boundary regulation behavior. 2. In this paper, filtering, ignoring, blocking, withdrawal, aggres- sion, compliance, and compromise represent coping mechanisms individuals use within SNSs to maintain their interpersonal boundaries. |
| 4 | A. Besmer and  H. Richter Lip- ford, Moving beyond untag- ging: Photo privacy in a tagged world, in Proc. SIGCHI | 1. The pro- posed system is a lightweight means for users to negotiate desired sharing. | 1. To improve privacy manage- ment in online social network- ing communities | 1. In this paper, using a focus group, we ex- plored the needs and concerns of users, re- sulting in a set of de- sign considerations for tagged photo privacy. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Author, Title and Journal Name | Advantages | Disadvantage | Refer Points |
|  | Conf. Human Factors Comput. Syst., 2010, pp.  15631572. | 2. Help users to achieve more de- sired privacy. |  | 2. This paper results identify the social ten- sions that tagging gen- erates, and the needs of privacy tools to ad- dress the social impli- cations of photo pri- vacy management. |
| 5 | J. M. Such,  A. Espinosa, and A. Garc Fornes, A sur- vey of privacy in multi-agent systems, Knowl. Eng. Rev., vol. 29, no. 03, pp. 314344, 2014. | 1. Interoperabil- ity and Open- ness 2. Pseudonym changer Agent 3. Disclosure Decision Making based on Multi- ple Criteria 4. Collective Disclosure Deci- sion Making 5. Learning the privacy sensitiv- ity of personal information 6. Personal Data Attribute Inference 7. Information dissemination detection 8. Integration of trust, rep- utation, and norms for pro- tecting against information dissemination |  | 1. In this paper, we have introduced the issue of privacy preser- vation and its relation to Multi-agent Sys- tems. 2. To prevent unde- sired information dis- semination based on trust and reputation on the one hand, and normative multi-agent systems on the other hand. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Author, Title and Journal Name | Advantages | Disadvantage | Refer Points |
| 6 | R. L. Fogues, J.  M. Such, A. Es- pinosa, and A. Garcia-Fornes, Open challenges in relationship- based privacy mechanisms  for social net- work services, Int. J. Human- Comput. In-  teraction, vol. 31, no. 5, pp.  350370, 2015. | 1. Including a content type as a new attribute of access control can improve the flexibility and expressiveness of privacy policies. 2. ReBAC mod- els in popular SNSs will im- prove the control of privacy for the users. | 1. ReBAC mod- els are complex 2. ReBAC model is not flexible | 1. This paper presents a list of privacy threats that can affect SNS users, and what requirements privacy mechanisms should fulfill to prevent these threats. 2. Visualization tools should explain to the users in an under- standable way how their information is disseminated accord- ing to a specific type of relationship. |
| 7 | R. Wishart,  D. Corapi, S. Marinovic, and  M. Sloman, Collaborative privacy policy authoring in a social network- ing context, in Proc. IEEE Int. Symp. Policies Distrib. Syst. Netw., 2010, pp. 18. | 1. The col- laborative  policy authoring process more user-friendly and accessible to average users of social networks. | 1. The scope of the policy can only be decreased by the nominated parties. 2. The inability of a user to claim co-ownership of a resource. 3. Provides limited help with authoring policies. | 1. In this paper, pro- pose a privacy-aware social networking ser- vice and then intro- duce a collaborative approach to authoring privacy policies for the service. 2. The approach per- mits the originators of content on the so- cial network to specify policies for the content they upload. |

**CHAPTER 4**

**PROBLEM DEFINITION AND SCOPE**

## Problem Statement

To detect and resolve multi-party privacy conflicts by using Conflict detection and conflict Resolution algorithm.

### Goals and objectives

* + - * To detect the conflicts occurred using conflict detection algorithm
      * To resolve the detected conflicts using conflict resolution algorithm
      * To send tag line to original sender

### Statement of scope

Use of this system is in online social media application for providing privacy controls.

•

Compare the individual privacy preferences of each negotiating user in order to detect conflicts among them.

•

Automated resolution of conflicts in multiparty privacy management for Social Media.

•

## Software context

Software which are used in application ellipse IDE, JDK 1.7 and Server which apache tomcat 7 and Store or load the Data base in Database as Mysql 5.1.

## Major Constraints

The following are the six constraints that are recognized as determining fac- tors in project management:

* Schedule - the time available to deliver a project
* Customer satisfaction - customer must be satisfied
* Quality - how successful and correct the project deliverable is

Scope - the features of the project (what the project does, how it does it)

•

* Risk - uncertainties, threats to the project
* Resources - people, budget and assets required to deliver the project

## Methodologies of Problem solving and ef- ficiency issues

Existing systems need too much human intervention during the conflict res- olution process, by requiring users to solve the conflicts manually or close to manually; e.g., participating in difficult-to-comprehend auctions for each and every co-owned item. Other approaches to resolve multi-party privacy conflicts are more automated but they only consider one fixed way of aggre- gating users privacy preferences without considering how users would actually achieve compromise and the concessions they might be willing to make to achieve it depending on the specific situation. We present the first compu- tational mechanism for social media that, given the individual privacy pref- erences of each user involved in an item, is able to find and resolve conflicts by applying a different conflict resolution method based on the concessions users may be willing to make in different situations. We compare the individ- ual privacy preferences of each negotiating user in order to detect conflicts among them. Each user is likely to have defined different groups of users, so privacy policies from different users may not be directly comparable. To compare privacy policies from different negotiating users for the same item, it considers the effects that each particular privacy policy has on the set of target users. Privacy policies dictate a particular action to be performed when a user tries to access the item. It assumes that the available actions are either 0 for denying access or 1 for granting access. The mediator figures the answer for every contention found by applying the concession rules through conflict resolution mechanism and the arrangement will be encoded into an activity vector.

**Algorithm and Technique:** Algorithm 1: Conflict Detection Algorithm 2: Conflict Resolution

## Scenario in which multi-core, Embedded and Distributed Computing used

Items shared through Social Media may affect more than one users priva- cye.g., photos that depict multiple users, comments that mention multiple users, events in which multiple users are invited, etc. The lack of multi-party privacy management support in current mainstream Social Media infrastruc- tures makes users unable to appropriately control to whom these items are actually shared or not. Computational mechanisms that are able to merge the privacy preferences of multiple users into a single policy for an item can help solve this problem. However, merging multiple users privacy preferences is not an easy task, because privacy preferences may conflict, so methods to resolve conflicts are needed. We propose the first computational mechanism to resolve conflicts for multi-party privacy management in Social Media that is able to adapt to different situations by modelling the concessions that users make to reach a solution to the conflicts.

## Outcome

* We detect conflicts which are occurred in social media.

We resolved detected conflicts for achieving privacy to original senders message.

•

We provide most privacy to message by sending tagline to original sender.

•

Reduce the amount of manual user interventions to achieve a satisfac- tory solution for all parties involved in multi-party privacy conflicts.

•

* We implement proposed work as web application.

## Applications

Our system is useful in social networking applications where users need privacy for item which particular user shared.

•

## Hardware Resources Required

* Processor -Pentium IV/Intel I3 core
* Speed - 1.1 GHz
* RAM - 512 MB (min)
* Hard Disk - 20GB
* Keyboard - Standard Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - LED Monitor

## Software Resources Required

Platform :

* Operating System - Windows XP/7
* Programming Language - Java/J2EE
* Software Version - JDK 1.7 or above
* Tools - Eclipse
* Front End - JSP
* Database - Mysql

# CHAPTER 5 PROJECT PLAN

## Project Estimates

In the waterfall model we start with the feasibility study and move down through the various phases up to Implementation, Testing, Deployment, maintenance and into live environment.

Following is a structure of waterfall model:

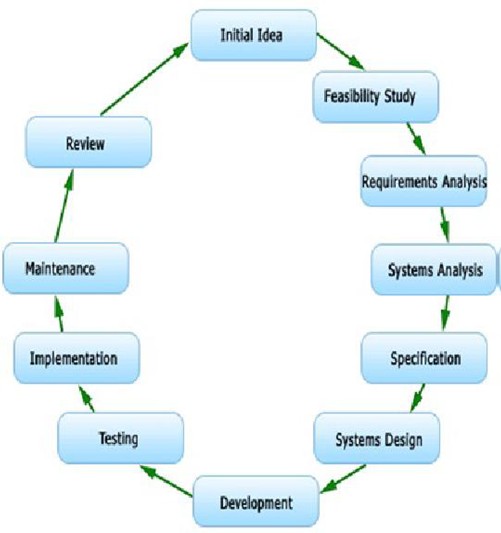


Figure 5.1: Software Plan

#### Requirement Gathering:

Requirement gathering and plan for the initial part of the project was as follows: Understanding the problem definition. Understanding the reliable factors for data sharing systems. Gathering information about required soft- ware. Gathering information about required Hardware Resources Preparing preliminary design of overall work flows of project deciding the modules re- quired for overall execution.

#### Analysis:

Analysis of system collects systems requirement. Detailed plan of project and estimation of budget is studied in this stage. It is useful for the future development of project. Document containing system requirements is end product of this phase. Management and user requirements, alternative plans

are further described by this document.

#### Design:

Design focuses on high level design like, software i.e. Net beans for J2ME software are needed and their interaction with each other, then interface de- sign to visualize how the project going to look like.

#### Implementation:

In this phase code is generated from the designs.

#### Testing:

System is tested in this phase. Individual modules are tested then whole sys- tem is tested. The system is tested to check inter related working of modules

i.e. integration testing and that the system does what the user requires.

#### Maintenance:

Maintenance is required for system. Because of some unexpected input values into the system the, changes in the system occurred.

* + 1. **Reconciled Estimates**

#### Cost Estimate

* + - * + Line Of code (LOC) is 3500 (Appr.)
        + LOC in KLOC (Kiloes LOC) is 3.5 KLOC. Effort:

The Effort is calculated by formula.

E=2.4\*(KLOC) ˆ 1.05

E=2.4\*(3.5) ˆ 1.05 E=8.94 Appr.

#### Time Estimates

The Development Time is calculated by formula. D = 2.5\*(E) ˆ 0.38

D = 2.5\*(8.94) ˆ 0.38 D = 5.75 Appr.

### Project Resources

The resource details include the following:

* Skill Level: skills may be knowledge of Java, Mysql, HTML, CSS.

Quantity: This is simply the number of professionals needed for com- pleting a task. Hours Required: According to project requirement we will conform hours.

•

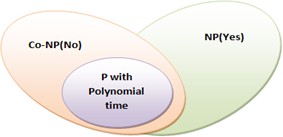
* Controller: Here we need admin as controller

## Risk Management w.r.t. NP Hard anal- ysis

Risk is inevitable in a business organization when undertaking projects. How- ever, the project manager needs to ensure that risks are kept to a minimal. Risks can be mainly divided between two types, negative impact risk and positive impact risk. Not all the time would project managers be facing neg- ative impact risks as there are positive impact risks too. Once the risk has been identified, project managers need to come up with a mitigation plan or any other solution to counter attack the risk.

#### What is P?

P is set of all decision problems which can be solved in polynomial time by a deterministic. Since it can be solved in polynomial time, it can be verified in polynomial time. Therefore P is a subset of NP.



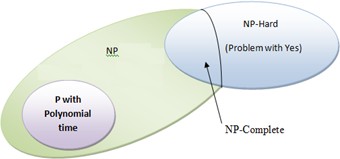
#### What is NP?

”NP” means ”we can solve it in polynomial time if we can break the normal rules of step-by-step computing”.

#### What is NP Hard?

A problem is NP-hard if an algorithm for solving it can be translated into one for solving any NP-problem (nondeterministic polynomial time) problem. NP-hard therefore means ”at least as hard as any NP-problem,” although it might, in fact, be harder.

#### What is NP-Complete?

Since this amazing ”N” computer can also do anything a normal computer can, we know that ”P” problems are also in ”NP”. So, the easy problems are in ”P” (and ”NP”), but the really hard ones are \*only\* in ”NP”, and they are called ”NP-complete”. It is like saying there are things that People can do (”P”), there are things that Super People can do (”SP”), and there are things \*only\* Super People can do (”SP-complete”).

So after implementing our system, it become NP complete.

### Risk Identification

Any unauthorized user should be prevented from accessing the system. Pass- word authentication can be introduced. To ensure the safety of the system, perform regular monitoring of the system so as to trace the proper working of the system. An internal staff has to be trained to ensure the safety of the system. He has to be trained to handle extreme error cases.

### Risk Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Risk Description | Probability | Impact | | |
| Schedule | Quality | Overall |
| 1 | System Failure | Low | Low | High | High |
| 2 | Connection Failure | Low | Low | Low | Low |

Table 5.1: Risk Table

|  |  |  |
| --- | --- | --- |
| Probability | Value | Description |
| High | Probability of occurrence is | *>* 75% |
| Medium | Probability of occurrence is | 26 − 75% |
| Low | Probability of occurrence is | *<* 25% |

Table 5.2: Risk Probability Definitions

|  |  |  |
| --- | --- | --- |
| Impact | Value | Description |
| Very high | *>* 10% | Schedule impact or Unacceptable quality |
| High | 5 − 10% | Schedule impact or Some parts of the project have low quality |
| Medium | *<* 5% | Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated |

Table 5.3: Risk Impact Definitions

### Overview of Risk Mitigation, Monitoring, Man- agement

The identification of Risk is central to the success and failure of the project, hence I have made a concentrated effort to minimize and even eliminate. Certain risk related. Software risk could be classified into categories. Internal and External risk, those risk which arise from the risk factor within the organization can be defined internal risk and the risk coming from outside is called external risk. Internal risk avoidance can be done by clear picturing the process, product risk.

Following are the details for each risk:

## Project Schedule

### Project task set

Major Tasks in the Project stages are:

* Task 1: Requirement Analysis (Base Paper Explanation).
* Task 2: Project Specification (Paper Work).
* Task 3: Technology Study and Design.
* Task 4: Coding and Implementation (Module Development).

|  |  |
| --- | --- |
| Risk ID | 1 |
| Risk Description | System failure |
| Category | Requirements |
| Source | This was identified during early development and testing |
| Probability | Low |
| Impact | High |
| Response | Accept |
| Strategy | Better Internet connection will solve this |
| Risk Status | Occurred |

Table 5.4: Risk 1 Overview

|  |  |
| --- | --- |
| Risk ID | 2 |
| Risk Description | Connection failure |
| Category | Requirements |
| Source | This was identified during early development and testing |
| Probability | Low |
| Impact | High |
| Response | Mitigate |
| Strategy | Proper authentication will resolve this issue |
| Risk Status | Identified |

Table 5.5: Risk 2 Overview

### Task network

Individual tasks and subtasks have interdependencies based on their se- quence. A task network is a graphic representation of the task flow for aproject. Project tasks and their dependencies are noted.

### Timeline Chart

A project timeline chart is presented.

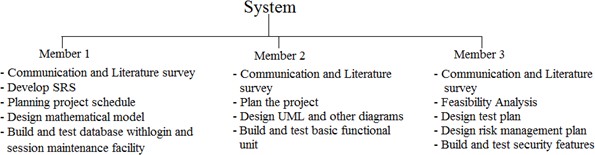
|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start date** | **End date** | **Duration** |
| **Initiate the project** | **04/8/17** | **24/8/17** | **21** |
| Communication | 04/8/17 | 10/8/17 | 7 |
| Literature survey | 11/8/17 | 17/8/17 | 7 |
| Define scope | 18/8/17 | 19/8/17 | 2 |
| Develop SRS | 20/8/17 | 24/8/17 | 5 |
| **Plan the project** | **25/8/17** | **17** | **42** |
| Design mathematical model | 25/8/17 | 31/8/17 | 7 |
| Feasibility Analysis | 01/9/17 | 07/9/17 | 7 |
| Develop work breakdown structure | 08/9/17 | 09/9/17 | 2 |
| Planning project schedule | 10/9/17 | 14/9/17 | 5 |
| Design UML and other diagrams | 15/9/17 | 21/9/17 | 7 |
| Design test plan | 22/9/17 | 28/9/17 | 7 |
| Design risk management plan | 29/9/17 | 5/10/17 | 7 |
| **Execute the project** | **05/01/18** | **29/03/18** | **84** |
| Build and test basic functional unit | 05/01/18 | 25/01/18 | 21 |

|  |  |  |  |
| --- | --- | --- | --- |
| Build and test Bluetooth mode | 16/02/18 | 08/03/18 | 21 |
| Build and test security features | 09/03/18 | 29/03/18 | 21 |

Table 5.6: Timeline Chart

## Team Organization

### Team structure

The team structure for the project is identified. Roles are defined.

### Management reporting and communication

Mechanisms for progress reporting and inter/intra team communication are identified as per assessment sheet and lab time table.

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Month** | **Description** |
| 1 | June | Discussion with guide regarding domain. Searching for IEEE paper for domain. |
| 2 | July | Short listing of IEEE papers within domain. Selection of IEEE paper. |
| 3 | August | Deciding Project name. Submission of Synopsis. |
| 4 | September | Requirement analysis. Designing of models. |
| 5 | October | Report preparation.  Stage-I report submission. |

Table 5.7: Management plan

# CHAPTER 6

**SOFTWARE REQUIREMENT SPECIFICATION**

## Introduction

This software requirement specification (SRS) report expresses complete de- scription about proposed System. This document includes all the functions and specifications with their explanations to solve related problems.

### Purpose and Scope of Document

The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system. This Document plays a vital role in the development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

### Overview of responsibilities of Developer

The developer is responsible for:

Developing the system, which meets the SRS and solving all the re- quirements of the system?

•

Demonstrating the system and installing the system at client’s location after the acceptance testing is successful.

•

Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.

•

Conducting any user training that might be needed for using the sys- tem.

•

* + - * Maintaining the system for a period of one year after installation.

## Usage Scenario

This section provides a usage scenario for the software. It is the organized information collected during requirements elicitation into use-cases.

### User profiles

|  |  |  |
| --- | --- | --- |
| Sr.No | Actor | Description |
| 1 | Admin | The Admin has to login himself. Admin can be performing operations like view all regis- tered users, check conflict detection, resolve conflicts. |
| 2 | User | The User has to firstly register itself by fill- ing the registration form and be the active member of the system. If a user is already the member of the system then he or she can perform login process. Then User can be per- forming operations like search friend, send friend request, view friend list, share image/ message, view conflict results, create group, view group, view share images, etc. |

### Use-cases

All use-cases for the software are presented. Description of all main Use cases using use case template is to be provided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Use Case | Description | Actors | Assumptions |
| 1 | Registration | The User have to firstly register it- self by filling the registration form and be the ac- tive member of the system | User | Assumption |
| 2 | Login | User and ad- min have to login to perform operations | User, Admin | Assumption |

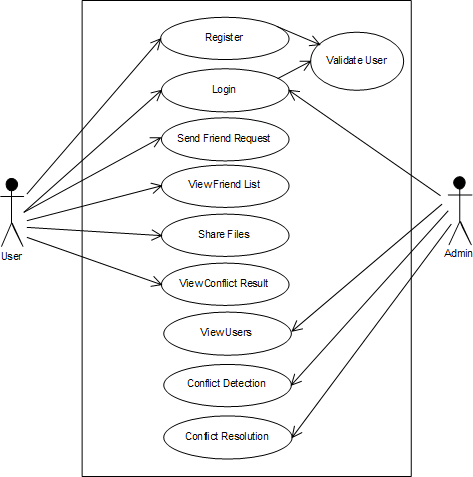
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Search friend | User can be search friend name to send friend request | User | Assumption |
| 4 | Send friend request | User can be send friend request | User, Admin | Assumption |
| 5 | View friend list | User have an authority to view his/her friend list | User | Assumption |
| 6 | Share files | User can be share message and image to his/her friends | User | Assumption |
| 7 | View conflict result | User can be view conflict result | User | Assumption |
| 8 | View Users | Admin have an authority  to view all registered users | Admin | Assumption |
| 9 | Conflict detection | Admin can be view detected all conflicts | Admin | Assumption |
| 10 | Conflict resolution | Admin can have an authority to resolve conflicts | Admin | Assumption |
| 11 | Logout | User and admin must logout the system | User, Admin | Assumption |

Table 6.1: Use cases

### Use Case View

A use case diagram is a graphical representation of a user’s interaction with the system and depicting the specifications of a use case. A use case diagram can show the different types of users of a system and the various ways in which they interact with the system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionality use cases are prepared and actors are identified. The purposes of use case diagrams can be as follows:

* + - * Used to gather requirements of a system.
      * Used to get an outside view of a system.
      * Identify external and internal factors influencing the system.
      * Show the interaction among the actors.

Figure 6.1: Use Case Diagram

## Data Model and Description

### Data Description

A database is a system intended to organize, store, and retrieve large amounts of data easily. It consists of an organized collection of data for one or more

uses, typically in digital form. MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site [(http://www.mysql.com/)](http://www.mysql.com/)) pro- vides the latest information about MySQL software.

* + - * MySQL is a database management system.
      * MySQL databases are relational.
      * MySQL software is Open Source.

The MySQL Database Server is very fast, reliable, scalable, and easy to use.

•

* + - * MySQL Server works in client/server or embedded systems.
      * A large amount of contributed MySQL software is available.

### Data objects and Relationships

In this system User register to system and login to this system. User enters login details. If password is matched then user get home page on that page user can enter query then he get result on result page.

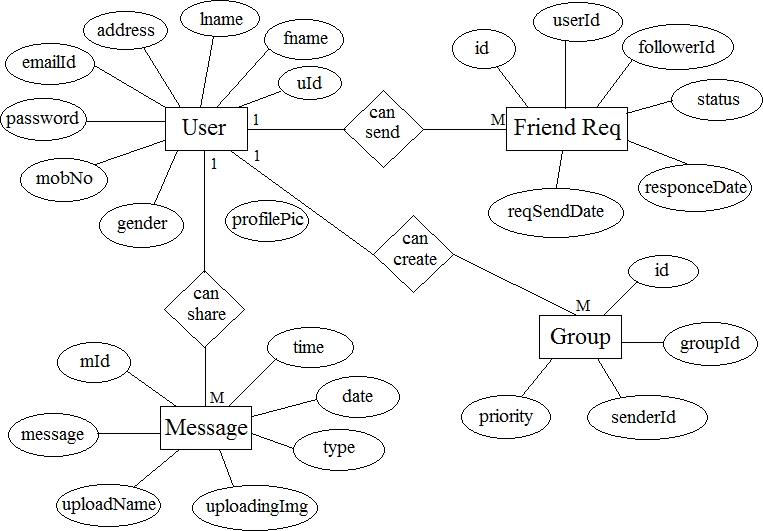


Figure 6.2: E-R Diagram

## Functional Model and Description

Functional Modelling is represented through a hierarchy of DFDs. The DFD is a graphical representation of a system that shows the inputs to the system, the processing upon the inputs, the outputs of the system as well as the internal data stores.

### Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the

timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

#### Level 0 Data Flow Diagram

The DFD Level 0 identifies external entities and processes of the system. Level 0 explains the architecture that would be used for developing a software product. A context diagram is a top level (also known as ”Level 0”) data flow diagram. It only contains one process node (”Process 0”) that generalizes the function of the entire system in relationship to external entities.

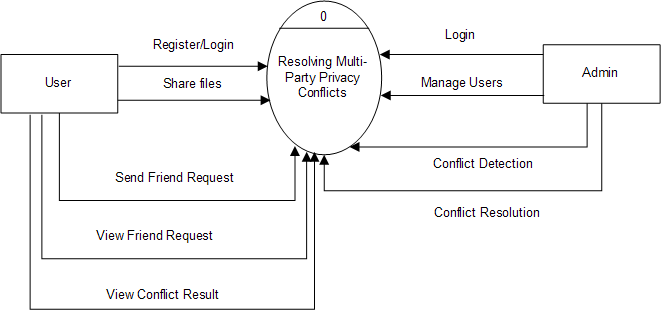


Figure 6.3: DFD Level-0

#### Level 1 Data Flow Diagram

This DFD Level 1 shows the main processes in the work and the entities involved in it. We usually begin withdrawing a context diagram, a sim- ple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with additional information about the major functions of the system. Level 1 is an extension of level 0 diagram.

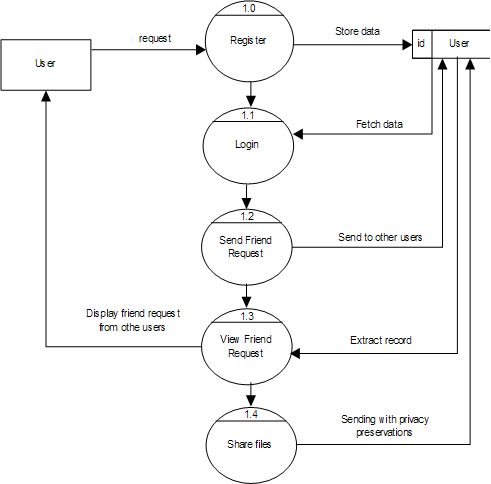


Figure 6.4: DFD Level-1

### Description of functions

Here user registers to the system. Then he login if his detail match with registered database fields. Then user enter query then system perform url matching and content matching .Then system display result to user.

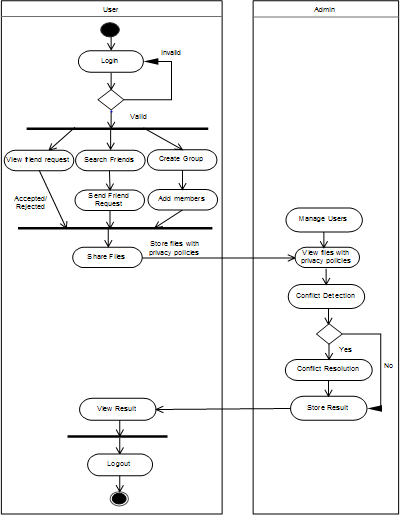
### Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise ac- tivities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both

computational and organizational processes (i.e. workflows). Activity dia- grams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most impor- tant shape types:

* + - * Rounded rectangles represent actions;
      * Diamonds represent decisions;
      * Bars represent the start (split) or end (join) of concurrent activities;
      * A black circle represents the start (initial state) of the work flow;
      * An encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen. Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for ex-pressing concurrency. However, the join and split symbols in activity diagrams only resolve this for sim- ple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

Figure 6.5: Activity Diagram

### Non Functional Requirements:

#### Interface Requirements

•

An interface requirement is a system requirement that involves an in- teraction with another system. The format of the interface requirement is such that it includes a reference (pointer) to the specific location in the definition document that defines the interface.

#### Performance Requirements

•

1. The software should be able to handle multiple requests and should provide a consistent view to all users.
2. The software should have a mechanism for authenticating a user sending a request.
3. There should be an efficient error reporting mechanism when the access to the information fails for some reason.

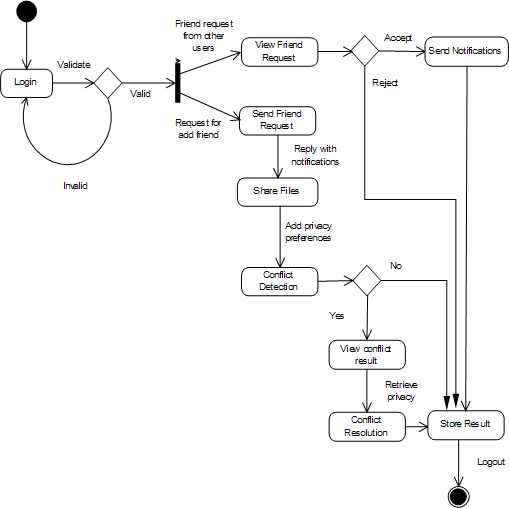
#### Software Quality Attributes

•

1. Runtime System Qualities: Runtime System Qualities can be mea- sured as the system executes.
2. Functionality: The ability of the system to do the work for which it was intended.
3. Performance:The response time, utilization, and throughput behav- ior of the system. Not to be confused with human performance or system delivery time.
4. Security:A measure of systems ability to resist unauthorized at- tempts at usage or behavior modification, while still providing service to legitimate users.
5. Availability: (Reliability quality attributes falls under this category) the measure of time that the system is up and running correctly; the length of time between failures and the length of time needed to resume operation after a failure.
6. Usability: The ease of use and of training the end users of the sys- tem. Sub qualities: learn ability, efficiency, affect, helpfulness, control.
7. Interoperability: The ability of two or more systems to cooperate at runtime.

### State Diagram:

The state transition diagram is a transitional flow of states within system. The states are represented in ovals and state of system gets changed when certain events occur. The transitions from one state to the other are repre- sented by arrows.

Figure 6.6: State Diagram

### Design Constraints

The following options can be placed after any datatype, adding other char- acteristics and capabilities to them.

**Primary Key** Used to differentiate one record from another. No two records can have the same primary key. This is obviously useful when it is imperative that no two records are mistaken to be the other.

•

**Auto Increment** A column with this function is automatically incre- mented one value (previous + 1) when an insertion is made into the record. The datatype is automatically incremented when ’NULL’ is inserted into the column.

•

**NOT NULL** Signifies that the column can never be assigned a NULL value.

•

ex.soc-sec-number INT PRIMARY KEY;No two soc-sec-number records can hold the same value.

ID-NUMBER INT AUTO-INCREMENT;Automatically increments in value, starting at ’1’, with every subsequent insertion.

### Software Interface Description

The java.sql package defines an interface called Java.sql.Driver that makes to

be implemented by all the JDBC drivers and a class called java.sql.DriverManager that acts as the interface to the database clients for performing tasks like connecting to external resource managers, and setting log streams. When

a JDBC client requests the DriverManager to make a connection to an ex- ternal resource manager, it delegates the task to an appropriate driver class implemented by the JDBC driver provided either by the resource manager vendor or a third party.

* + - * Operating System : Windows7
      * Application Server : Apache Tomcat7.0
      * Coding language : JAVA
      * Database : MySQL
      * Tool : Eclipse

# CHAPTER 7

**DETAILED DESIGN DOCUMENT USING ANNEXURE A AND B**

## Introduction

Social media sites have an extensive presence in nowadays society. User can learn a lot of useful information about human behavior and interaction by paying attention to the information and relations of social media users. This information can be open or private. Ensuring the private data of the clients in informal organizations is a genuine concern. It proposes different method to solve these privacy conflicts. As of late we have been viewing a huge increment in the development of on-line social systems. OSNs empower individuals to share individual and open data and make social associations with companions, relatives and different people or groups. Notwithstanding the fast increment in the utilization of interpersonal organization, it raises various security and protection issues. While OSNs permit clients to confine access to shared information, they as of now don’t give any component to thoroughly authorize security issue solver connected with different clients. Existing system need too much human intervention during the conflict res- olution process, by requiring users to solve the conflicts manually or close to manually; e.g., participating in difficult-to-comprehend auctions for each and every co-owned item. Other approaches to resolve multi-party privacy conflicts are more automated, but they only consider one fixed way of aggre- gating users privacy preferences without considering how users would actually achieve compromise and the concessions they might be willing to make to achieve it depending on the specific situation. In this project, we present the first computational mechanism for social media that, given the individ- ual privacy preferences of each user involved in an item, is able to find and resolve conflicts by applying a different conflict resolution method based on the concessions users may be willing to make in different situations. Also we recommend friends to active user based on his/her interest.

## Architectural Design

Figure 7.1 shows the system architecture of the proposed system. It com- pares the individual privacy preferences of each negotiating user in order to detect conflicts among them. Each user is likely to have defined different groups of users, so privacy policies from different users may not be directly comparable. To compare privacy policies from different negotiating users for the same item, it considers the effects that each particular privacy policy has on the set of target users. Privacy policies dictate a particular action to be performed when a user tries to access the item. It assumes that the available actions are either 0 for denying access or 1 for granting access. The mediator

figures the answer for every contention found by applying the concession rules through conflict resolution mechanism and the arrangement will be encoded into an activity vector. Also we recommend friends to active user based on his/her interest.

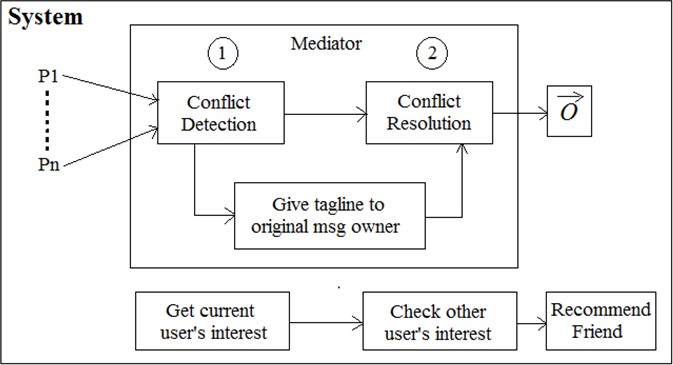
Figure 7.1: System Architecture

Figure 7.1 depicts an overview of the mechanism proposed:

* + 1. The mediator inspects the individual privacy policies of all users for the item and flags all the conflicts found. Basically, it looks at whether in- dividual privacy policies suggest contradictory access control decisions for the same target user. If conflicts are found the item is not shared preventively.
    2. The mediator proposes a solution for each conflict found. To this aim, the mediator estimates how willing each negotiating user may be to concede by considering: her individual privacy preferences, how sensi- tive the particular item is for her, and the relative importance of the conflicting target users for her.
    3. If all users accept the solution proposed, it will be applied. Otherwise, users will need to turn into a manual negotiation by other means.

First, privacy visualization tools already proved to be highly us- able for social media could be used to show and/or modify the suggested solution.

•

Second, users could define a default response to the solutions sug- gested, e.g., always accept the suggested solution without asking me.

•

* + 1. System recommends friends to current active user according to current users interest.

## Data design (using Appendices A and B)

The java.sql package defines an interface called Java.sql.Driver that makes to

be implemented by all the JDBC drivers and a class called java.sql.DriverManager that acts as the interface to the database clients for performing tasks like connecting to external resource managers, and setting log streams. When

a JDBC client requests the DriverManager to make a connection to an ex- ternal resource manager, it delegates the task to an appropriate driver class implemented by the JDBC driver provided either by the resource manager vendor or a third party.

### Internal software data structure

You should now have a basic understanding of the creation of tables, one of the most important concepts of the MySQL server. You now know that tables are constructed using datatypes, which when grouped together form a record. In the next section, we will begin learning how to manipulate the database. A database can be manipulated in four possible ways: addition, deletion, modification, and search. These topics will all be briefly covered in the following two sections. However, before we begin, I would like to highlight the fact that SQL, like many computer languages, is somewhat particular about command syntax. The slightest error in placement of a parentheses, comma, or semicolon will almost surely end in error. As a result, take care to be attentive of command syntax.

### Global data structure

The only kind of data available globally to the whole system is the database itself stored in Microsoft Access. After the JDBS connection any data stored can be fetched, updated, or deleted.

### Temporary data structure

The division and replication of data in cloud for optimal performance and security scheme for cloud storage application do not use any intermediate temporary files of its own. As data is being stored in a DBM system all the intermediate data it being stored by the DBMS by the application. Hence no intermediate files or data structures are used.

### Database description

Below are mentioned all tables, their corresponding attributes and a small description of each.

**Table Name :** User Registration

**Attributes :** Id, Name, Email Id, Password, Contact No, Gender, Profile Pic

**Description :** It holds all the general information of all the Users. Primary key of this table is User Id.

**Table Name :** Follower

**Attributes :** Id, User Id, Follower Id, Status, Request Send Date, Response Date

**Description :** It holds all the general information of all Followers i.e send friend request. Primary key of this table is Follower Id.

**Table Name :** Group

**Attributes :** Id, Group Id, Sender Id, Priority

**Description :** It holds all the information of all the Group that is created by user. Primary key of this table is Group Id.

**Table Name :** Admin

**Attributes :** Id, Email Id, Password.

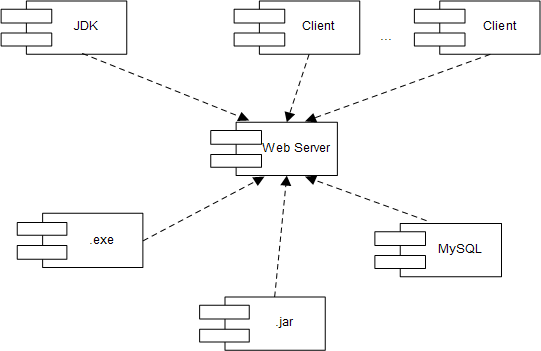
**Description :** This table will hold the login name and password for Admin. Primary key of this table is Admin Id.

## Component Design

The objective of the component design activity is to produce a design for an adaptable component that satisfies applicable products requirements. To represent component design we used class diagram and sequence diagram.

### Component Diagram

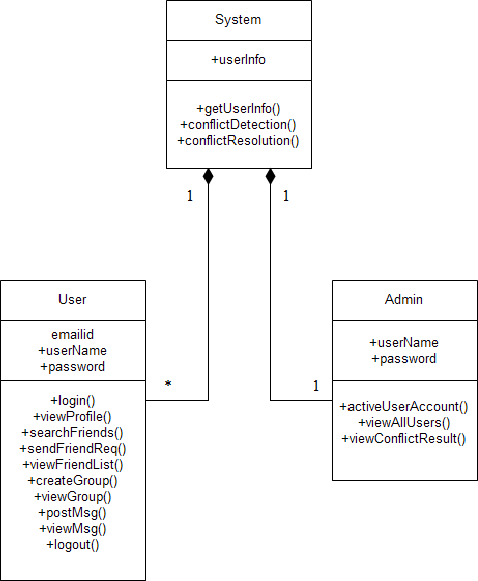
A Component Diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components. Components communicate with each other using interfaces. The interfaces are linked using connectors.

Figure 7.2: Component Diagram

### Class Diagram

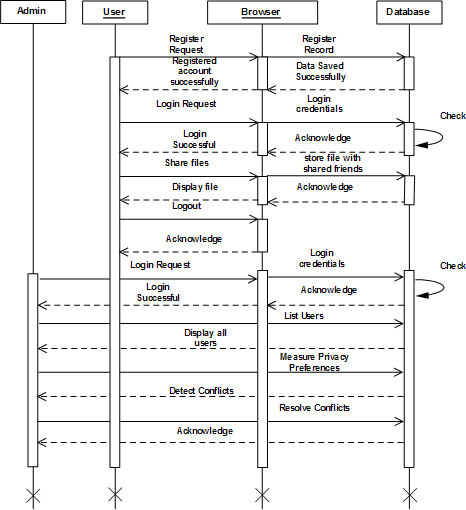
The class diagram is a static diagram. It represents the static view of an ap- plication. Class diagram is not only used for visualizing, describing and doc- umenting different aspects of a system but also for constructing executable

code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram. The purpose of the class diagram is to model the static view of an application.

Figure 7.3: Class Diagram

### Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes op- erate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under devel- opment. Sequence diagrams are sometimes called event diagrams or event scenarios.

Figure 7.4: Sequence Diagram

# CHAPTER 8 PROJECT IMPLEMENTATION

## Introduction

In this project, we present the first computational mechanism for social media that, given the individual privacy preferences of each user involved in an item, is able to find and resolve conflicts by applying a different conflict resolution method based on the concessions users may be willing to make in different situations. Also we recommend friends to active user based on his/her interest.

## Tools and Technologies Used

### Software Resources Required

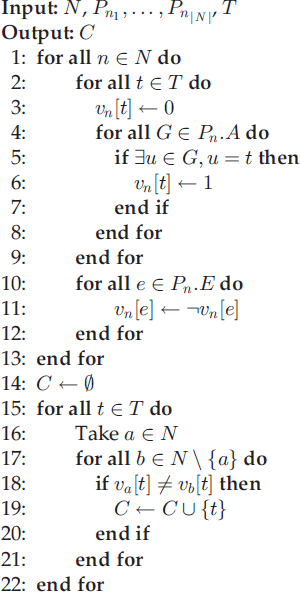
* + - * Operating System - Windows XP/7
      * Programming Language - Java/J2EE
      * Software Version - JDK 1.7 or above
      * Tools - Eclipse
      * Front End - JSP
      * Database - Mysql

### Hardware Resources Required

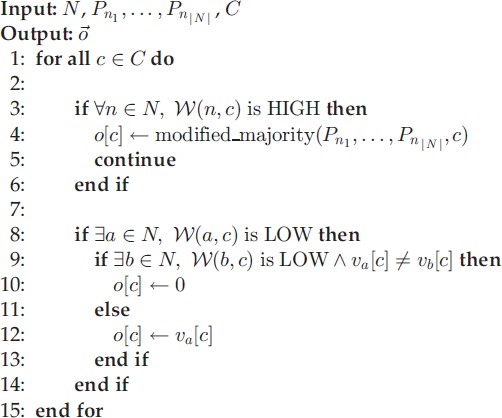
* + - * Processor - Pentium IV/Intel I3 core
      * Speed - 1.1 GHz
      * RAM - 512 MB (min)
      * Hard Disk - 20GB
      * Keyboard - Standard Keyboard
      * Mouse - Two or Three Button Mouse
      * Monitor - LED Monitor

## Methodologies/Algorithm Details

### Conflict Detection:



### Conflict Resolution:



## Verification and Validation for Acceptance

In the context of testing, ”Verification and Validation” are very widely and commonly used terms. Most of the times, we consider the terms same, but actually the terms are quite different.

There are two aspects of V and V tasks:

* Confirms to requirements (Producer view of quality)
* Fit for use (consumers view of quality)

Producers view of quality, in simpler terms means the developers perception of the final product.

Consumers view of quality means users perception of final product.

When we carry out the V and V tasks, we have to concentrate both of these

views of quality.

#### What is Verification?

Verification is a process of evaluating the intermediary work products of a software development lifecycle to check if we are in the right track of creat- ing the final product. In other words we can also state that verification is a process to evaluate the mediator products of software to check whether the products satisfy the conditions imposed during the beginning of the phase.

#### What is Validation?

Validation is the process of evaluating the final product to check whether the software meets the business needs. In simple words the test execution which we do in our day to day life are actually the validation activity which includes smoke testing, functional testing, regression testing, systems testing etc

#### Validation testing

Validation testing allows us to determine whether the system is able to satisfy the specified requirements or not. This can be done at the end of the imple- mentation process. Validation ensures that the system meets the specified requirements. In terms of simulation it can be said that the validation is the process of determining the degree to which the simulation and its associated data are exact representation of the real world.

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Customer Requirements** | **Validation Criteria** |
| 1 | Storage of nodes | Use of proper application. |
| 2 | Fetching stored data | Proper results with less execution time. |
| 3 | Scheduling of task | Duplicate task authentication avoided. |
| 4 | Faster execution | Minimum time required for execution. |

Table 8.1: Validation testing

# CHAPTER 9 SOFTWARE TESTING

## Type of Testing Used

#### Unit Testing:

Unit testing involves the design of test cases that validate that the in- ternal 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 com- ponent 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.

#### 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.

#### Functional Testing:

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 exer- cised.

•

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

•

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

#### System Testing:

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.

#### 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.

#### 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.

## Test Cases and Test Results

* + 1. **Unit Testing:** Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not un- common for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach:** Field testing will be performed man- ually 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.
    1. **Integration Testing:** 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.
    2. **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.

#### Test cases:

Testing of project problem statement using generated test data (using math- ematical models, GUI, Function testing principles, if any) selection and ap- propriate use of testing tools, testing of UML diagrams reliability.

#### Module-ID:1

Modules to be tested:-Registration

1. Enter the case insensitive Username click on Submit button. Expected: It should display error.
2. Enter the case sensitive Username click on Submit button. Expected: It should accept.
3. Enter the case insensitive Password click on Submit button. Expected: It should display error.
4. Enter the case sensitive Password click on Submit button. Expected: It should accept.
5. Enter the case insensitive Mobile Number click on Submit button. Expected: It should display error.
6. Enter the case sensitive Mobile Number click on Submit button. Expected: It should accept.
7. Enter the wrong address and click on Submit button. Expected: It should display error.
8. Enter the correct address and click on Submit button. Expected: It should accept.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test CaseID | Description | Test case I/P | Actual Result | Expected result | Test case criteria (P/F) |
| 101 | Enter the case insensitive User- name click on Submit button. | Username | Error comes | Error Should come | P |
| 102 | Enter the case sensitive User- name click on Submit button. | Username | Accept | Accept Username | P |
| 201 | Enter the case insensitive Pass- word click on Submit button. | Password | Error comes | Error Should come | P |
| 202 | Enter the case sensitive Pass- word click on Submit button. | Password | Accept | Accept Password | P |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 301 | Enter the case insensitive Mo- bile Number click on Submit button. | Mobile Number | Error comes | Error Should come | P |
| 302 | Enter the case sensitive Mobile Number click on Submit button. | Mobile Number | Accept | Accept Password | P |

Table 9.1: Test Cases

#### Module-ID:2

Modules to be tested:- Login

1. Enter the correct username and wrong password click on Submit but- ton. Expected: It should display error.
2. Enter the wrong username and correct password and click on Submit button. Expected: It should display error.
3. Enter the correct username and password and click on Login button. Expected: It should display welcome page.
4. After login with valid credentials click on back button. Expected: The page should be expired.
5. After login with valid credentials copy the URL and paste in another browser. Expected: It should not display the user’s welcome page.
6. Check the password with Lower case and upper case. Expected: Pass- word should be case sensitive.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test CaseID | Description | Test case I/P | Actual Result | Expected result | Test case criteria (P/F) |
| 001 | Enter the cor- rect username  and wrong password click onLogin button. | Username Password | Error comes | Error Should come | P |
| 002 | Enter the cor- rect username  and password and click on Login button. | Username Password | Accept | Accept Username | P |

Table 9.2: Test Cases

# CHAPTER 10 RESULTS

## Screen shots

Outputs / Snap shots of the results

## Outputs

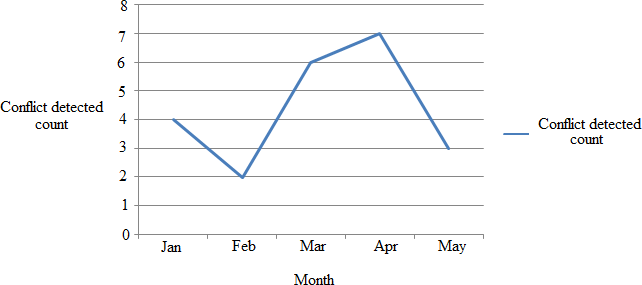
#### Result Table

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Month** | **Conflict detected count per month** |
| 1 | Jan | 4 |
| 2 | Feb | 2 |
| 3 | March | 6 |
| 4 | April | 7 |
| 5 | May | 3 |

Table 10.1: Detected conflict count per month

Above table shows the conflict detected count per month.

#### Result Graph

Figure 10.1: Month wise conflict detection count

The above graph shows the month wise conflict detection. In above graph X-axis represents months and Y-axis represents conflict detected count. Above graph shows that our system works effectively and efficiently and also shows that our system works better as compare to other state of art systems.

# CHAPTER 11

**DEPLOYMENT AND MAINTENANCE**

## Installation and un-installation

#### Install JAVA:

1. After the download completes, run the installer.
2. For, windows, the installer executable file have the .exe extension. Double-click the installer file to run it.
3. For Solaris and Linux platforms, the installer file has the.sh ex- tension. For these platforms, you need to make the installer files executable by using the following the following command: chmod

+x ”installer filename”.

1. If you downloaded the all or Java bundle you can customize your installation.

Perform the following steps at the welcome page of the installation wizard:

1. Click customizes.
2. In the customize Installation dialog box, make your selections. Click OK.
3. At the Welcome page of the installation wizard click next.
4. At the License agreement page, review the license agreement, click the acceptance check box, and click next.
5. At the JUnit License Agreement page, decide if you want to install JUnit and click the appropriate option, click Next.
6. If you are installing Apache Tomcat, on its installation page, ac- cept the default installation directory or specify another installa- tion location. Click Next.
7. At the summary page, do the following:
8. Verify that the list of components to be installed is correct.
9. Select the check for updates check box if you want to check the update center for possible updates and have the JUnit library installed during the installation.

(Provided you accepted the license in step 5)

1. Verify that you have adequate space on your system for the in- stallation.
2. Click Install to begin the Installation
3. At the Setup complete page, provide anonymous usage data if desired, and click
4. Finish.

#### Install JDK:

•

The latest version of \*Java JDK from sun Microsystems Double-click on the install file and it should open an installer Click next, then read and accept the license. On the next screen you will encounter some options. Just leave alone and click next unless you know what you are doing. The next page you encounter should install (and in some cases download) the Java Development Kit. After the installer is finished, open run by clicking start? Run or by typing Windows Key+R. In the text box, type ”cmd” and click OK. A simple window should be opened with a black background and a text prompt. This is called the ”Command Prompt”. My command prompt background is red, but it could really be any color. After focusing the window type ”Javac” and press enter. If the prompt returns something along the lines of: ”’javac’ is not recognized as an internal or external command, oper- able program or batch file” then continue with the next step. Open the properties of ”My Computer” by either right clicking the icon on the desktop or right-clicking Start? My Computer. When the popup menu opens, scroll to the bottom and select ”Properties”. This should open a window named ”System Properties”. Click on the ”Advanced” tab and then click ”Environment Variables”. Next, another window opens with a lot of confusing sentences and letters. Double-click on the ”Path” variable on either of the option boxes. It is recommended to edit the variable in the box ”User variables for (your username)” Once the variable is opened, a text box in yet another window appears. Careful not to delete anything in this box. At the end of text box, add a semi-colon if there is not one already, and add ”C: files 1.6.0” to the text box. This is assuming you did not change the file path of the installation. Click ”Apply” and ”OK” to all the windows you have just opened. Open the command prompt again, while following steps 6-9 to see if that ”Javac” command works.

#### Install MySQL Query Browser:

* 1. MySQL GUI Tools run on Windows, Linux, and Mac OS X. You can find MySQL GUI Tools for the operating system of your choice on the MySQL GUI Tools Downloads page.
  2. Choose the version 1.2.17 and click on it.
  3. After download, MySQL GUI Tools can be installed on all Win- dows operating systems using the Windows Installer (.msi) instal- lation package. The MSI package is contained within a ZIP archive named mysql-gui-tools-version-win32.msi, where version indicates the MySQL GUI Tools version. Click on .exe file.
  4. At the Setup Type window you may choose a complete or custom installation. If you do not wish to install all of the MySQL GUI Tools choose the custom option. Custom installation also gives you the option of installing support for languages other than En- glish. MySQL GUI Tools supports, German, Greek, Japanese, Polish, and Brazilian Portuguese.
  5. Unless you choose otherwise, MySQL GUI Tools are installed in C PROGRAM FILES MySQL Tools for version where machine and version is the version number of MySQL GUI Tools. The might be C:n Program Files or C: n program.
  6. At the startup enter username ”root” and password ”root”. Once the MySQL database is installed follow the below given steps for limiting account resources. Re- source limits are stored in the max questions, max updates, max connections and max user connec- tions columns. If your user table doesn’t have these columns, it must be upgraded? Please follow below given steps:
  7. Go to MySQL prompt
  8. Type: use mysql?
  9. Type: select max questions, max updates, max connections, max user connections, user from user?
  10. Output from Step 4 3 should have all zero’s (0) in all rows. If the output of result is not as per Step 4 then type:

## User help

1. Register : User must have to register himself
2. Login: Here simply by entering login details all users can logged in
3. Search friends : One user can be send friend request to other users
4. Create group and give priority : User must have to create group and give priority to each friend in a group
5. Send post : User can be send post with images and message to public or private friends

# CHAPTER 12

**CONCLUSION AND FUTURE SCOPE**

## Conclusion

In this Project, we make an attempt to use conflict detection and conflict resolution techniques in social media. To reduce the amount of manual user interventions to achieve a satisfactory solution for all parties involved in multi-party privacy conflicts. These Techniques are a stepping stone towards more automated resolution of conflicts in multi-party privacy management for Social Media. Also we recommend friends to activate user based on his/her interest. As future work, the proposed system, we plan to continue researching on what makes users concede or not when solving conflicts in this domain. In particular, we are also interested in exploring if there are other factors that could also play a role in this, like for instance if concessions may be influenced by previous negotiations with the same negotiating users or the relationships between negotiators themselves.

## Future Scope

In our future work, we will work on conflict detection and resolution auto- matically i.e. without admin module. Also we will work on conflict detection and resolution on videos.

# REFERENCES

* + 1. K. Thomas, C. Grier, and D. M. Nicol, Unfriendly: Multi-party privacy risks in social networks, in Proc. 10th Int. Symp. Privacy Enhancing Technol., 2010, pp. 236252.
    2. A. Lampinen, V. Lehtinen, A. Lehmuskallio, and S. Tamminen, Were in it together: Interpersonal management of disclosure in social network services, in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2011, pp. 32173226.
    3. P. Wisniewski, H. Lipford, and D. Wilson, Fighting for my space: Cop- ing mechanisms for SNS boundary regulation, in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2012, pp. 609618.
    4. A. Besmer and H. Richter Lipford, Moving beyond untagging: Photo privacy in a tagged world, in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2010, pp. 15631572.
    5. J. M. Such, A. Espinosa, and A. Garc Fornes, A survey of privacy in multi-agent systems, Knowl. Eng. Rev., vol. 29, no. 03, pp. 314344, 2014.
    6. R. L. Fogues, J. M. Such, A. Espinosa, and A. Garcia-Fornes, Open challenges in relationship-based privacy mechanisms for social network services, Int. J. Human-Comput. Interaction, vol. 31, no. 5, pp. 350370, 2015.
    7. R. Wishart, D. Corapi, S. Marinovic, and M. Sloman, Collaborative privacy policy authoring in a social networking context, in Proc. IEEE Int. Symp. Policies Distrib. Syst. Netw., 2010, pp. 18.
    8. H. Hu, G.-J. Ahn, and J. Jorgensen, Detecting and resolving privacy conflicts for collaborative data sharing in online social networks, in Proc. 27th Annu. Comput. Security Appl. Conf., 2011, pp. 103112. [Online]. Available: [http://doi.acm.org/10.1145/2076732.2076747.](http://doi.acm.org/10.1145/2076732.2076747)
    9. H. Hu, G. Ahn, and J. Jorgensen, Multiparty access control for online social networks: Model and mechanisms, IEEE Trans. Knowl. Data Eng., vol. 25, no. 7, pp. 16141627, Jul. 2013.
    10. P. Fong, Relationship-based access control: Protection model and pol- icy language, in Procs. 1st ACM Conf. Data Appl. Security Privacy, 2011, pp. 191202.

SIT Lonavala, Department of Computer Engineering 2018-19 84

* + 1. B. Carminati and E. Ferrari, Collaborative access control in onlinesocial networks, in Proc. 7th Int. Conf. Collaborative Comput.: Netw. Appl. Worksharing, 2011, pp. 231240.
    2. L. Fang and K. LeFevre, Privacy wizards for social networking sites, in Proc. 19th Int. World Wide Web Conf., 2010, pp. 351360.
    3. A. C. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede, A3p: Adap- tive policy prediction for shared images over popular content sharing sites, in Proc. 22nd ACM Conf. Hypertext Hypermedia, 2011, pp. 261270.
    4. E. Gilbert and K. Karahalios, Predicting tie strength with social media, in Proc. Conf. Human Factors Comput. Syst., 2009, pp. 211220.
    5. A. Acquisti and R. Gross, Imagined communities: Awareness, infor- mation sharing, and privacy on the Facebook, in Proc. 6th Int. Conf. Privacy Enhancing Technol., 2006, pp. 3658.

# ANNEXURE A

**LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHMIC DESIGN**

To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix.

•

Refer [**?**] for IDEA Matrix and Knowledge canvas model. Case studies are given in this book. IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

|  |  |  |  |
| --- | --- | --- | --- |
| I | D | E | A |
| Increase | Drive | Educate | Accelerate |
| Improve | Deliver | Evaluate | Associate |
| Ignore | Decrease | Eliminate | Avoid |

Table A.1: IDEA Matrix

Project problem statement feasibility assessment using NP-Hard, NP- Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.

•

* + - * input x,output y, y=f(x)

SIT Lonavala,Department of Computer Engineering 2017-18

# ANNEXURE B

**LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN**

SIT Lonavala,Department of Computer Engineering 2017-18

It should include assignments such as

Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading in functions (if any), and functional relations and any other dependencies

•

(as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading

Use of above to draw functional dependency graphs and relevant Soft- ware modeling methods, techniques including UML diagrams or other necessities using appropriate tools.

•

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selec- tion and appropriate use of testing tools, testing of UML diagram’s reliability. Write also test cases [Black box testing] for each identified functions. You can use Mathematica or equivalent open source tool for generating test data.

•

Additional assignments by the guide. If project type as Entreprenaur, Refer [**?**],[**?**],[**?**], [**?**]

•

SIT Lonavala,Department of Computer Engineering 2017-18

# ANNEXURE C PROJECT PLANNER

SIT Lonavala,Department of Computer Engineering 2017-18

Using planner or alike project management tool.

SIT Lonavala,Department of Computer Engineering 2017-18

# ANNEXURE D

**REVIEWERS COMMENTS OF PAPER SUBMITTED**

SIT Lonavala,Department of Computer Engineering 2017-18

(At-least one technical paper must be submitted in Term-I on the project de- sign in the conferences/workshops in IITs, Central Universities or UoP Con- ferences or equivalent International Conferences Sponsored by IEEE/ACM)

1. Paper Title:
2. Name of the Conference/Journal where paper submitted :
3. Paper accepted/rejected :
4. Review comments by reviewer :
5. Corrective actions if any :

SIT Lonavala,Department of Computer Engineering 2017-18

# ANNEXURE E PLAGIARISM REPORT

# SIT Lonavala,Department of Computer Engineering 2017-18

Plagiarism report

# ANNEXURE F

**TERM-II PROJECT LABORATORY ASSIGNMENTS**

SIT Lonavala,Department of Computer Engineering 2017-18

1. Review of design and necessary corrective actions taking into consid- eration the feedback report of Term I assessment, and other competi- tions/conferences participated like IIT, Central Universities, University Conferences or equivalent centers of excellence etc.
2. Project workstation selection, installations along with setup and instal- lation report preparations.
3. Programming of the project functions, interfaces and GUI (if any) as per 1 st Term term-work submission using corrective actions recom- mended in Term-I assessment of Term-work.
4. Test tool selection and testing of various test cases for the project per- formed and generate various testing result charts, graphs etc. including reliability testing.

#### Additional assignments for the Entrepreneurship Project:

1. Installations and Reliability Testing Reports at the client end.

SIT Lonavala,Department of Computer Engineering 2017-18

# ANNEXURE G

**INFORMATION OF PROJECT GROUP MEMBERS**

one page for each student .

studentphoto.png

1. Name :
2. Date of Birth :
3. Gender :
4. Permanent Address :
5. E-Mail :
6. Mobile/Contact No. :
7. Placement Details :
8. Paper Published :