A Project Report on

MONITORING USER BEHAVIOUR IN SOCIAL MEDIA

Submitted in partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

Submitted by:

G.GIRIDHAR REDDY	**(16091A0530)**
C.DIVAKAR REDDY	**(16091A0525)**
D.KARISHMA	**(16091A0548)**
C.GNANA SAI	**(16091A0532)**

Under the Guidance of

Mr. K.E.NARESH KUMAR M.Tech.,(Ph.D.)
Associate Professor, Dept. of CSE.

(ESTD-1995)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)
Affiliated to JNTUA Anantapuramu, APPROVED BY A.I.C.T.E., NEW DELHI,
ACCREDITED BY NAAC of UGC, NEWDELHI with "A+" grade,
ACCREDITED BY N.B.A, NEWDELHI,
NANDYAL-518501, (Estd-1995)

(2019-2020)

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

Rajeev Gandhi Memorial College of Engineering &Technology
(AUTONOMOUS)
(Affiliated to JNTU Anantapuramu)
APPROVED BY A.I.C.T.E., NEW DELHI, ACCREDITED BY N.B.A, NEWDELHI,
NANDYAL-518501

(ESTD – 1995)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that **G.GIRIDHAR REDDY** (16091A0530), **C.DIVAKARREDDY** (16091A0525), **D.KARISHMA**(16091A0548) and **C.GNANASAI** (16091A0532) of final year B.Tech, C.S.E, have carried out the major project work entitled "**MONITORING USER BEHAVIOUR IN SOCIAL MEDIA**" under the supervision and guidance of **Mr. K.E. NARESH KUMAR** M.Tech.,(Ph.D.) Associate Professor, CSE Department, in partial fulfillment of the requirements for the award of Degree of **Bachelor of Technology** in **Computer Science & Engineering** from **Rajeev Gandhi Memorial College of Engineering & Technology (Autonomous)**, Nandyal is a bonafide record of the work done by them during 2019-2020.

Project Guide [illegible]

Mr. K.E. NARESH KUMAR M.Tech., (Ph.D.)
Associate Professor, Dept. of CSE

Head of the Department [illegible]

Dr. K. SUBBA REDDY Ph.D.
Professor, Dept. of CSE

Place: Nandyal
Date:

External Examiner

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

Candidate's Declaration

We hereby declare that the project work entitled **"MONITORING USER BEHAVIOUR IN SOCIAL MEDIA"** was carried out and written by us under the guidance of K. E. Naresh Kumar M.Tech.,(Ph.D.), Associate Professor, Dept. of Computer Science & Engineering, **R.G.M. College of Engineering & Technology**, and this project work is submitted in the partial fulfillment of the requirements for the award of the degree of "Bachelor of Technology" in **Computer Science & Engineering**. We have not submitted the matter presented in this report anywhere for the award of any other Degree.

G. Giridhar Reddy	**(16091A0530)**
C. Divakar Reddy	**(16091A0525)**
D. Karishma	**(16091A0548)**
C. Gnana Sai	**(16091A0532)**

Dept. of CSE,
RGMCET.

Name of Guide:

Dr. K. E. Naresh Kumar M.Tech.,(Ph.D.)
Associate Professor,
Dept. of CSE.

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

ACKNOWLEDGEMENT

At the outset, we would be failing in our duties if we do not express our sincere gratitude to our guide & supervisor, internal guide **Mr. K. E. Naresh Kumar, Associate Professor** for the guidance and assistance to us, which contribute to successful completion of this project.

Our special thanks to **Dr. K. Subba Reddy, Head of the Department (CSE), Rajeev Gandhi Memorial College of Engineering & Technology,** for providing all the facilities and guidelines, required for our academic pursuit.

Our special thanks to **Dr. T. JAYACHANDRA PRASAD, Principal, Rajeev Gandhi Memorial College of Engineering & Technology,** for providing all the necessary facilities, required for our academic pursuit.

We would like to express our sincere and grateful thanks to the management of **Rajeev Gandhi Memorial College of Engineering & Technology,** under the leadership of **Dr. M. SANTHIRAMUDU, Chairman for** providing us an opportunity to fulfill our Aspirations.

BY

G.Giridhar Reddy	**(16091A0530)**
C.Divakar Reddy	**(16091A0525)**
D.Karishma	**(16091A0548)**
C.Gnana Sai	**(16091A0532)**

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

ABSTRACT

In current trend many social networking sites created and providing services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provide major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produces per day. According to research survey many more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc. Crawling the user information from the user micro blogs is also less practical. Anonymous Users can easily manipulate the public profile information.

So in this project we are implementing the web based application which will find the anonymous users according to the user behavior. We calculate the user behavior according to the chat statements of the user which he/she do with others. By the taking the advantages Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

The scope of our work will improve the security in online social network platform. We are pointing the loopholes of the current system and proposing new architecture to fulfill the loopholes. The scope of our work is to monitor the user behavior using chat history and provide some security features to the user.

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

CONTENTS

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

LIST OF ABBREVIATIONS

GUI-Graphical User Interface

ML-Machine Learning

EDM-Entity Data Model

EDA-Exploratory Data Analysis

OOP-Object Oriented Programming

UML-Unified Modelling Language

ELM-Extreme Learning Machine

14

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501. Kurnool (Dist), A.P.

LIST OF FIGURES

LIST OF TABLES

[illegible]

Dr. SUBBA REDDY KUNAM
BE, M.Tech, Ph.D, MIEEE, MISTE, MCSI, FIETE, FIE(I)
Professor & Head of the Department of CSE
RGM College of Engg. & Tech., (Autonomous)
NANDYAL-518 501, Kurnool (Dist), A.P.

CHAPTER-1

INTRODUCTION

1.1 Introduction:

This project is aimed at developing the web based application called “**Monitoring User Behaviour in Social Media**”. In current trend many social networking sites created and providing services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provide major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produces per day. According to research survey many more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc. Crawling the user information from the user micro blogs is also less practical. Anonymous Users can easily manipulate the public profile information.

Social media are interactive computer-mediated technologies that facilitate the creation or sharing of information, ideas, career interests and other forms of expression via virtual communities and networks. The variety of stand-alone and built-in social media services currently available introduces challenges of definition; however, there are some common features:

- Social media are interactive Web 2.0 Internet-based applications.
- User-generated content such as text posts or comments, digital photos or videos, and data generated through all online interactions, is the lifeblood of social media.
- Users create service-specific profiles and identities for the website or app that are designed and maintained by the social media organization.
- Social media facilitate the development of online social networks by connecting a user's profile with those of other individuals or groups.

Users usually access social media services via web-based apps technologies on desktops and laptops, or download services that offer social media functionality to their mobile devices (e.g., smartphones and tablets). As users engage with these electronic services, they create highly interactive platforms through which individuals, communities, and organizations can share, co-create, discuss, participate and modify user-generated content or self-curated content posted online.

In this project we are implementing the web based application which will find the anonymous users according to the user behavior. We calculate the user behavior according to the chat statements of the user which he/she do with others. By the taking the advantages Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

1.2 Objectives:

- To monitor the user behavior in social media.
- Classifying the anonymous users in the social media.
- Blocking those users.
- Calculate the user behavior according to the chat statements of the user which he/she do with others.

1.3 Scope:

- The scope of our work will improve the security in online social network platform. We are pointing the loopholes of the current system and proposing new architecture to fulfill the loopholes.
- We survey on the Machine learning concept which will help to track the anonymous user in OSN.
- The scope of our work is to monitor the user behavior using chat history and provide some security features to the user.
- Many studies focus on this topic based on profile attributes, like names, profile picture, location, birth days etc. But we are focus on the user’s generate data which user will generate more data compare with the profile attributes.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction:

In current trend many social networking sites created and providing services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provide major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produces per day. According to research survey many more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc. But in the current social sites not focus on services like tracking the user behavior of anonymous behavior. In current system, social network sites need to focus the user microblogs and need to capture the user behavior whether his/she anonymous user or not.

Few surveys' providing concepts to tracking the attackers like using profile matching techniques and network based techniques etc. But in real-time to apply those concepts in social network is less practical. Crawling the user information from the user micro blogs is also less practical. Anonymous Users can easily manipulate the public profile information.

2.2 Related Work:

2.2.1 Profile-Based User Identification:

Several literature surveys studies that addressing the tracking the anonymous online social users have concentrate on the public profile attributes. These may includes screen-names, gender, date of birth, location, city, and profile picture etc. A screen name is the publically required profile feature in almost all SMNs. It has been broadly investigated as an approach to perceive online social users crosswise over various SMNs. Perito et al. determined the comparability of screen names and distinguished online social users utilizing parallel classifiers. Correspondingly, Liu et al. coordinated online social users in an unaided methodology utilizing screen names. Zafarani and Liu proposed a technique to outline crosswise over various SMN stages, experimentally approving a few theories.

Over this work, they further built up a client mapping strategy by demonstrating client conduct on screen names. Among open profile traits, the profile picture is another component that has gotten significant examination. Acquisti et al. tended to the client distinguishing proof assignment with a face acknowledgment calculation. Albeit both screen name and profile picture can distinguish online social users , they can't be applied to huge SMNs. This is on the grounds that a few online social users may have a similar screen name and profile pictures. For instance, numerous online social users have the screen name "John Smith" on Facebook.

Clearly, utilizing a blend of profile highlights can bring about better client recognizable proof. Iofciu et al. proposed a methodology by estimating the separation between client profiles. Motoyama and Varghese accumulated traits (training, occupation, and so forth.) as sets of words and coordinated online social users by ascertaining the likeness of online social users . Goga connected records having a place with a similar individual character, in light of on the profile data.

Without a doubt, open profile traits give amazing data to client distinguishing proof. In any case, a few traits are copied in huge scale SMNs, and are effectively imitated. Hence, absolutely profile-based plans have constraints when they are applied to enormous scale SMNs.

2.2.2 Network Structure-Based User Identification:

Network structure-based studies on user identification across multiple SMNs are used to recognize identical users solely by user network structures and seed, or priori, identified users. As shown above, network-based user identification poses several major challenges, with few studies to build on. Network structure-based (NS) user identification is a hard nut to crack, and can be used to identify only a portion of identical users. NS, the first network structure-based user recognition algorithm across SMNs, can carry out user recognition tasks by using only the network structure, and identified 30.8 percent identical users in a ground-truth dataset .

Suppose that there are two SMNs: $SMN_A$and SMN_B. NS first calculates a set of mapping scores for each single, unmapped user entity (UE) in $SMN_A$to every unmapped user entity in SMN_B. Then an eccentricity is applied to determine whether or not a user in

SMN_B can be matched. Only if the eccentricity is larger than a threshold would a user match be accepted. In addition, NS requires a reverse match to confirm the user match, which is costly in experiments. In this analysis of cross-platform SMNs, we deeply mined friend relationships and network structures. In the real world, people tend to have mostly the same friends in different SMNs, or the friend cycle is highly individual. The more matches in two unmapped users' known friends, the higher the probability that they belong to the same individual in the real world.

2.2.3 Content-Based User Identification:

Content-Based User Identification arrangements endeavor to perceive online social users dependent on the occasions and areas that online social users post content, just as the composing style of the microblog. Zheng proposed a system for creation ID utilizing the composing style of online messages and arrangement strategies. Proposed connecting online social users crosswise over various SMNs by abusing the composing style of creators. Multi-Network Anchoring (MNA) to delineate. They determined the consolidated similitude's of user's social, spatial, transient and content data in various SMNs, and analyzed a stable coordinating issue between two arrangements of user accounts.

Zheng misused the geo-area appended to online social users posts, the timestamp of posts, and online social users composing style to address client recognizable proof assignments. Geo-area seems to have intense highlights for client acknowledgment. Be that as it may, this data is regularly meager in SMNs, since just a little part of online social users are eager to post their areas. Despite the fact that composing style arrangements perform well in situations including long substance, these systems are not material to SMNs, for example, Twitter and Sina Microblog, in which short sentences are in all probability posted.

2.3 Problem Statement:

In current trend many social networking sites created and providing services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provide major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produces per day. According to research survey many more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc . But in the current social sites not focus on services like tracking the user behavior of anonymous behavior. In current system, social network sites need to focus the user microblogs and need to capture the user behavior whether his/she anonymous user or not.

Few surveys' providing concepts to tracking the attackers like using profile matching techniques and network based techniques etc. But in real-time to apply those concepts in social network is less practical. Crawling the user information from the user micro blogs is also less practical. Anonymous Users can easily manipulate the public profile information.

Disadvantages:

- Based existing surveys attacker easily morph the data.
- Very less practical, we can't find the attacker using small tiny blogs
- Based on the network based models we should effort heavy data for detecting

2.4 Proposed work:

In current trend many social networking sites created and providing services of communications, multi-media services, e-commerce etc immensely. Lot of anonymous users accounts are creating very rapidly. We need to focus for the tracking the anonymous users. In our proposed system we are implementing the web based application which will find the anonymous users according to the user behavior. We calculate the user behavior according to the chat statements of the user which he/she do with others. By the taking the advantages Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

2.5 Advantages:

- We are depending on the chat history instead of the profile attributes or any other media attributes, so we can conclude with minimum amount of the computation.
- Due to machine learning classifications we can get accurate results

2.6 System requirements:

2.6.1 User Interfaces:

The system is provided with keyboard shortcuts with a facility to use most to trigger the required actions. They act as shortcuts and provide an easy navigation with in their software.Appropriate error handling is done using exceptions in order to isolate abnormal results or conditions .These messages are popped up to the user in the form of dynamic html or alert.

2.6.2 Non-functional requirements:

- **Usability:**

  Prioritize the important functions of the system based on usage patterns.Frequently used functions should be tested for usability, as should complex and critical functions. Be sure to create a requirement for this.

- **Reliability:**

  Reliability defines the trust in the system that is developed after using it for a period of time. It defines the likeability of the software to work without failure for a given time period. The number of bugs in the code, hardware failures, and problems can reduce the reliability of the software. Your goal should be a long MTBF (mean time between failures). It is defined as the average period of time the system runs before failing. Create a requirement that data created in the system will be retained for a number of years without the data being changed by the system. It's a good idea to also include requirements that make it easier to monitor system performance.

- **Performance:**

  What should system response times be, as measured from any point, under what circumstances are there specific peak times when the load on the system will be unusually high Think of stress periods, for example, at the end of the month or in conjunction with payroll disbursement.

- **Supportability:**

  The system needs to be cost-effective to maintain. Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documentation, e.g. which test cases and test plans will accompany the system.

2.6.3 Hardware requirements:

• Hardware	:Pentium
• Speed	:1.1 GHz
• RAM	:1GB
• Hard Disk	:20 GB

2.6.4 Software requirements:

• Operating System	: Windows Family
• Technology	: Java, J2EE and Python 3.7
• Web Technologies	: Html, JavaScript, CSS
• Web Server	: Apache Tomcat 8.0
• Database	: My SQL 5.5 or Higher
• UML's	: StarUml

CHAPTER-3
SYSTEM DESIGN

3.1 Modules:

3.1.1 User module:

The user can login in to the system, they can upload images in their profile, and also send text messages in API provided to them, then can accept or decline user requests. They can view other user's profile in the social media. Along with they can search for other users and can send requests to them.

3.1.2 Admin module:

The admin can monitor the user activities and can add bulling words, he/she can detect the malicious users and with the provided API they will be automatically blocked.He can view the users and their friends in social media.

3.2 System Architecture:

System architecture conveys the informational content of the elements consisting of a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that an architecture description may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people.

In this project we are implementing the web based application which will find the anonymous users according to the user behavior. We calculate the user behavior according to the chat statements of the user which he/she do with others. By the taking the advantages Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.
In the following architecture fig5.2.1 explain the implementation flow.

- Expected outcomes will track the anonymous users based chat.
- User will get the warning notification to for the first identification
- User account will block for the anonymous behavior after first notification.

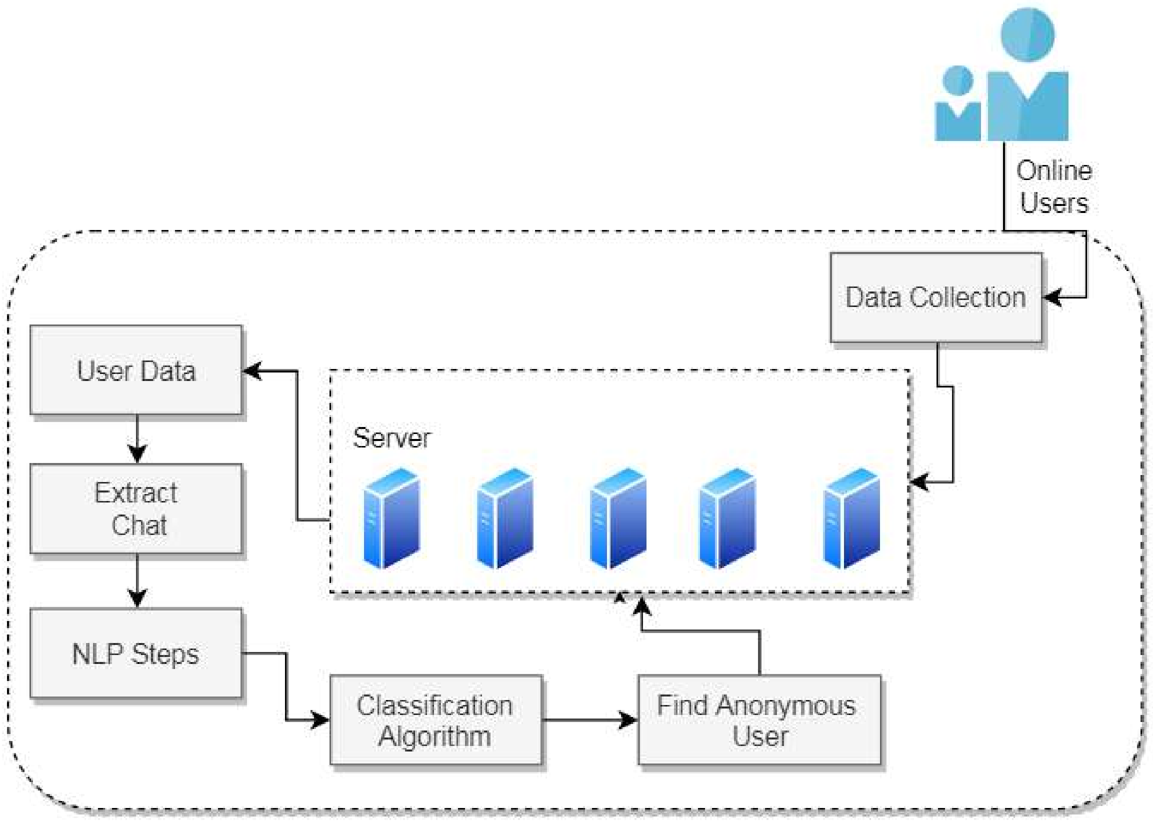


Figure 3.2: System Architecture

3.2.1 Software Development Life Cycle (SDLC):

SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time. SDLC includes a detailed plan for how to develop, alter, maintain, and replace a software system.Popular SDLC models include the waterfall model, spiral model, and Agile model. These are the following steps of SDLC

- Project Requisites Accumulating and Analysis
- Application System Design
- Practical Implementation
- Manual Testing of My Application
- Application Deployment of System
- Maintenance of the Project

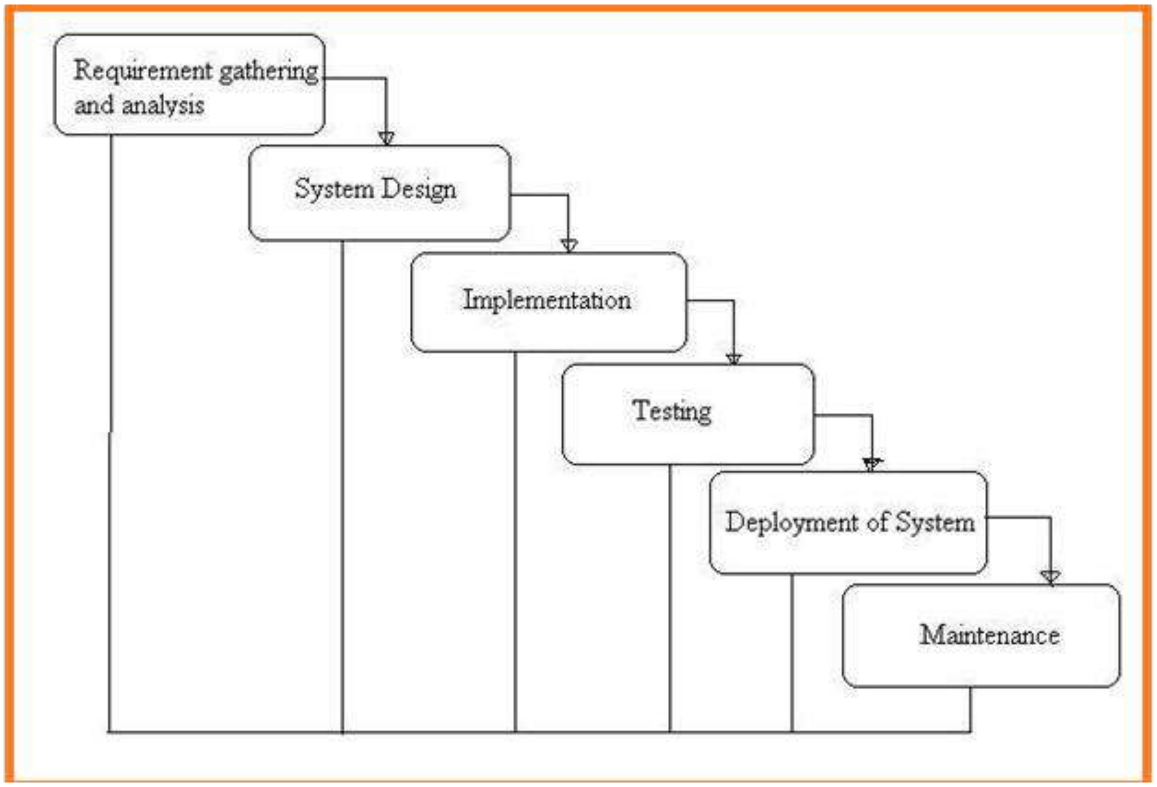


Figure 3.2.1: SDLC Diagram

3.2.2 Requisites Accumulating and Analysis:

It's the first and foremost stage of the any project as our is a an academic leave for requisites amassing we followed of IEEE Journals and Amassed so many IEEE Relegated papers and final culled a Paper designated "Individual web revisitation by setting and substance importance inputand for analysis stage we took referees from the paper and did literature survey of some papers and amassed all the Requisites of the project in this stage

3.2.3 System Design:

In System Design has divided into three types like GUI Designing, UML Designing with avails in development of project in facile way with different actor and its utilizer case by utilizer case diagram, flow of the project utilizing sequence, Class diagram gives information about different class in the project with methods that have to be utilized in the project if comes to our project our UML Will utilizable in this way The third and post import for the project in system design is Data base design where we endeavor to design data base predicated on the number of modules in our project .

3.2.4 Implementation:

The Implementation is Phase where we endeavor to give the practical output of the work done in designing stage and most of Coding in Business logic lay coms into action in this stage its main and crucial part of the project

3.2.5 Testing:

- Unit Testing**:**

  It is done by the developer itself in every stage of the project and fine-tuning the bug and module predicated additionally done by the developer only here we are going to solve all the runtime errors.

- Manual Testing**:**

  As our Project is academic Leave we can do any automatic testing so we follow manual testing by endeavor and error methods.

3.2.6 Deployment of System:

Once the project is complete we will come to deployment of client system, we did deployment our laptop only with all need Software's with having Windows OS

3.2.7 Maintenance:

The Maintenance of our Project is one time process only.

3.3 UML DIAGRAMS:

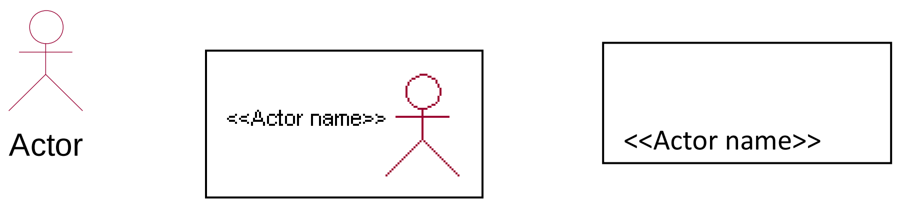
The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

3.3.1 Global Use Case Diagrams:

Actor:

Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

Graphical representation:



An actor is someone or something that:

Interacts with or uses the system.

- Provides input to and receives information from the system.
- Is external to the system and has no control over the use cases.

Actors are discovered by examining:

- Who directly uses the system
- Who is responsible for maintaining the system
- External hardware used by the system.
- Other systems that need to interact with the system.

Questions to identify actors:

- Who is using the system Or, who is affected by the system Or, which groups need help from the system to perform a task
- Who affects the system Or, which user groups are needed by the system to perform its functions These functions can be both main functions and secondary functions such as administration.
- Which external hardware or systems (if any) use the system to perform tasks.
- What problems does this application solve (that is, for whom).
- And, finally, how do users use the system (use case) What are they doing with the system.

The actors identified in this system are:

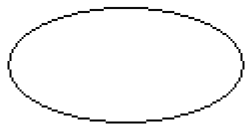
- System Administrator
- Customer
- Customer Care

Identification of usecases:

Usecase:

A use case can be described as a specific way of using the system from a user's (actor's) perspective.

Graphical representation:



A more detailed description might characterize a use case as:

- Pattern of behavior the system exhibits
- A sequence of related transactions performed by an actor and the system
- Delivering something of value to the actor

Use cases provide a means to:

- capture system requirements
- communicate with the end users and domain experts
- test the system

  Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

  Guide lines for identifying use cases:

- For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
- Name the use cases.
- Describe the use cases briefly by applying terms with which the user is familiar.

  This makes the description less ambiguous

Questions to identify use cases:

- What are the tasks of each actor
- Will any actor create, store, change, remove or read information in the system
- What use case will store, change, remove or read this information
- Will any actor need to inform the system about sudden external changes
- Does any actor need to inform about certain occurrences in the system
- What usecases will support and maintains the system

Flow of Events:

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favorite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

- When and how the use case starts and ends
- Use case/actor interactions
- Data needed by the use case
- Normal sequence of events for the use case
- Alternate or exceptional flows

Construction of Usecase diagrams:

Use-case diagrams graphically depict system behavior (use cases). These diagrams present a high level view of how the system is used as viewed from an outsider's (actor's) perspective. A use-case diagram may depict all or some of the use cases of a system.

A use-case diagram can contain:

- actors ("things" outside the system)
- use cases (system boundaries identifying what the system should do)
- Interactions or relationships between actors and use cases in the system including the associations, dependencies, and generalizations.

Relationships in use cases:

- **Communication:**

  The communication relationship of an actor in a usecase is shown by connecting the actor symbol to the usecase symbol with a solid path. The actor is said to communicate with the usecase.

- **Uses:**

  A Uses relationship between the usecases is shown by generalization arrow from the usecase.

- **Extends:**

The extend relationship is used when we have one usecase that is similar to another usecase but does a bit more. In essence it is like subclass.

3.3.2 Sequence diagrams:

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

There are two main differences between sequence and collaboration diagrams: sequence diagrams show time-based object interaction while collaboration diagrams show how objects associate with each other. A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.

- **Object:**

  An object has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class. Each object in a diagram indicates some instance of a class. An object that is not named is referred to as a class instance.

  The object icon is similar to a class icon except that the name is underlined:

  An object's concurrency is defined by the concurrency of its class.

- **Message:**

  A message is the communication carried between two objects that trigger an event. A message carries information from the source focus of control to the destination focus of control. The synchronization of a message can be modified through the message specification. Synchronization means a message where the sending object pauses to wait for results.

- **Link:**

  A link should exist between two objects, including class utilities, only if there is a relationship between their corresponding classes. The existence of a relationship between two classes symbolizes a path of communication between instances of the classes: one object may send messages to another. The link is depicted as a straight line between objects or objects and class instances in a collaboration diagram. If an object links to itself, use the loop version of the icon.

3.3.3 Class diagram:

A class is a set of objects that share a common structure and common behavior (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items.

There are 4 approaches for identifying classes:

Noun Phrase Approach:

The guidelines for identifying the classes:

- Look for nouns and noun phrases in the usecases.
- Some classes are implicit or taken from general knowledge.
- All classes must make sense in the application domain; Avoid computer implementation classes – defer them to the design stage.
- Carefully choose and define the class names After identifying the classes we have to eliminate the following types of classes:
- Adjective classes.

Common class pattern approach:

The following are the patterns for finding the candidate classes:

- Concept class.
- Events class.
- Organization class
- Peoples class
- Places class
- Tangible things and devices class

Use case driven approach:

We have to draw the sequence diagram or collaboration diagram. If there is need for some classes to represent some functionality then add new classes which perform those functionalities.

CRC approach:

The process consists of the following steps:

- Identify classes' responsibilities (and identify the classes)
- Assign the responsibilities

- Identify the collaborators.

Identification of responsibilities of each class:

The questions that should be answered to identify the attributes and methods of a class respectively are:

- What information about an object should we keep track of
- What services must a class provide

Identification of relationships among the classes:

Three types of relationships among the objects are:

Association: How objects are associated

Super-sub structure: How are objects organized into super classes and sub classes

Aggregation: What is the composition of the complex classes

Association:

The questions that will help us to identify the associations are:

- Is the class capable of fulfilling the required task by itself
- If not, what does it need
- From what other classes can it acquire what it needs

Guidelines for identifying the tentative associations:

- A dependency between two or more classes may be an association. Association often corresponds to a verb or prepositional phrase.
- A reference from one class to another is an association. Some associations are implicit or taken from general knowledge.

Some common association patterns are:

Location association like part of, next to, contained in…..

Communication association like talk to, order to ……

We have to eliminate the unnecessary association like implementation associations, ternary or n-ary associations and derived associations.

Super-sub class relationships:

Super-sub class hierarchy is a relationship between classes where one class is the parent class of another class (derived class).This is based on inheritance.

Guidelines for identifying the super-sub relationship, a generalization are

- **Top-down*:***

  Look for noun phrases composed of various adjectives in a class name. Avoid excessive refinement. Specialize only when the sub classes have significant behavior.

- **Bottom-up*:***

  Look for classes with similar attributes or methods. Group them by moving the common attributes and methods to an abstract class. You may have to alter the definitions a bit.

- **Reusability:**

  Move the attributes and methods as high as possible in the hierarchy.

- **Multiple inheritances*:***

  Avoid excessive use of multiple inheritances. One way of getting benefits of multiple inheritances is to inherit from the most appropriate class and add an object of another class as an attribute.

Aggregation or a-part-of relationship:

It represents the situation where a class consists of several component classes.

3.3.4 Usecase diagram:

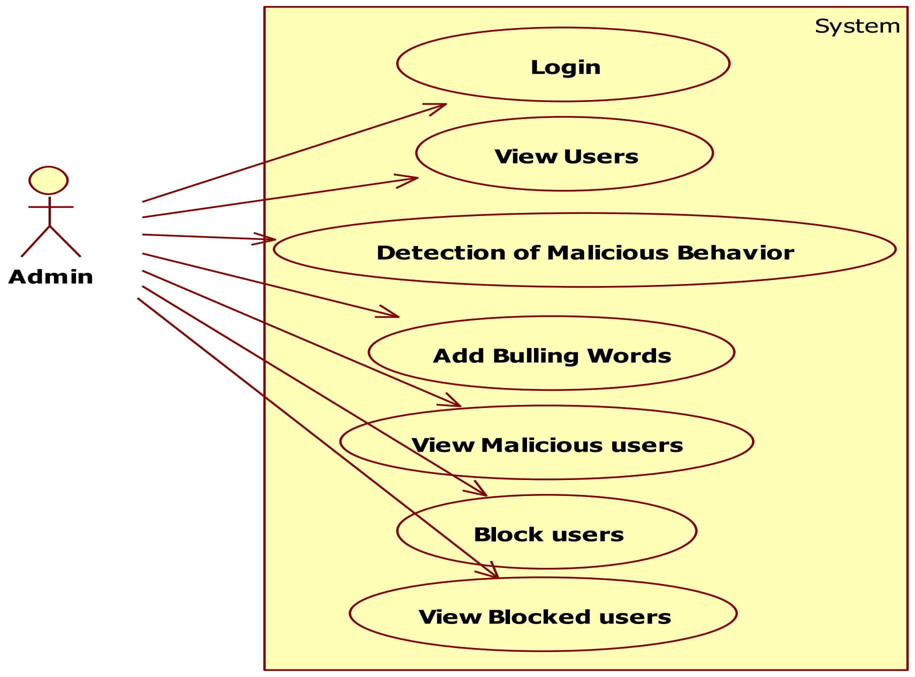


Figure 3.3.1: Usecase Diagram of Admin

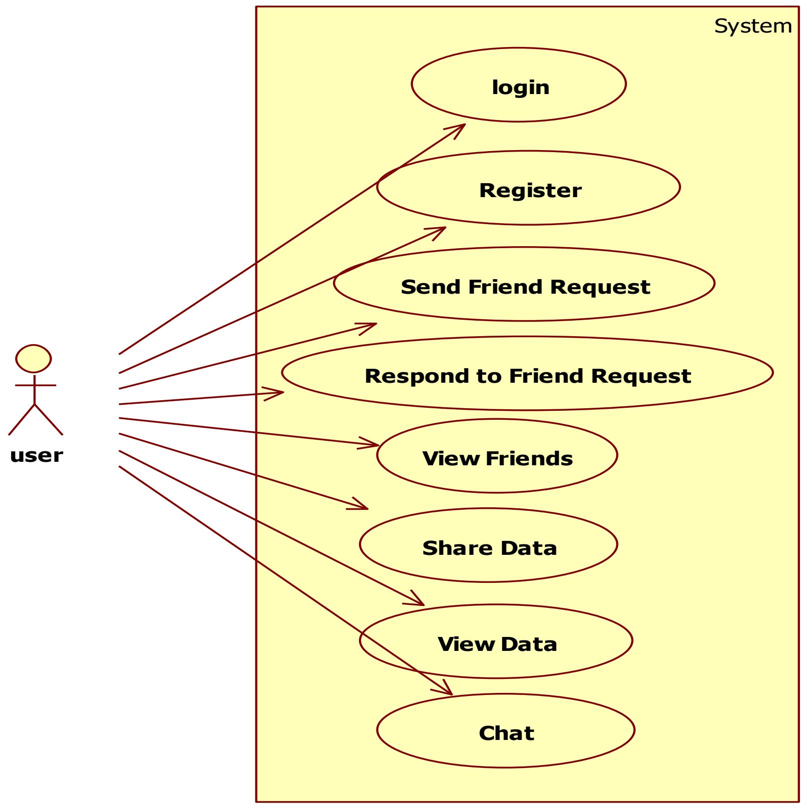


Figure 3.3.2: Use case diagaram of User

Description:

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

3.3.5 Sequence Diagram:

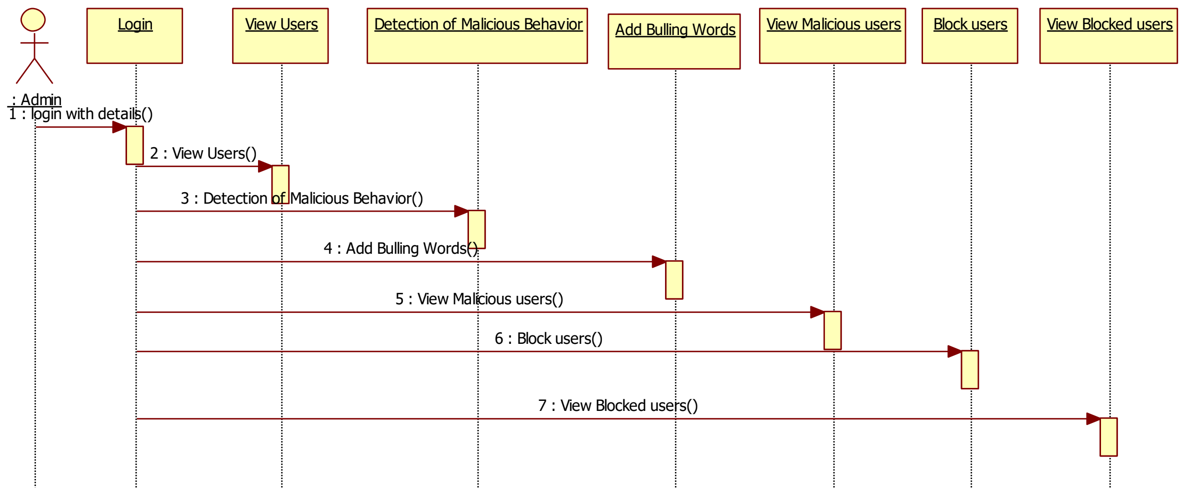


Figure 3.3.3: Sequence diagram Admin

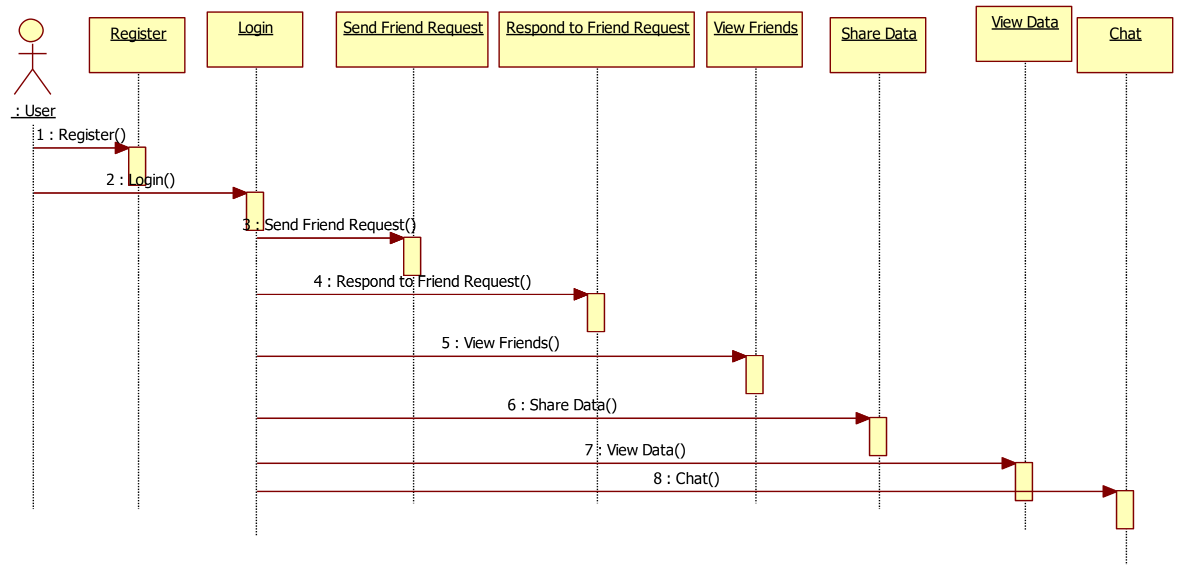


Figure 3.3.4: Sequence Diagram User

Description:

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 development. Sequence diagrams are sometimes called event diagrams or event scenarios.

3.3.6 State Diagram:

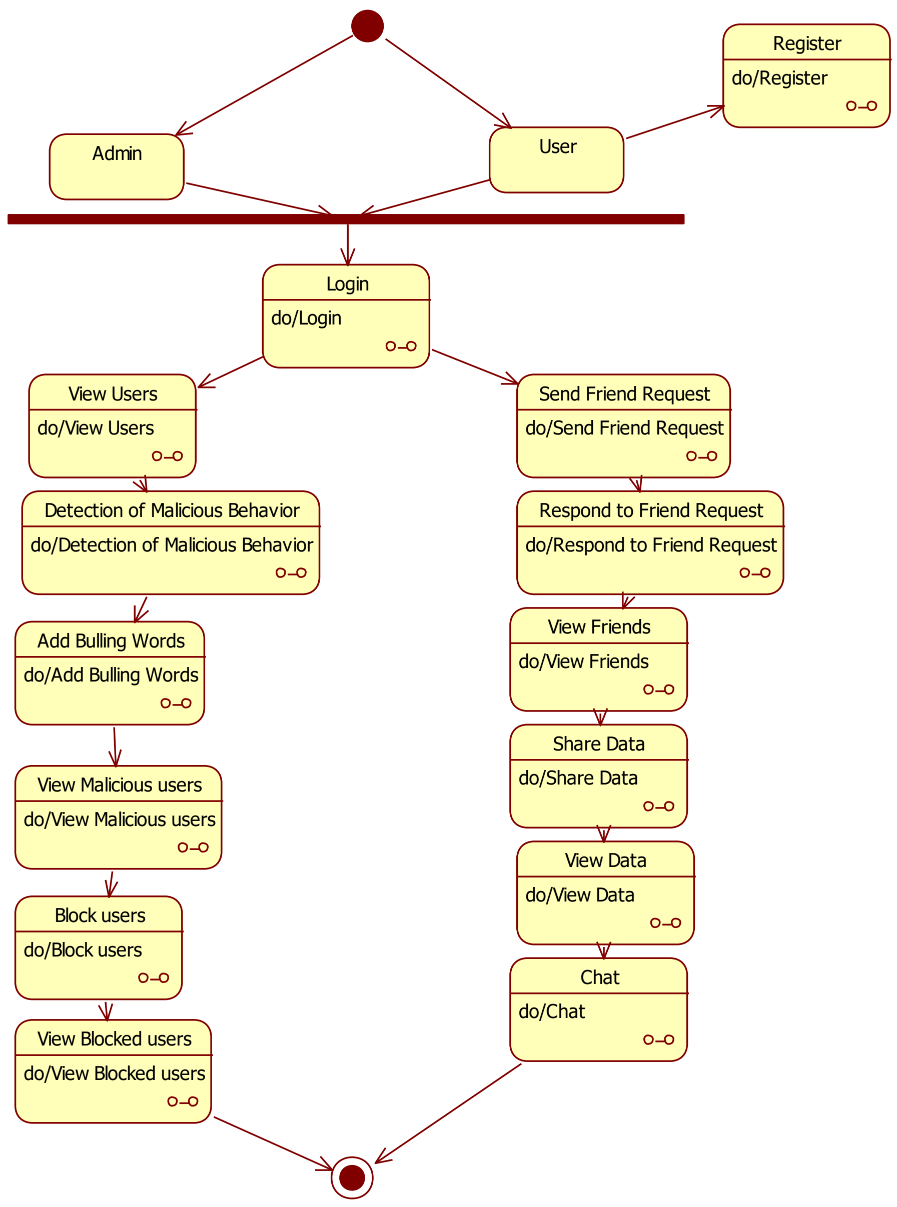


Figure 3.3.5: State Diagram

Description:

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states

3.3.7 Activity Diagram:

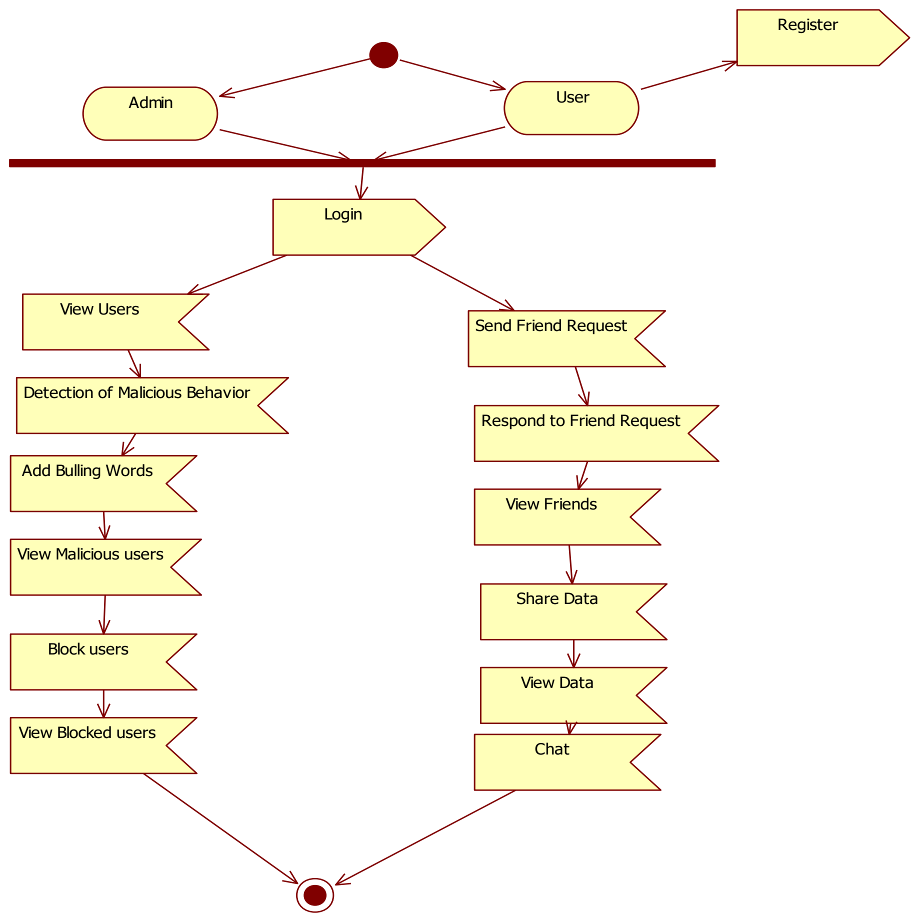


Figure 3.3.6: Activity Diagram

Description:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

3.3.8 Deployment Diagram:

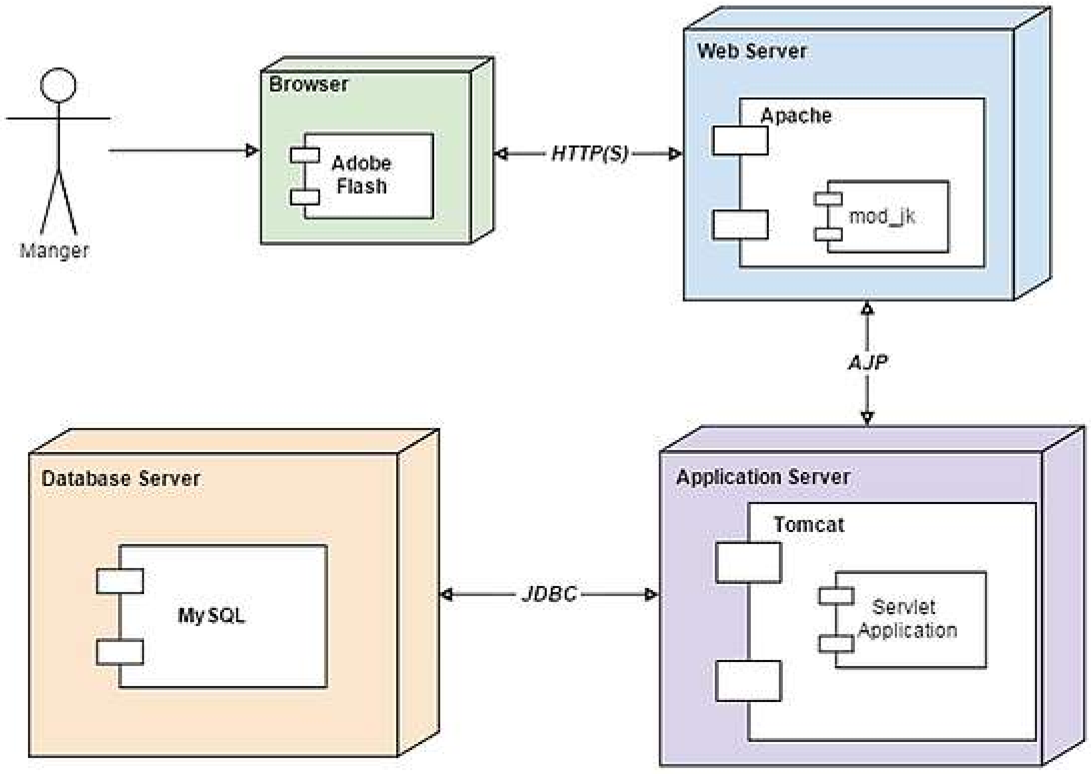


Figure 3.3.7: Deployment Diagram

Description:

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.

CHAPTER-4

SYSTEM IMPLEMENTATION

4.1 Technologies

4.1.1 Naive bayes algorithm:

Naive Bayes classifier is based on Bayes theorem. It has strong independence assumption. It is also known as independent feature model. It assumes the presence or absence of a particular feature of a class is unrelated to the presence or absence of any other feature in the given class. Naïve bayes classifier can be trained in supervised learning setting. It uses the method of maximum similarity. It has been worked in complex real world situation. It requires small amount of training data. It estimates parameters for classification. Only the variance of variable need to be determined for each class not the entire matrix. Naïve bayes is mainly used when the inputs are high. It gives ouput in more sophisticated form. The probability of each input attribute is shown from the predictable state. Machine learning and data mining methods are based on naïve bayes classification.

Bayes theorem:-

$$P(H|X) = \frac{P(X|H)\ P(H)}{P(X)}$$

- Where P(H|X) is posterior probability of H conditioned on X
- P(X|H) is posterior probability of X conditioned on H
- P(H)is prior probability of H P(X) is prior probability of X

4.1.2 Project Software's:

JAVA, Apache Server, MSQL, EDIT ++.In our web Application Development we are using one tier architecture as total applicant will be developed in single system with all the three layers of application development like presentation layer where we use our web technologies to make of GUI of the application like HTML, HTML-5, CSS, JS Etc. and in second layer we have to make our business logic or called as implementation of application where we are using java, J2EE and also we use JDBC to connect from our

Business layer to data base layer and final our data base layer where we develop the Data structure of the application

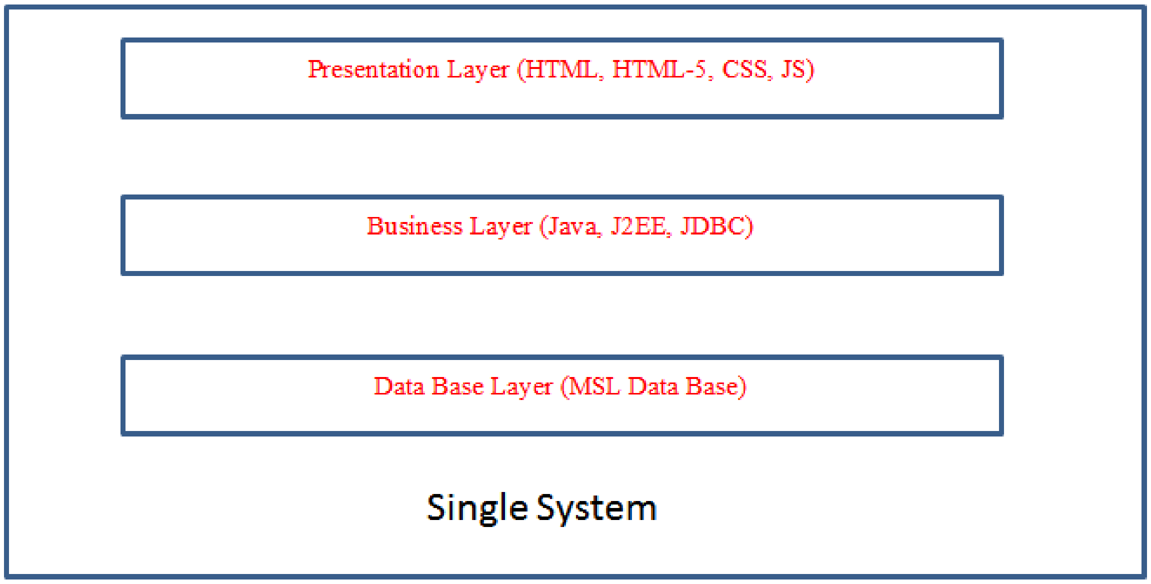

Presentation Layer (HTML, HTML-5, CSS, JS)
Business Layer (Java, J2EE, JDBC)
Data Base Layer (MSL Data Base)
Single System

Figure 4.1.1: Single tire Architecture Project Development

How we used java in our Project Development:

- **Installation and Setup in our system**

  The Software we download form the oracle website as it's an open source as per the software we have installed it in our system and for we have set the system path of java in our OS location We have used the main logic of our algorithm by core java concepts only for web application we have used all JSP concepts and to connect data base we have used JDBC with all this concept we have done the application in Single tire Architecture Project

- **Data Storage in MYSQL**

  We have taken open source software MYSQL from the provide website and run in our system we used for creating our project data base related tables as per project requirement's even for user friendly access of my sql we used Software called SQL Yog where we can do all the operation of mysql by click & use

- **About the role of apace tomcat webserver**

  As our project is a web applicant we need webserver so for that we used again open sour software where our total project source code will be in webapps of the server form that location the application run into web browser where users can see the implementation of the total project

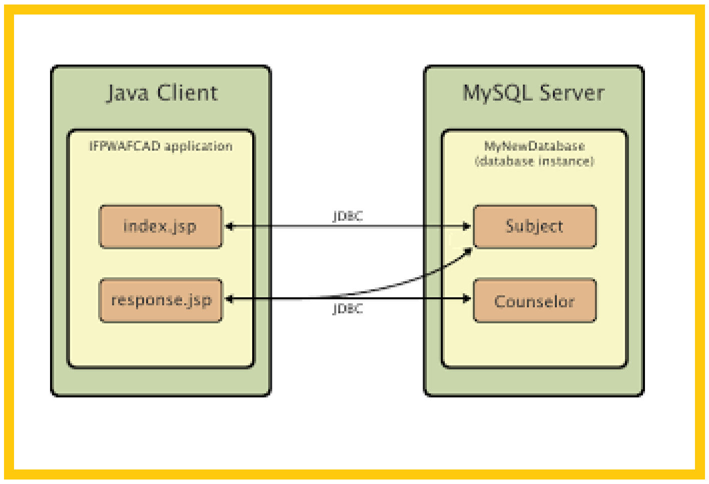


Figure 4.1.2: Application Development Structure

4.1.3 MySQL:

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object- oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

MySQL is the world's most popular open source database software, with over 100million copies of its software downloaded or distributed throughout it's history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production- tested software, proactive monitoring tools, and premium support services are available.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast- growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open- source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database- driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Google.

The most common use for mySQL however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store.

In association with a scripting language such as **PHP** or **Perl** (both offered on our hosting accounts) it is possible to create websites which will interact in real-time with a mySQL database to rapidly display categorised and searchable information to a website user.

How MySQL works:

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.

Features:

MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

MySQL is written in C and C++ and accessible and available across over 20 platforms, including Mac, Windows, Linux and Unix. The RDBMS supports large databases with millions records and supports many data types including signed or unsigned integers 1, 2, 3, 4, and 8 bytes long; FLOAT; DOUBLE; CHAR; VARCHAR; BINARY; VARBINARY; TEXT; BLOB; DATE; TIME; DATETIME; TIMESTAMP; YEAR; SET; ENUM; and Open GIS spatial types.

Fixed- and variable-length string types are also supported.

For security, MySQL uses an access privilege and encrypted password system that enables host-based verification. MySQL clients can connect to MySQL Server using several protocols, including TCP/IP sockets on any platform. MySQL also supports a number of client and utility programs, command-line programs and administration tools such as MySQL Workbench.

Offshoots of MySQL, also known as forks, include the following:

- Drizzle, a lightweight open source database management system in development based on MySQL 6.0;
- MariaDB, a popular community-developed "drop-in" replacement for MySQL that uses MySQL APIs and commands;
- Percona Server with XtraDB, an enhanced version of MySQL known for horizontal scalability

Compatibility with other services:

MySQL was designed to be compatible with other systems. It supports deployment in virtualized environments, such as Amazon RDS for MySQL, Amazon RDS for MariaDB and Amazon Aurora for MySQL. Users can transfer their data to a SQL Server database by usingdatabase migration tools like AWS Schema Conversion Tool and the AWS Database Migration Service.

The database object semantics between SQL Server and MySQL are similar, but not identical. There are architectural differences that must be considered when migrating from SQL Server to MySQL. In MySQL, there is no difference between a database and a schema, while SQL Server treats the two as separate entities.

Deployment:

MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are, however, limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server continually pushes binlog events to connected slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

6.1.4 MySQL Functions:

mysql_affected_rows	— Get number of affected rows in previous MySQL
mysql_client_encoding	— Returns the name of the character set
mysql_close	— Close MySQL connection
mysql_connect	— Open a connection to a MySQL Server
mysql_create_db	— Create a MySQL database
mysql_data_seek	— Move internal result pointer
mysql_db_name	— Retrieves database name from the call to mysql
mysql_db_query	— Selects a database and executes a query on it
mysql_drop_db	— Drop (delete) a MySQL database
mysql_error	— Returns the text of the error message from previous operation
mysql_escape_string	— Escapes a string for use in a mysql_query
mysql_fetch_array	— Fetch a result row as an associative array, a numeric array, or both
mysql_fetch_assoc	— Fetch a result row as an associative array
mysql_fetch_field	— Get column information from a result and return as an
object mysql_fetch_lengths	— Get the length of each output in a result
mysql_fetch_object	— Fetch a result row as an object
mysql_fetch_row	— Get a result row as an enumerated array
mysql_field_flags	— Get the flags associated with the specified field in a
result mysql_field_len	— Returns the length of the specified field

mysql_field_name	— Get the name of the specified field in a result
mysql_field_seek	— Set result pointer to a specified field offset
mysql_field_table	— Get name of the table the specified field is in
mysql_field_type	— Get the type of the specified field in a result
mysql_free_result	— Free result memory
mysql_get_client_info	— Get MySQL client info
mysql_get_host_info	— Get MySQL host info
mysql_get_proto_info	— Get MySQL protocol info
mysql_get_server_info	— Get MySQL server info
mysql_info	— Get information about the most recent query
mysql_insert_id	— Get the ID generated in the last query
mysql_list_dbs	— List databases available on a MySQL server
mysql_list_fields	— List MySQL table fields
mysql_list_processes	— List MySQL processes
mysql_list_tables	— List tables in a MySQL database
mysql_num_fields	— Get number of fields in result
mysql_num_rows	— Get number of rows in result
mysql_pconnect	— Open a persistent connection to a MySQL server
mysql_ping	— Ping a server connection or reconnect if there is no
connection mysql_query	— Send a MySQL query
mysql_real_escape_string	— Escapes special characters in a string for use in an SQL
mysql_result	— Get result data
mysql_select_db	— Select a MySQL database
mysql_set_charset	— Sets the client character set
mysql_stat	— Get current system status

4.2 Sample code:

DatabaseCon.java

```
package com.mysql;
import java.sql.*;

public class DatabaseCon
{
    static Connection co;
    public static Connection getConnection()
    {


        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            co = DriverManager.getConnection("jdbc:mysql://localhost:3306/SocialMonitor","root","root");
        }
        catch(Exception e)
        {
            System.out.println("Database Error"+e);
        }
        return co;
    }

}
```

InsertChat.java

```
public class InsertChat
{

public static void main(String data,String user1,String user2){

try{
    Connection con = DatabaseCon.getConnection();
    PreparedStatement ps=con.prepareStatement("insert into msgs(msg,user_,time_, chatbw) values(?,?,?,?)");
    ps.setString(1,data);
    ps.setString(2,user1);
    ps.setString(3,DateDemo.getTime());
    ps.setString(4,user1+"|"+user2);
    ps.executeUpdate();
    }
    catch(Exception e){
        System.out.println(e);
    }


}
public static void clear(String user1,String user2){

try{
    Connection con = DatabaseCon.getConnection();
    PreparedStatement ps=con.prepareStatement("delete from msgs where chatbw=? or chatbw=? ");
    ps.setString(1,user1+"|"+user2);
    ps.setString(2,user2+"|"+user1);
    ps.executeUpdate();
    }
    catch(Exception e){
        System.out.println(e);
    }


}
```

Naive.java

```
Connection con = DatabaseCon.getConnection();
Statement st = con.createStatement();
String sssl = "select * from words";
System.out.println(sssl);
ResultSet rs=st.executeQuery(sssl);
double p_c=0.0;


String docn="";
while(rs.next()){
docn= docn+" "+rs.getString("subject_");
}
res=0; p_c=0;

docn=StopWords.main(docn);

double c2=docn.split("\\s+").length;


p_c=c2/(c1+c2);
pw.print("  P(c)=(c2/c1+c2)="+p_c);


double tk=0;
for(String token:Tokenizer.main(docn))
{
    tk=tk+CountWords.main(doc,token);
}
//Smoothing
tk=tk;

pw.print("  Count Matched tk+1="+tk);

res=tk/(c1+c2);

pw.print("  res=tk+1/(c1+c2+1)="+res);
```

Data base tables

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/`socialmonitor`
DROP TABLE IF EXISTS `frequest`;
CREATE TABLE `frequest` (
  `ufrom` varchar(300) NOT NULL,
  `uto` varchar(300) NOT NULL,
  `requ` varchar(200) DEFAULT NULL
)
DROP TABLE IF EXISTS `friends`;
CREATE TABLE `friends` (
  `user1` varchar(100) NOT NULL,
  `user2` varchar(100) NOT NULL,
  PRIMARY KEY (`user1`,`user2`)
)

CREATE TABLE `msgs` (
  `sno` int(11) NOT NULL AUTO_INCREMENT,
  `msg` longtext,
  `user_` varchar(100) DEFAULT NULL,
  `time_` varchar(100) DEFAULT NULL,
  `chatbw` varchar(200) DEFAULT NULL,
  `status` varchar(100) DEFAULT 'non',
  `ver` varbinary(100) DEFAULT 'non',
  PRIMARY KEY (`sno`)
)

CREATE TABLE `post` (
  `id` int(3) DEFAULT NULL,
  `msg` varchar(500) DEFAULT NULL,
  `userid` varchar(100) DEFAULT NULL,
  `uname` varchar(100) DEFAULT NULL,
  `img` longblob,
  `pdate` varchar(100) DEFAULT NULL
)
CREATE TABLE `profilepic` (
  `email` varchar(200) NOT NULL,
  `pic` longblob,
  PRIMARY KEY (`email`)
)
CREATE TABLE `users` (
  `name` varchar(200) DEFAULT NULL,
  `pwd` varchar(200) DEFAULT NULL,
  `email` varchar(100) NOT NULL,
  `ph` varchar(100) DEFAULT NULL,
  `city` varchar(100) DEFAULT NULL,
  `st_` int(11) DEFAULT '0',
  PRIMARY KEY (`email`)
)
CREATE TABLE `words` (
  `sno` int(11) DEFAULT NULL,
  `subject_` varchar(500) DEFAULT NULL
)
```

4.3 Results:

4.3.1 Registration:

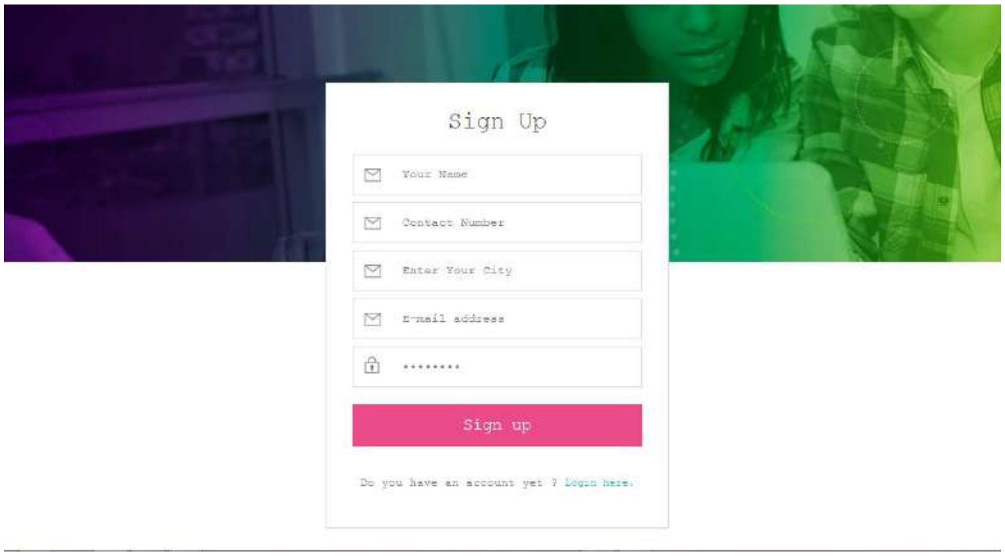


Figure 4.3.1: Registration

Description:

The User can register here using his details and clicks the signup button provided .

4.3.2 Login:

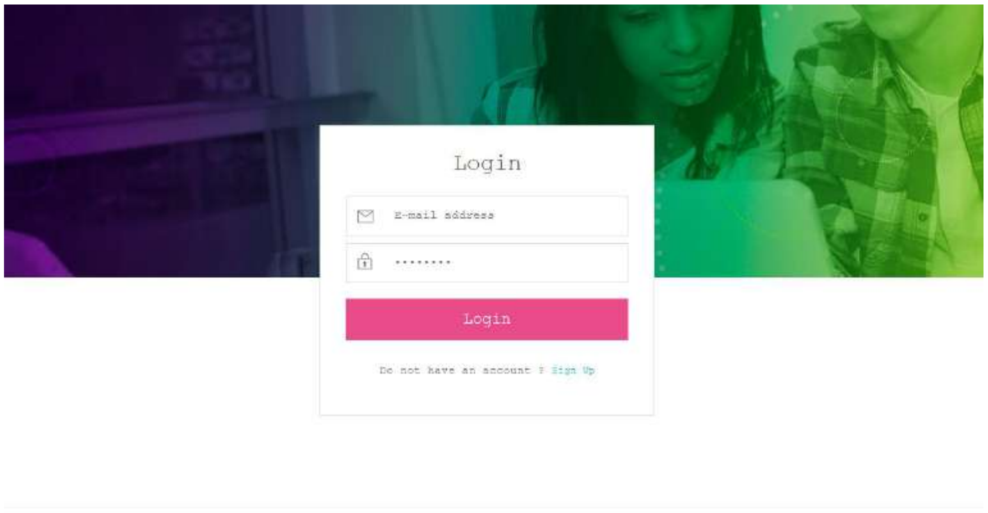


Figure 4.3.2: Login

Description:

The user enters the valid credentials and logs in in to the system.

4.3.3 Profile:

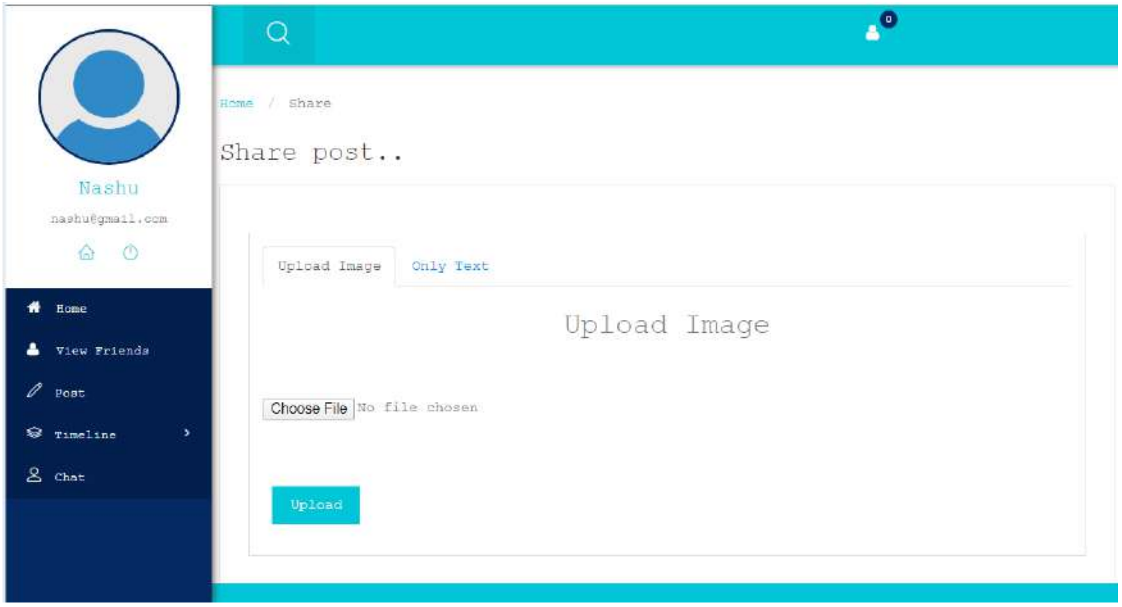


Figure 4.3.3: Profile

Description:

A user **profile** is a visual display of personal data associated with a specific user, or a customized desktop environment

4.3.4 Message section:

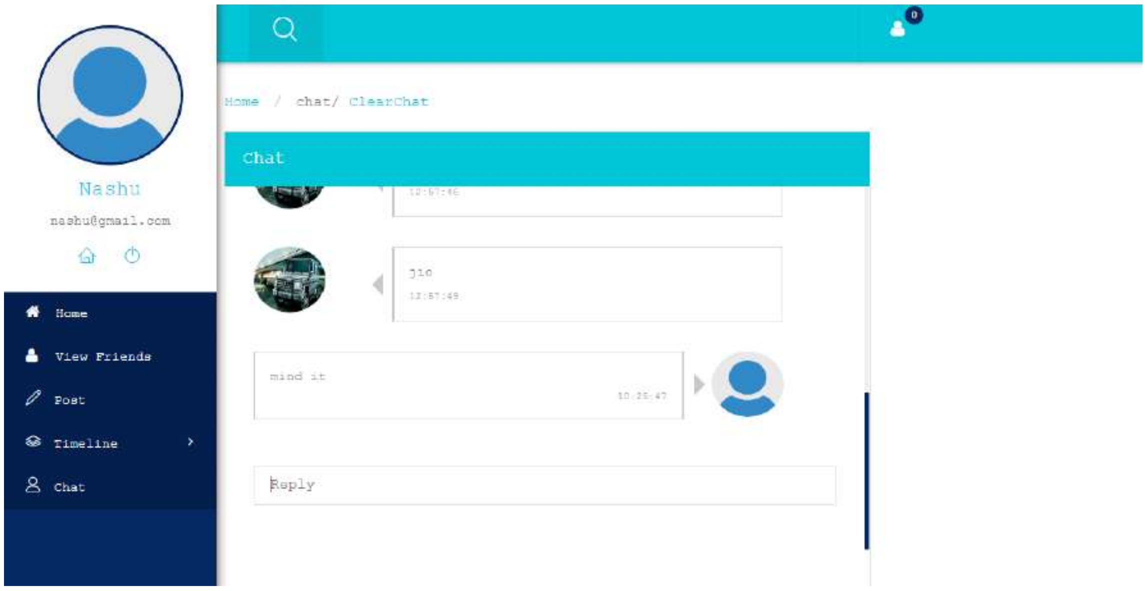


Figure 4.3.4: Message section

Description:

The Users can send messages using this section in social media.

4.3.5 Admin:

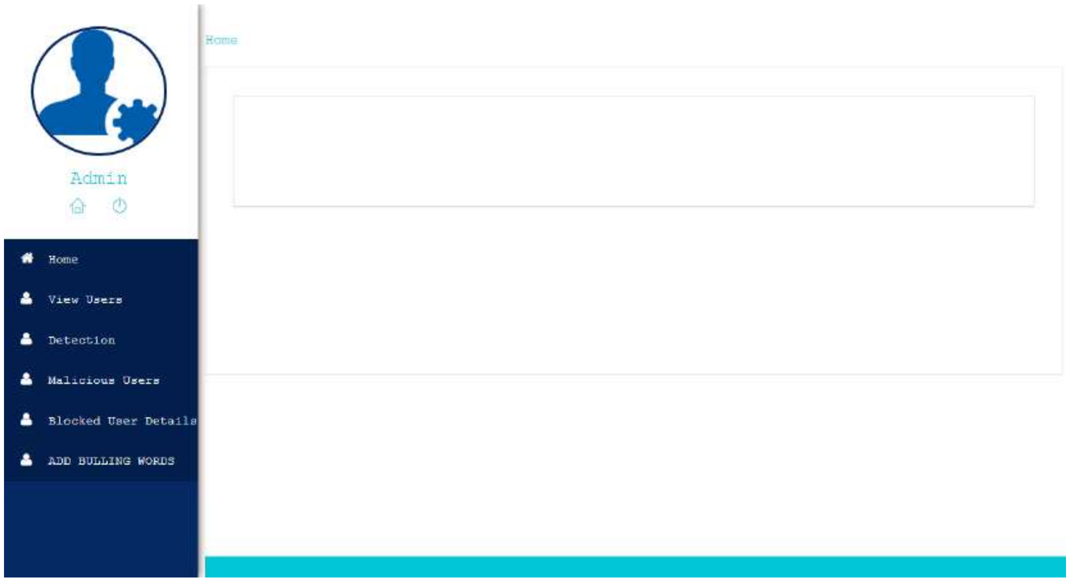
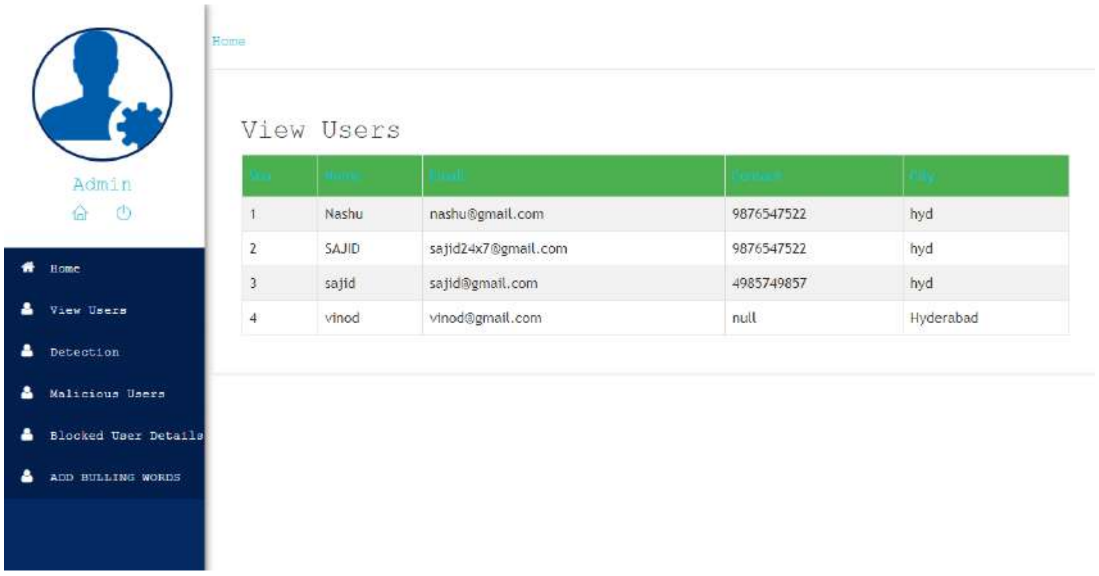


Figure 4.3.5: Admin

Description:

The Admin controls monitors and detects the malicious users in social media.

4.3.6 List of Users:



View Users

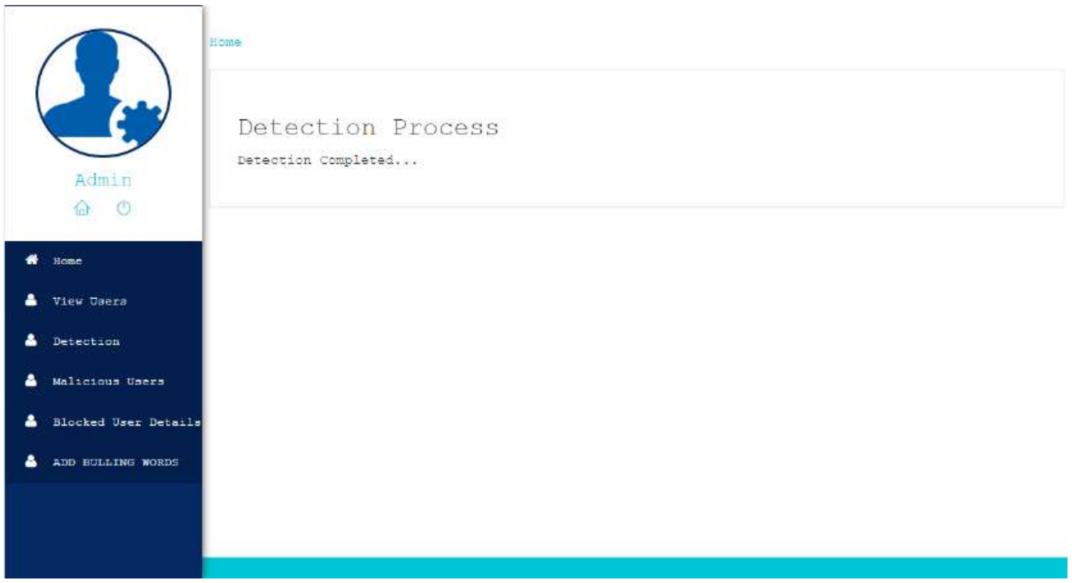
Sno	Name	Email	Contact	City
1	Nashu	nashu@gmail.com	9876547522	hyd
2	SAJID	sajid24x7@gmail.com	9876547522	hyd
3	sajid	sajid@gmail.com	4985749857	hyd
4	vinod	vinod@gmail.com	null	Hyderabad

Figure 4.3.6: List of Users

Description:

The no.of. users present in the system are displayed here.

4.3.7 Malicious User Detection and Blocking:



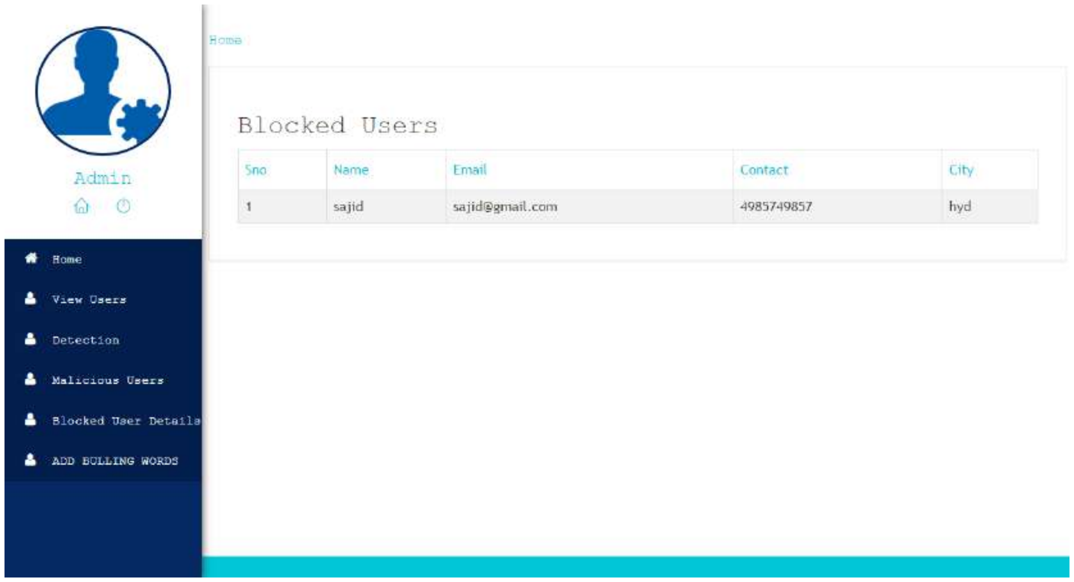
Detection Process

Detection Completed...

Figure 4.3.7: Malicious User detection

Description:

Detection process is carried out here.



Blocked Users

Sno	Name	Email	Contact	City
1	sajid	sajid@gmail.com	4985749857	hyd

Figure 4.3.8: Blocked Users

Description:

The users who are blocked are displayed in the blocked users screen.

CHAPTER-5

SYSTEM TESTING

Testing Software is a critical process which includes many activities, elements of software excellence assertion and represents the ultimate review of specification, design and coding, Software Testing presents a wide nature of an interesting variance for the software developers.

Testing Objectives

- Testing is a series of steps which includes executing a program with various inputs and intent of finding an error from the inputs and making the developer to make corrections on error finding.
- A good Software test case is one that has a possibility of finding an undiscovered error in the designed program.
- A successful Software Testing is one that exposes an unknown or undiscovered error.
- These above objectives imply a dramatic change in view port.
- Software Testing is a series of steps but it cannot show the absence of defects and errors but it can only show various errors that are found software or program.

5.1 Testing Methodologies:

Any Software product can be tested in one of two ways

5.1.1 White Box Testing

White Box Testing is also called as Open or Glass box testing. In White Box Testing, by finding the specified program or function that a software product or a software program has been designed or developed to perform or execute the test can be implemented and conducted for the demonstrates each program or function in a fully operated at the same time finding for errors in each program. It is a glass box or open test case design method that uses the wide control on structure of the procedural program and design to find and drive the test cases. The starting path testing activities is a white box testing, internal activities of the product or program or application can be tested.

5.1.2 Black Box Testing

In Black Box testing by understanding and knowing the various program internal operation of a application or product or program, Black Box Testing can be conducted to guarantee that all gears mesh of the internal activities of the product or program or application can be tested. The process provides a internal operation to check the performance and specifications of all the internal mechanism which have been passably exercised. Black Box Testing fundamentally focuses on the functional activities and requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing
- Graph matrices

5.2 Implementation and Testing:

Implementation is one of the most important tasks in this is the phase in which one has to be cautions because all the efforts undertaken during the system will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

5.2.1 Software Testing Implementation

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

5.2.2 Software Testing Strategies

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

- **System Testing**

  Testing has become an integral part of any system especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

- **Module Testing**

  To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

- **Integration Testing**

  After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system allmodules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

- **Acceptance Testing**

  When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

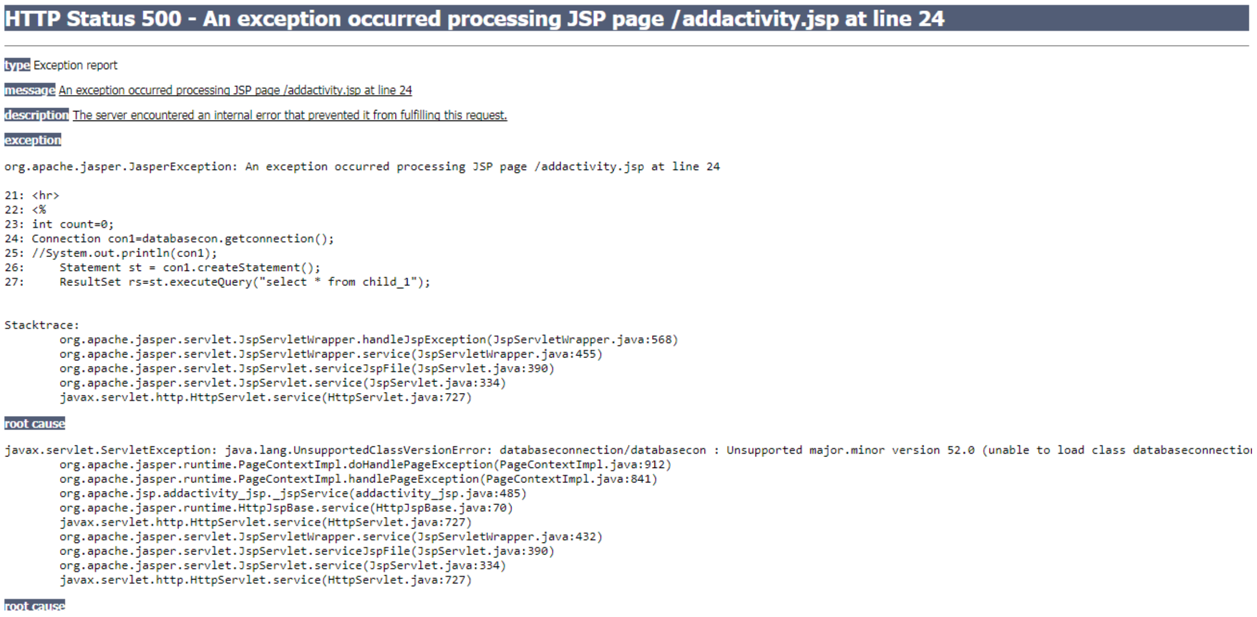
- **Performance Testing**

  In software engineering, performance testing is processed to check the workload, usage of system, memory, processing, network and other system functionalities. It can also serve to investigate measure the program structure and its process activities inside the system. In performance testing, it is often crucial for the test conditions to be similar to the expected actual use. However, in practice this is hard to arrange and not wholly possible, since production systems are subjected to unpredictable workloads. Test workloads may mimic occurrences in the production environment as far as possible, but only in the simplest systems can one exactly replicate this workload variability. It is critical to the cost performance of a new system that performance test efforts begin at the inception of the development project and extend through to deployment.

- **Unit testing**

  The unit testing is done in the stage of implementation of the project only the error are solved in development stage some of the error we come across in development are given below

Class version Error in our application



HTTP Status 500 - An exception occurred processing JSP page /addactivity.jsp at line 24

type Exception report

message An exception occurred processing JSP page /addactivity.jsp at line 24

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
org.apache.jasper.JasperException: An exception occurred processing JSP page /addactivity.jsp at line 24

21: <hr>
22: <%
23: int count=0;
24: Connection con1=databasecon.getconnection();
25: //System.out.println(con1);
26:     Statement st = con1.createStatement();
27:     ResultSet rs=st.executeQuery("select * from child_1");


Stacktrace:
	org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:568)
	org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:455)
	org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
	org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
```

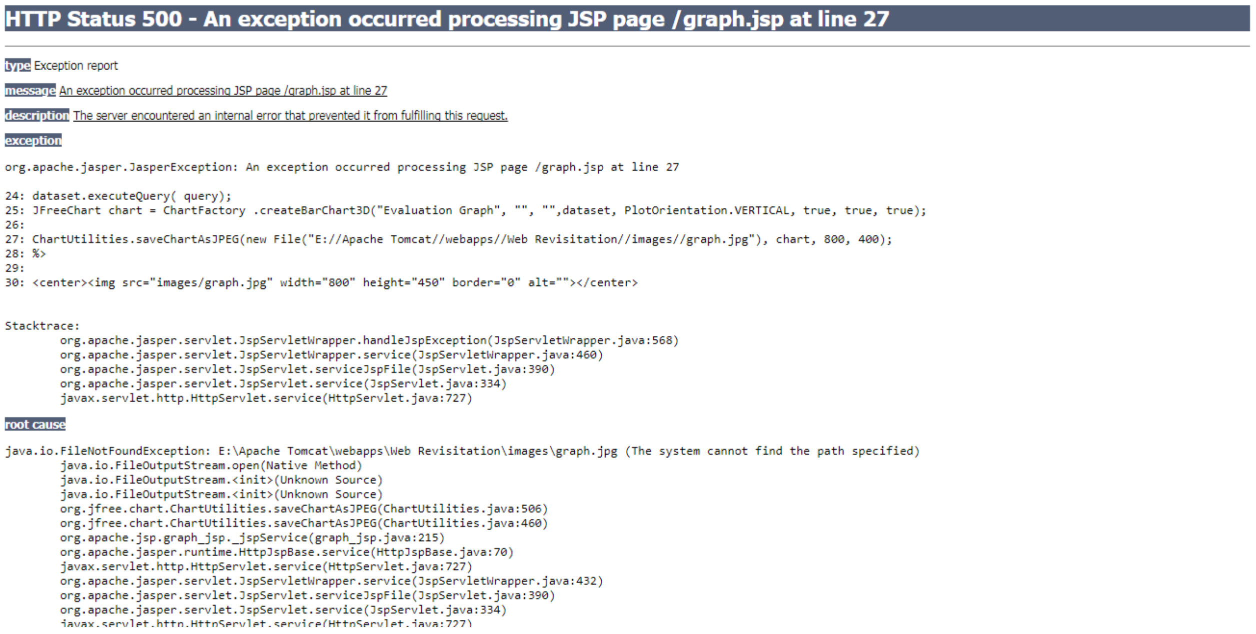
root cause

```
javax.servlet.ServletException: java.lang.UnsupportedClassVersionError: databaseconnection/databasecon : Unsupported major.minor version 52.0 (unable to load class databaseconnectio
	org.apache.jasper.runtime.PageContextImpl.doHandlePageException(PageContextImpl.java:912)
	org.apache.jasper.runtime.PageContextImpl.handlePageException(PageContextImpl.java:841)
	org.apache.jsp.addactivity_jsp._jspService(addactivity_jsp.java:485)
	org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
	org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:432)
	org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
	org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
```

root cause

This Error Come when we move our application from one system to other and mainly when we version issues in the software's we us .

Path related error in our application



HTTP Status 500 - An exception occurred processing JSP page /graph.jsp at line 27

type Exception report

message An exception occurred processing JSP page /graph.jsp at line 27

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
org.apache.jasper.JasperException: An exception occurred processing JSP page /graph.jsp at line 27

24: dataset.executeQuery( query);
25: JFreeChart chart = ChartFactory .createBarChart3D("Evaluation Graph", "", "",dataset, PlotOrientation.VERTICAL, true, true, true);
26:
27: ChartUtilities.saveChartAsJPEG(new File("E://Apache Tomcat//webapps//Web Revisitation//images//graph.jpg"), chart, 800, 400);
28: %>
29:
30: <center><img src="images/graph.jpg" width="800" height="450" border="0" alt=""></center>


Stacktrace:
	org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:568)
	org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:460)
	org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
	org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
```

root cause

```
java.io.FileNotFoundException: E:\Apache Tomcat\webapps\Web Revisitation\images\graph.jpg (The system cannot find the path specified)
	java.io.FileOutputStream.open(Native Method)
	java.io.FileOutputStream.<init>(Unknown Source)
	java.io.FileOutputStream.<init>(Unknown Source)
	org.jfree.chart.ChartUtilities.saveChartAsJPEG(ChartUtilities.java:506)
	org.jfree.chart.ChartUtilities.saveChartAsJPEG(ChartUtilities.java:460)
	org.apache.jsp.graph_jsp._jspService(graph_jsp.java:215)
	org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
	org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:432)
	org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
	org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
```

This Error Came when I have Performance Metrics to show in graph when I missed my server directly path in the system so we got this error in the applicant in development stage

5.3 Test Cases:

5.3.1 Registration test case

Test Case ID 1		**Test Case Description** - Validations in Registration Form		
S.No.	**Prerequisites**	**S#**	**Test Data Requirement**	
1	User should be Registered	1	Data should be valid	
Test Condition Entering data in registration form				
Step No.	**Step Details**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	User gives First and Last Name	Pop showing email verification message	Enter valid email/password	Fail
2	Submitting the form without entering any details	Pop showing email verification message	Enter email /password	Fail
3	User enters invalid format of email id	Pop showing email verification message	Enter valid email id	Fail
4	User enters a phone number with < 10 digits	Pop showing email verification message	Enter valid phone number	Fail
5	Entering valid username and password	Pop showing email verification message	Pop showing email verification message	Pass

5.3.2 Login test case

Test Case ID #2		**Test Case Description** - Validations in Login Form		
S#	**Prerequisites**	**S#**	**Test Data Requirement**	
1	User should have an email id	1	Data should be valid	
Test Condition				
Entering data in login form				
Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass/Fail/Not Executed/Suspended**
1	User gives aemail or password of <6 characters	User logged in	Enter valid email/password	Fail
2	Submitting the form without entering any details	User logged in	Enter email /password	Fail
3	User enters wrong Email and (or) password	User logged in	Enter correct email /password	Fail

5.3.3 Validation testing for project application

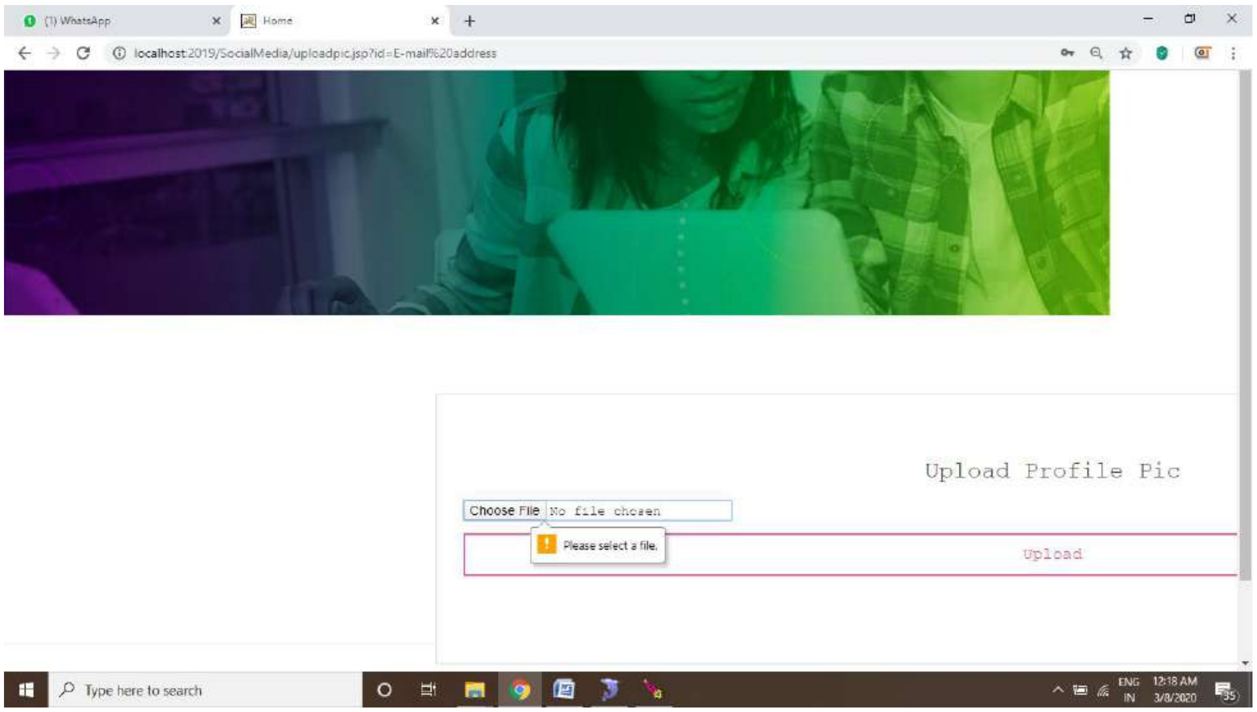

localhost:2019/SocialMedia/uploadpic.jsp?id=E-mail%20address
Upload Profile Pic
Choose File No file chosen
Please select a file.
Upload

Figure 5.3.1: File not Selected Validation

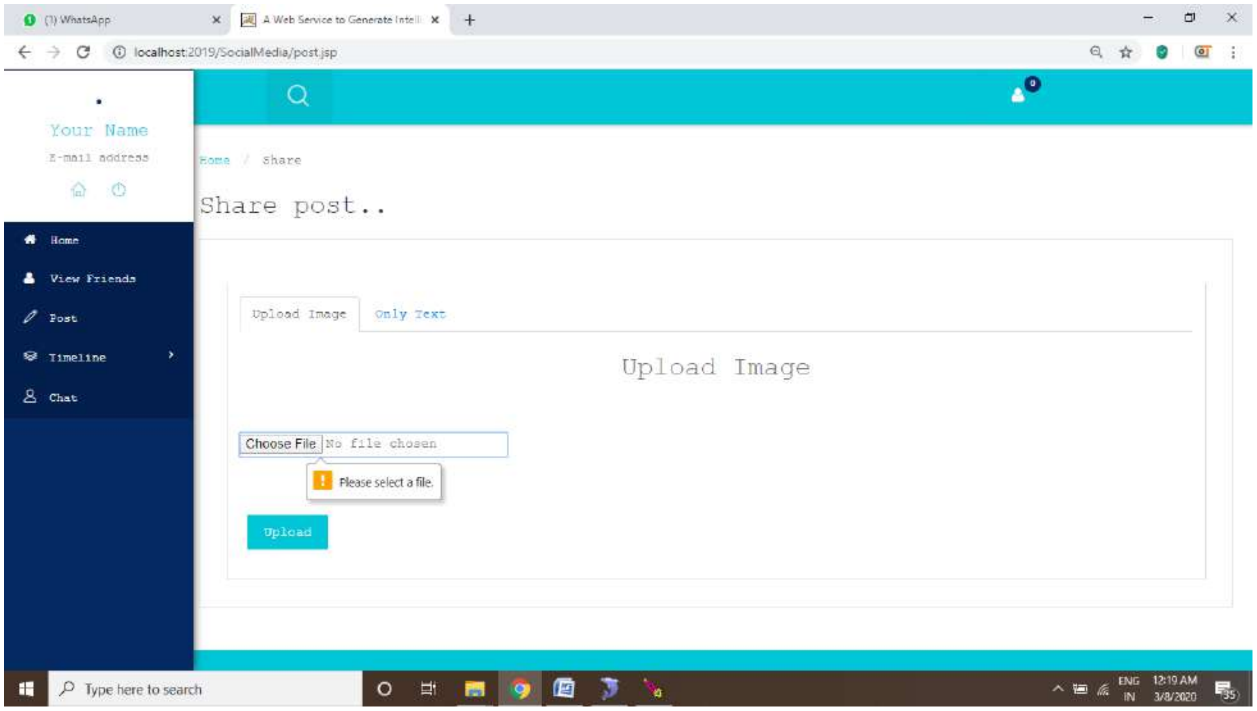


Figure 5.3.2: No Photo uploaded Validation

CHAPTER-6

CONCLUSION AND FUTURE ENHANCEMENT

Lot of anonymous users accounts are creating very rapidly. We need to focus for the tracking the anonymous users. In our project we have calculated the user behavior according to the chat statements of the user which he/she do with others. By taking the advantages Machine Learning algorithms we classify the anonymous users. By using Naïve Bayes algorithm to perform the classification of the users.

Social media will be more integrated into personal, social and business lives**.** Without realizing it, these platforms will be a natural part of our lives, streamlining our everyday activities and work. Social media will have longer-term implications for individuals as a result of a life lived in public. Think of it as George Orwell's 1984 come to life. Instead of a government documenting our lives, individuals and businesses, of their own freewill, will create these digital footprints from birth**.** Social media will translate personal information and data into a form of currency**.** As a result, privacy and control of personal information will grow in importance.

BIBLIOGRAPHY

- T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, “Identifying users across social tagging systems,” in Proc. 5th Int. AAAI Conf. Weblogs Social Media,
- T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, “Identifying users across social tagging systems,”
- Narayanan and V. Shmatikov, “De-anonymizing social networks,”
- www.google.com
- www.javatpoint.com
- www.w3schools.com
- www.tutorialspoint.com
- N. Korula and S. Lattanzi, “An efficient reconciliation algorithm for social networks,” arXiv preprint arXiv:1307.1690, 2013