

A Report On
“Social Network Mental Disorder Detection”

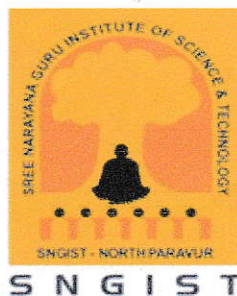
Submitted to the
Department of Computer Applications

In partial fulfillment of the Course
Master of Computer Applications

Under the guidance of
Mrs Reashma Raj P R

BY

Limi Michael
(Reg no: SGI17MCA-D033)



DEPARTMENT OF COMPUTER APPLICATIONS

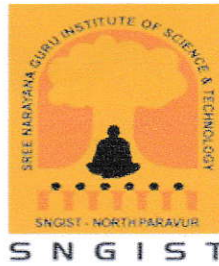
SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520

2017-2019

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520



BONAFIDE CERTIFICATE

Certified that the Project Work entitled

“Social Network Mental Disorder Detection”

is a bonafide work done by

Limi Michael

In partial fulfillment of the requirement for the

Award of

MASTER OF COMPUTER APPLICATIONS

Degree From

APJ Abdul Kalam Technological University, Thiruvananthapuram

(2017-2019)

Head of Department



6/5/19
Project Guide

Submitted for the Viva-Voce Examination held on

External Examiner1

External Examiner2

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520



CERTIFICATE

This is to certify that the project entitled "**Social Network Mental Disorder Detection**" has been successfully carried out by **Limi Michael**(Reg no:SGI17MCA-D033) in partial fulfillment of the Course **Master of Computer Applications**.

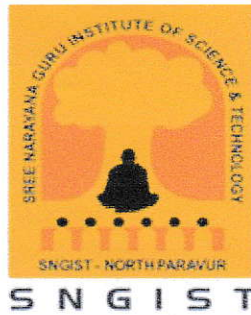
Mrs. Kavitha C R

HEAD OF DEPARTMENT

Date: 06 / 06 / 2019

SNGIST GROUP OF INSTITUTIONS


North Paravur- 683520



CERTIFICATE

This is to certify that the project entitled “**Social Network Metal Disorder Detection**” has been successfully carried out by **Limi Michael**(Reg no:SGI17MCA-D033) in partial fulfillment of the course **Master of Computer Applications** under my guidance .

Date: 6/6/19


6/6/19
Mrs. Reashma Raj P R
INTERNAL GUIDE

Project Completion Certificate

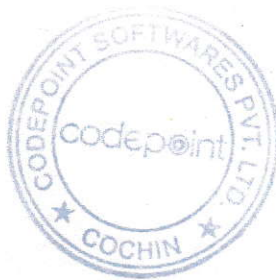
This is to certify that Ms. Limi Michael (Reg No:SGI17MCA-D033) S4 Second Year Direct M.C.A degree student of SNGIST Group of Institutions, N.PARAVUR has successfully completed her main project on the topic "Social Network Mental Disorder Detection" from CODEPONT SOFTWARES PVT. LTD during 20-Feb-2019 to 03-June-2019



PROJECT COORDINATOR

SARITHA M

CODEPONT SOFTWARES PVT.LTD

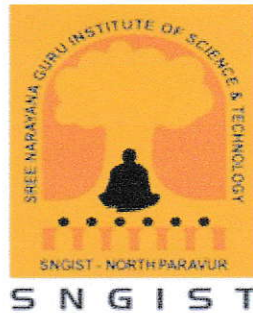


05-JUNE-2019

KOCHI

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520



DECLARATION

I, **Limi Michael**, hereby declare that the project work entitled “**Social Network Mental Disorder Detection**” is an authenticated work carried out by me under the guidance of **Mrs. Reashma Raj P R** for the partial fulfillment of the course **MASTER OF COMPUTER APPLICATIONS**. This work has not been submitted for similar purpose anywhere else except to **SNGIST GROUP OF INSTITUTIONS, North Paravur**, affiliated to **APJ ABDUL KALAM UNIVERSITY, THIRUVANANTHAPURAM**. I understand that detection of any such copying is liable to be punished in any way the college deems fit.

Date: 06/06/2019

Place: North Paravur

Limi Michael

TABLE OF CONTENTS

Pages

1 Executive Summary	01
2 Introduction	01
2.1 Existing System	02
2.2 Definition of problem	02
2.3 Proposed System	02
2.4 Objective of the project	03
2.5 Scope of the project	03
3 Methodology	03
3.1 Scrum	03
3.2 Scrum Roles	04
3.3 Sprint Planning Meeting	04
3.4 Daily Scrum Meeting	04
3.5 Sprint Review Meeting	05
3.6 Product Backlog	05
4 Development Tools	06
4.1 Introduction to java	06
4.2 Features of Java	06
4.3 Jsp	07
4.4 Html	07
4.5 Netbeans	07
4.6 Weka	07
4.7 MySql5.5	08
4.8 GIT12	08
5 Milestones	08
5.1 Sprint 1	08
5.2 Sprint 2	09
5.3 Sprint 3	09
5.4 Sprint 4	09
5.5 Sprint 5	09
6 Module Description	09
7 System Design	10
7.1 Database Design	10
7.2 UI Design	14
8 UML Diagrams	21
8.1 Use Case Diagram	21
8.2 Class Diagram	22
9 Testing	23
9.1 Test Cases	23
9.2 Test Report	24
10 System Implementation	24

11 Conclusion & Future Enhancement	25
12.1 Appendix A	26
12.1.1 Sample Source Code	26
12.2 Appendix B	37
12.2.1 Bibliography	37

ACKNOWLEDGEMENT

I express my sense of gratitude and sincere thanks to Almighty for his abiding presence and the abounding grace of unseen hand yet tangible guidance all through the formation of this project.

I am thankful to **Prof Dr M Sivanadhan, Principal** SNGIST Group of Institutions for his kind support in all aspect during my study.

I sincerely thank **Mrs. Kavitha C R, HOD**, Department of Computer Applications, SNGIST Group of Institutions, for her encouragement and inspiration for execution of this project work.

I especially thank to my Internal guide **Mrs. Reashma Raj P R**., faculty, SNGIST Group of Institutions, for her guidance and constant supervision as well as for providing necessary information regarding the project.

I want to thank the Department of Computer Applications for giving me the permission to prepare the project on the topic "Social Network Mental Disorder Detection".

Limi Michael

1. EXECUTIVE SUMMARY

The explosive growth in popularity of social networking leads to the problematic usage. An increasing number of social network mental disorders (SNMDs), such as Cyber-Relationship Addiction, Information Overload, and Net Compulsion, have been recently noted. Symptoms of these mental disorders are usually observed passively today, resulting in delayed clinical intervention. In this paper, mining online social behavior provides an opportunity to actively identify SNMDs at an early stage. It is challenging to detect SNMDs because the mental status cannot be directly observed from online social activity logs. Our approach, new and innovative to the practice of SNMD detection, does not rely on self-revealing of those mental factors via questionnaires in Psychology. Instead, we propose a machine learning framework, namely, Social Network Mental Disorder Detection (SNMDD), that exploits features extracted from social network data to accurately identify potential cases of SNMDs.

2. INTRODUCTION

With the explosive growth in popularity of social network have become a part of many people's daily lives. Most research on social network mining focuses on discovering the knowledge behind the data for improving people's life. While OSNs seemingly expand their users' capability in increasing social contacts, they may actually decrease the face-to-face interpersonal interactions in the real world. Due to the epidemic scale of these phenomena, new terms such as Phubbing (Phone Snubbing) and Nomophobia (NoMobile Phone Phobia) have been created to describe those who cannot stop using mobile social networking apps.

Social network mental disorder detection site helps to detect some cases of disorders like addiction. For implementing the concept there is a need of resource for mining appropriate features. But the lack of availability of social network data here created a small blogging site. Using this site the concept of social network mental disorder detection concept is implemented. From the blogging site features for prediction is extracted using Rule Based Classification. The result of the feature values then passed as test data set to the naïve Bayes classification. Which predict the final result using the training data set along with test data.

2.1. Existing System

Excessive use of social networking apps – usually associated with a loss of the sense of time or a neglect of basic drives, and withdrawal – including feelings of anger, tension, and/or depression when the computer/apps are inaccessible. SNMDs are social-oriented and tend to happen to users who usually interact with others via online social media. Those with SNMDs usually lack offline interactions, and as a result seek cyber-relationships to compensate.

Today, identification of potential mental disorders often falls on the shoulders of supervisors (such as teachers or parents) passively. However, since there are very few notable physical risk factors, the patients usually do not actively seek medical or psychological services. Therefore, patients would only seek clinical interventions when their conditions become very severe.

2.2 Problem Definition

Generally, there are difficulties involved in determining mental health in OSN due to a few issues in non-face-to-face communication and human-computer interaction. The main difficulty is usually from the language barrier that commonly happened while determining the exact meaning of mental health behind the words and language written in the OSN. However, there are several ways to solve this issue, and one of them is through machine learning to learn, understand and determine the possibility of mental health behind the words and language written in OSN. It is difficult to interpret the mental health detection as the OSN users nowadays usually use other types of data to express themselves in OSN. Some of them introduce features selection which is limited to their current study. A good feature selection will improve the performance of learning algorithms.

2.3 Proposed System

Today online SNMDs are usually treated at a late stage. To actively identify potential SNMD cases, we propose an innovative approach, new to the current practice of SNMD detection, by mining data logs of OSN users as an early detection system. We develop a machine learning framework to detect SNMDs, called Social Network Mental Disorder Detection (SNMDD). We also design and analyze many important features for identifying SNMDs from OSNs, such as disinhibition, parasociality, self-disclosure, etc. The proposed framework can be deployed to

provide an early alert for potential patients. In this project, using rule based classification certain features are filtered based on that using naïve Bayes classification the final prediction is made.

2.4 Objective of the Project

To explore data mining techniques to detect types of SNMDs .to explore multiple data sources (i.e., OSNs) in SNMDD, in order to derive a more complete portrait of users' behavior and effectively deal with the data scarcity problem to employ tensor decomposition to extract common latent factors from different sources and objects.

2.5 Scope of the Project

Identify the disorder as soon as possible. To actively identify potential SNMD cases, we propose an innovative approach, new to the current practice of SNMD detection, by mining data logs of OSN users as an early detection system. We develop a machine learning framework to detect SNMDs, called Social Network Mental Disorder Detection (SNMDD). We also design and analyze many important features for identifying SNMDs from OSNs, such as disinhibition, parasociality, self-disclosure, etc. The proposed framework can be deployed to provide an early alert for potential patients.

3. METHODOLOGY

3.1 Scrum

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. This is why in Scrum development, for example, a sprint planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint) instead of a set of Entry criteria, Task definitions, Validation criteria, Exit criteria (ETVX) and so on, as would be provided in most methodologies. Scrum relies on a self-organizing, cross-functional team. The scrum team is self-organizing in that there is no overall team leader who decides which person will do which task or how a problem will be solved. Those are issues that are decided by the team as a whole.

3.2 Scrum Roles

Product Owner

MsSaritha M, was the product owner for this project, and acted as spokesman for the customer and defines features of the product based on each Backlog item or each specific request of the customer. She would prioritize these features according to the market value, decide on a release date for the product, and is responsible for the profitability of the product. The product owner should also adjust the contents of the features and their priority after every Sprint, and decide if what has been produced is acceptable.

Scrum Master

Mrs. Kavitha C.R, HOD MCA was the Scrum master for this project. The Scrum master is responsible for making sure a Scrum team lives by the values and practices of Scrum, and for removing any impediments to the progress of the team. As such, she should shield the team from external interferences, and ensure that the Scrum process is followed, including issuing invitations to the daily Scrum meetings.

Scrum Team

The Scrum team consists of a group of people developing the software product. In this project, the scrum team consists of MsSaritha M, the product owner, Mrs. Kavitha C.R, who acted as Scrum master as well as the project supervisor and Ms. Limi Michael, Developer. There is no personal responsibility in Scrum, the whole team fails or succeeds as a single entity.

3.3 Sprint Planning Meeting

Most of the time our sprint planning meetings went as planned, though some times the product owner was unavailable. In these cases the meeting simply needed to be scheduled one or two days later. These extra days would come in handy for cleaning up what we had produced the earlier sprint.

3.4 Daily Scrum Meeting

Our daily Scrums took place at 10.15. People could arrive as early as 9.00 and work until then, but as long as they did arrive before the meeting started it did not matter (formally).

3.5 Sprint Review Meeting

Our review meetings were always held on Fridays. The product owner would visit the team project room along with any other interested parties, and the team would demonstrate new features on a live system, and answer any questions that might arise during the demo. Usually we would spend one or two days before the demo checking if everything was working, and run test demonstrations internally.

3.6 Product Backlog

- As an administrator wants to login
- As an administrator wants to logout
- As an administrator wants to view users
- As an administrator wants to view posts
- As an administrator wants to view mental health
- As an administrator wants to message to user
- As a user wants to login
- As a user wants to logout
- As a user wants to register
- As a user wants to Post
- As a user wants to view Post
- As a user wants to comment to the post
- As a user wants to like Post
- As a user wants to follow other users

4. DEVELOPMENT TOOLS

4.1 Introduction to Java

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The target of Java is to write a program once and then run this program on multiple operating systems. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Java is defined by a specification and consists of a programming language, a compiler, core libraries and a runtime (Java virtual machine). The Java runtime allows software developers to write program code in other languages than the Java programming language which still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries.

4.2 Features of Java

- Platform independent: Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program (which is standard-compliant and follows certain rules) can run unmodified on all supported platforms, e.g., Windows or Linux.
- Object-orientated programming language: Except the primitive data types, all elements in Java are objects.
- Strongly-typed programming language: Java is strongly-typed, e.g., the types of the used variables must be pre-defined and conversion to other objects is relatively strict, e.g., must be done in most cases by the programmer.
- Interpreted and compiled language: Java source code is transferred into the byte code format which does not depend on the target platform. These byte code instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compiler which translates performance critical byte code instructions into native code instructions.
- Automatic memory management: Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector automatically deletes objects to which no active pointer exists.

4.3 JSP

JSP technology is used to create dynamic web applications. JSP pages are easier to maintain than a Servlet. JSP pages are opposite of Servlets as a servlet adds HTML code inside Java code, while JSP adds Java code inside HTML using JSP tags. Everything a Servlet can do, a JSP page can also do it. JSP enables us to write HTML pages containing tags, inside which we can include powerful Java programs. Using JSP, one can easily separate Presentation and Business logic as a web designer can design and update JSP pages creating the presentation layer and java developer can write server side complex computational code without concerning the web design. And both the layers can easily interact over HTTP requests.

4.4 HTML

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

4.5 NetBeans IDE 8.1

NetBeans is an open-source integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages. NetBeans is also referred to as a platform of modular components used for developing Java desktop applications. NetBeans is coded in Java and runs on most operating systems with a Java

Virtual Machine (JVM), including Solaris, Mac OS, and Linux. NetBeans uses components, also known as modules, to enable software development. NetBeans dynamically installs modules and allows users to download updated features and digitally authenticated upgrades. NetBeans IDE modules include NetBeans Profiler, a Graphical User Interface (GUI) design tool, and NetBeans JavaScript Editor. NetBeans framework reusability simplifies Java Swing desktop application development, which provides platform extension capabilities to third-party developers.

4.6 Weka 3.6

Waikato Environment for Knowledge Analysis (**Weka**) It's a data mining/machine learning tool developed by University of Waikato.⁴⁹ data processing tools,⁷⁶ classification / regression

algorithms, 8 clustering algorithms, 3 algorithms for finding association rule and 15 attribute / subset evaluators + 10 search algorithms for feature selection.

4.7 MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. It is becoming so popular because of many good reasons – It is released under an open-source license. So you have nothing to pay to use it. It is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets. It is very friendly to PHP, the most appreciated language for web development. It supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

4.8 GIT Lab

GitLab is a web-based Git-repository manager with wiki and issue-tracking features, using an open-source license, developed by GitLab Inc. The software was written by Dmitriy Zaporozhets, Anita Arya, and Valery Sizov. As of December 2016, the company has 150 team members and more than 1400 open-source contributors. It is used by organizations such as IBM, Sony, Research Center, NASA, Alibaba, Invincea, O'Reilly Media, Leibniz-Rechenzentrum (LRZ), CERN and SpaceX. The code was originally written in Ruby, with some parts later rewritten in Go.

5. MILESTONES

5.1. Sprint 1

The first meeting was held on 1st January 2019. The team consisted of Ms Saritha S, the product owner and Ms. Limi Michael, the developer. The requirement collection was mainly done in this sprint which took nearly 3 weeks. And the sprint ended on 28th January 2019.

5.2. Sprint 2

Second sprint started on 22nd January 2019 in which the database design was started and the user interface of the system was also designed

5.3. Sprint 3

Third sprint started on 11th February 2019 in which the one of the main modules that is organization module's and that of admin module's development was started it also include the development of sub modules.

5.4 Sprint 4

Fourth sprint started on 3rd march 2019 in which User module and worker module was developed.

5.5 Sprint 5

Fourth sprint started on 21st march 2019 in which testing and validation are done for each module.

6. MODULE DESCRIPTION

Admin

Admin module is the main module. Admin can view users and their posts. As well as admin can inspect their mental health and send message to the user giving a warning of their usage the very early stage. The prediction is done by choosing the features by rule based classification method. And the features are then passed to the naïve bayes classifier.

User

First user needs to register it. User can edit their profile with suitable pictures. User is able to send tweets. And can like and comment to the other Tweets. User also can follow other users and view their tweets.

7. SYSTEM DESIGN

7.1 Database Design

7.1.1 admin_login

Field	Type	Description
login_id	Int	Primary key
usernameAd	Varchar(45)	Username of Admin
passAd	Varchar(10)	Password of admin

7.1.2 keyword_tbl

Field	Type	Description
key_id	int	Primary key
keyword	varchar(45)	Negative keywords

7.1.3 message_tbl

Field	Type	Description
msg_id	int	Primary key
uid	int	Foreign key[user_tbl]
subject	varchar(150)	Subject of the message
msg	varchar(4500)	Message to send
send_at	timestamp	Date and time of message
msg_status	boolean	Message is read or not

7.1.4 loginstatus

Field	Type	Description
status_id	int	Primary key
user	int	Foreign key[user_tbl]
checkIn	timestamp	Login time and date
checkOut	timestamp	Logout time and date
duration	varchar(450)	Active time in seconds

7.1.5 user_tbl

Field	Type	Description
user_id	int	Primary key
username	varchar(45)	Username for user
password	varchar(10)	Password for user
fname	varchar(45)	First name of user
lname	varchar(45)	Last name of user
mailId	varchar(45)	Email id of user
dob	date	Date of birth
gender	tinyint(4)	Gender
location	varchar(45)	Country of user
aboutUser	varchar(145)	Bio of user
profile_id	long blob	Profile picture of user
signUp_on	timestamp	Date ,time of registration

7.1.6 tweet_tbl

Field	Type	Description
tweet_id	int	Primary key
user_id	int	Foreign key[user_tbl]
content	varchar(4500)	Tweet content
img	longblob	Pictures for tweet
created_at	timestamp	Date and time of tweet

7.1.7 like_tbl

Field	Type	Description
like_id	int	Primary key
like_tweetId	int	Foreign key[user_id]
like_userId	int	Foreign key[user_id]
Idate	timestamp	Date and time of like

7.1.8 cmt_tbl

Field	Type	Description
cmt_id	int	Primary key
tweetId	int	Foreign key[tweet_id]
cmt_user	int	Foreign key[user_id]
cmt	varchar(145)	Comment
cmt_date	timestamp	Date of comment

7.1.9 retweet_tbl

Field	Type	Description
retweet_id	int	Primary key
tid	int	Foreign key[tweet_id]
uid	int	Foreign key[user_id]
rcontent	varchar(145)	Retweet content
retweet_date	timestamp	Date of retweet

7.1.10 relation_tbl

Field	Type	Description
rid	int	Primary key
follower_id	int	Foreign key[user_id]
following_id	int	Foreign key[user_id]
rdate	timestamp	Date of relationship

7.1.11 location

Field	Type	Description
loc_id	int	Primary key
locFlag	varchar(45)	Flag icon class
locName	varchar(45)	Location name

7.1.12 update_status

Field	Type	Description
update_id	int	Primary key
userId	int	Foreign key[user_id]
update_at	timestamp	Date of profile update

7.1.13 block_tbl

Field	Type	Description
blk_id	int	Primary key
uid	int	Foreign key[user_id]
blk_by	int	Foreign key[user_id]
blocked_at	timestamp	Date of unfollow

7.2 UI Design

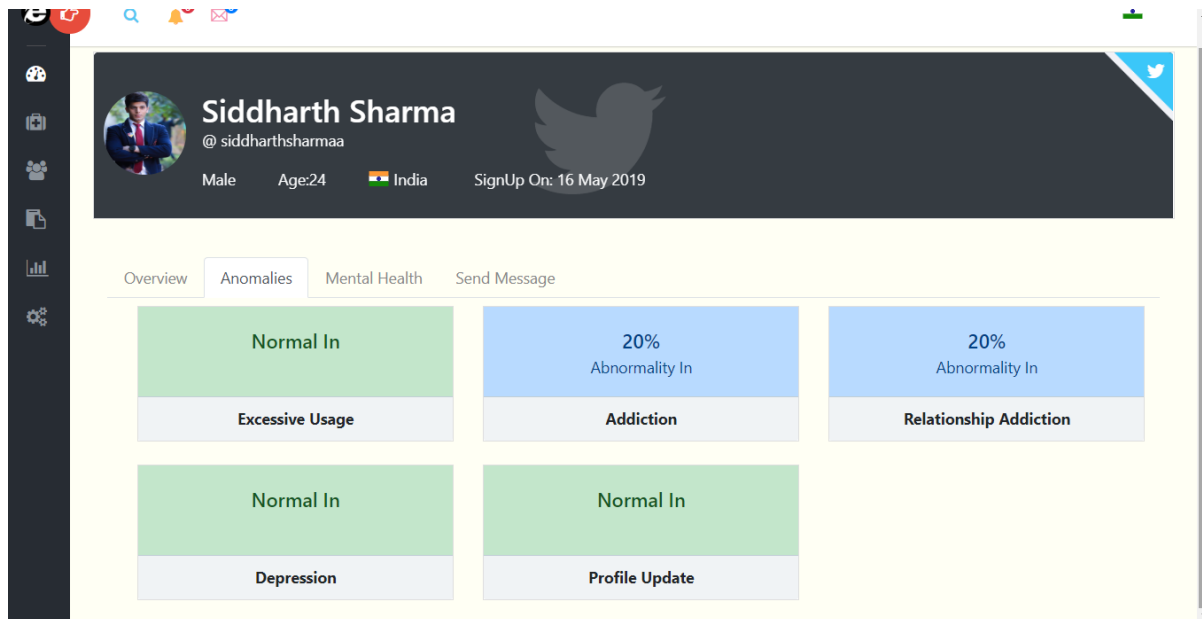


Fig 7.2.1 Anomalies Page

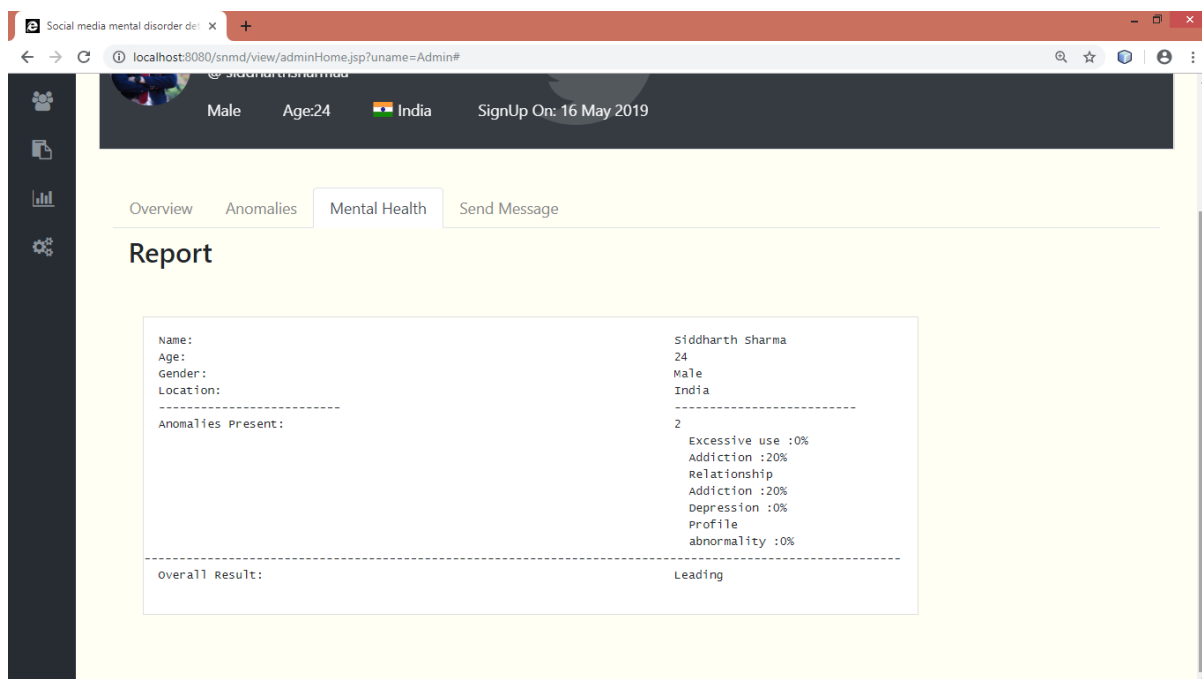


Fig 7.2.2 Mental Health Report

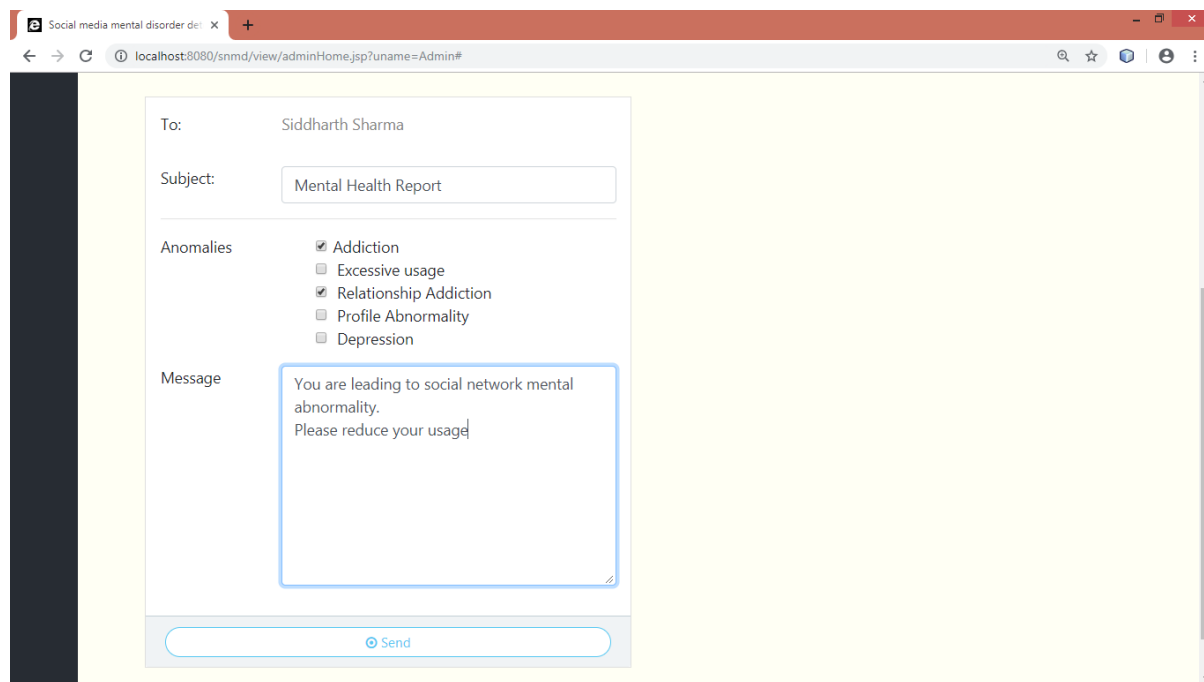


Fig 7.2.3 Message sent Page

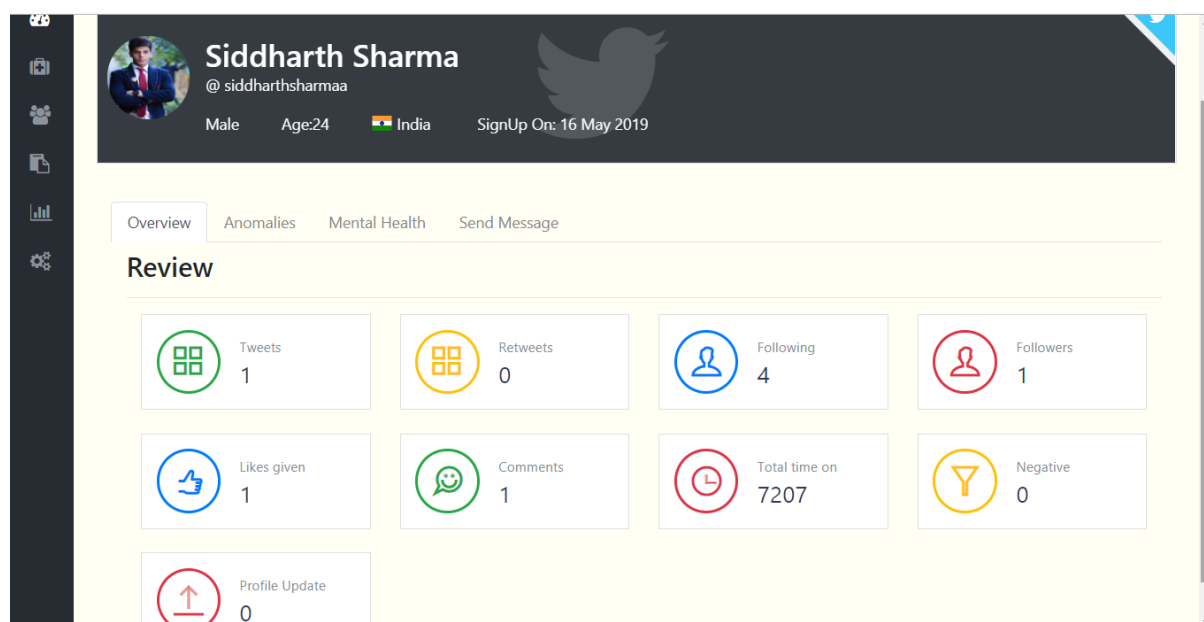
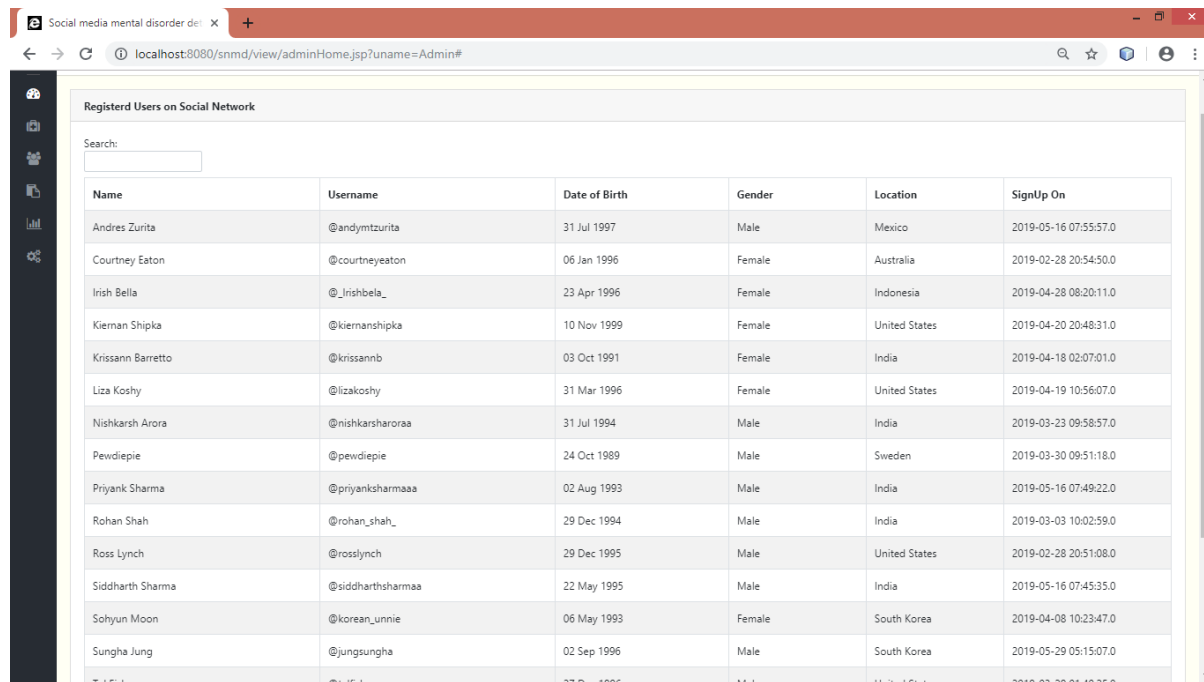


Fig 7.2.4 Counts view



Registered Users on Social Network

Search:

Name	Username	Date of Birth	Gender	Location	SignUp On
Andres Zurita	@andymtzurita	31 Jul 1997	Male	Mexico	2019-05-16 07:55:57.0
Courtney Eaton	@courtneyeaton	06 Jan 1996	Female	Australia	2019-02-28 20:54:50.0
Irish Bella	@_Irishbela_	23 Apr 1996	Female	Indonesia	2019-04-28 08:20:11.0
Kiernan Shipka	@kiernanshipka	10 Nov 1999	Female	United States	2019-04-20 20:48:31.0
Krissann Barretto	@krissannb	03 Oct 1991	Female	India	2019-04-18 02:07:01.0
Liza Koshy	@lizakoshy	31 Mar 1996	Female	United States	2019-04-19 10:56:07.0
Nishkarsh Arora	@nishkarsharoraa	31 Jul 1994	Male	India	2019-03-23 09:58:57.0
Pewdiepie	@pewdiepie	24 Oct 1989	Male	Sweden	2019-03-30 09:51:18.0
Priyank Sharma	@priyanksharmaaa	02 Aug 1993	Male	India	2019-05-16 07:49:22.0
Rohan Shah	@rohan_shah_	29 Dec 1994	Male	India	2019-03-03 10:02:59.0
Ross Lynch	@rosslynch	29 Dec 1995	Male	United States	2019-02-28 20:51:08.0
Siddharth Sharma	@siddharthsharmaa	22 May 1995	Male	India	2019-05-16 07:45:35.0
Sohyun Moon	@korean_unnie	06 May 1993	Female	South Korea	2019-04-08 10:23:47.0
Sungba Jung	@jungsungba	02 Sep 1996	Male	South Korea	2019-05-29 05:15:07.0
Tal Fishman	@talfishman	27 Dec 1996	Male	United States	2019-03-30 09:10:35.0

Fig 7.2.5 Admin view user

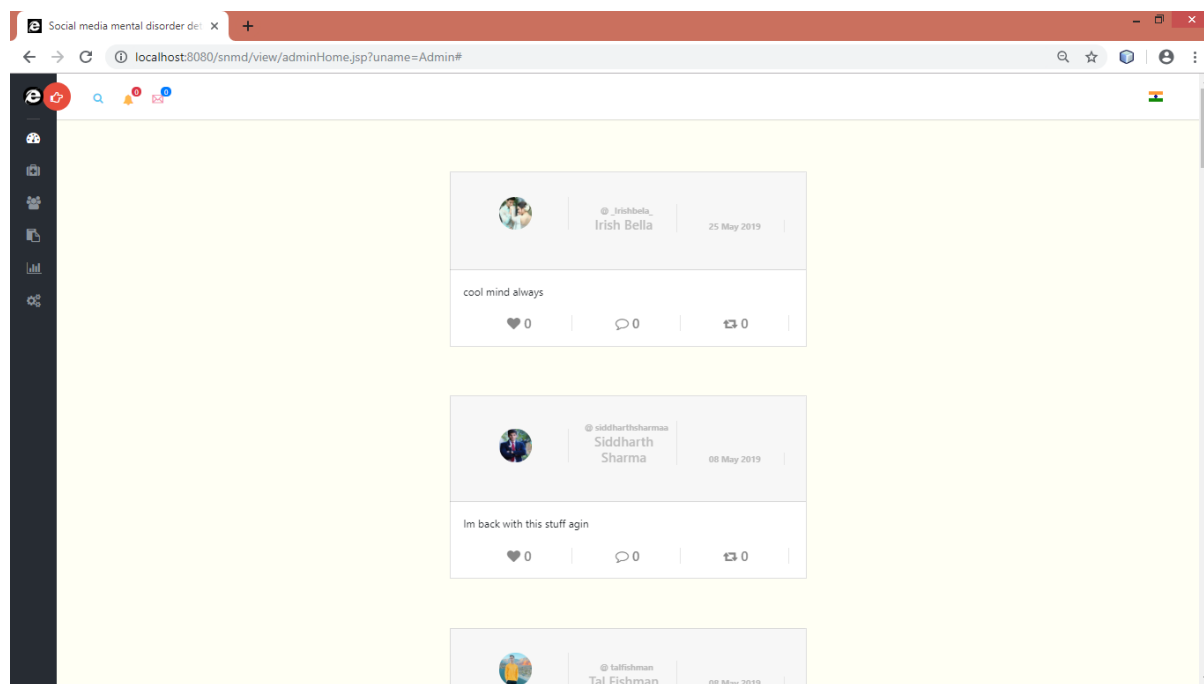


Fig 7.2.6 Admin view tweets

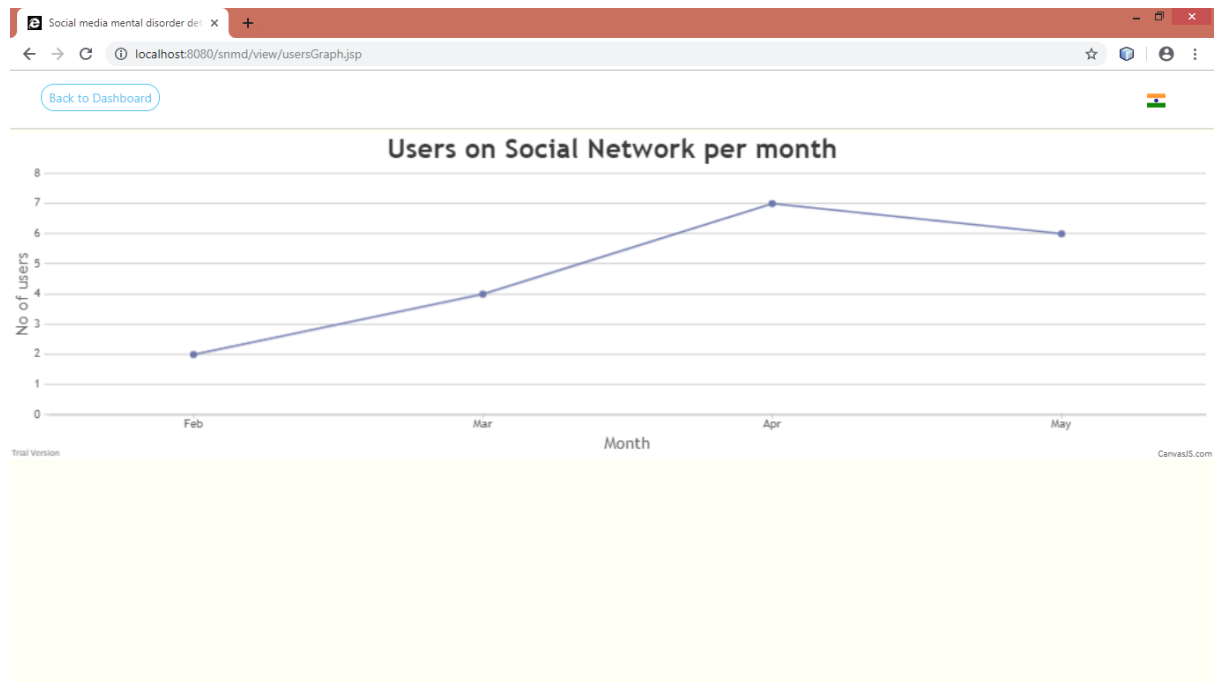


Fig 7.2.7 User graph

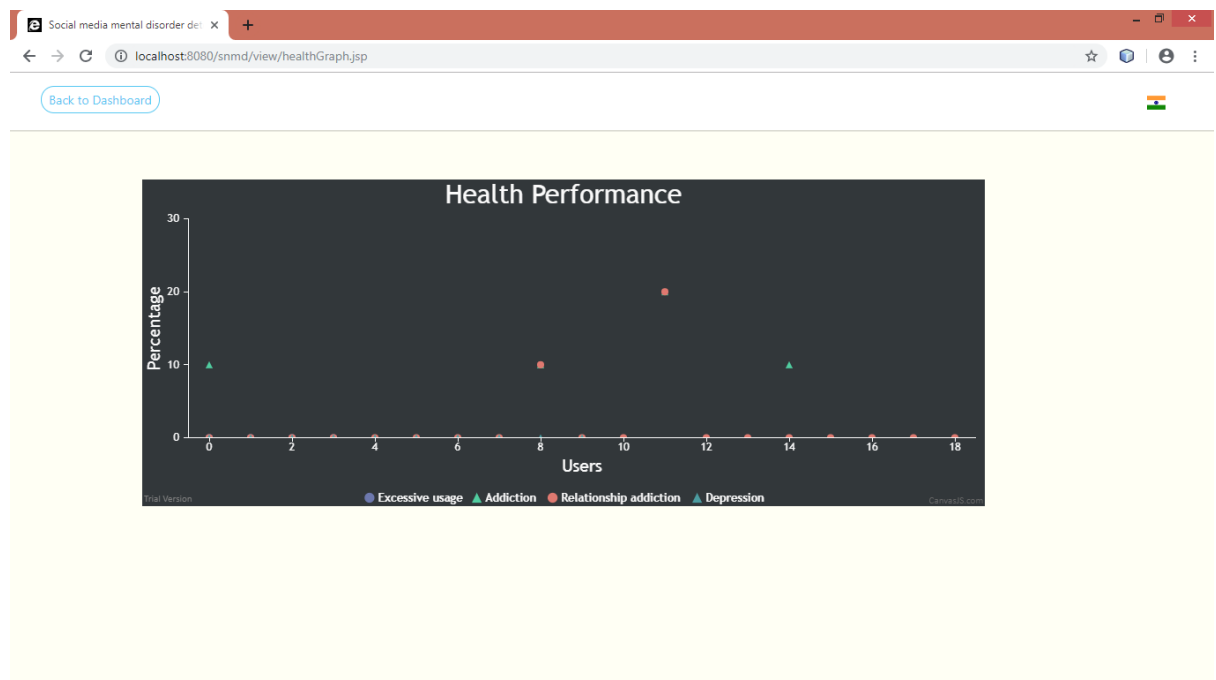


Fig 7.2.8 Health graph

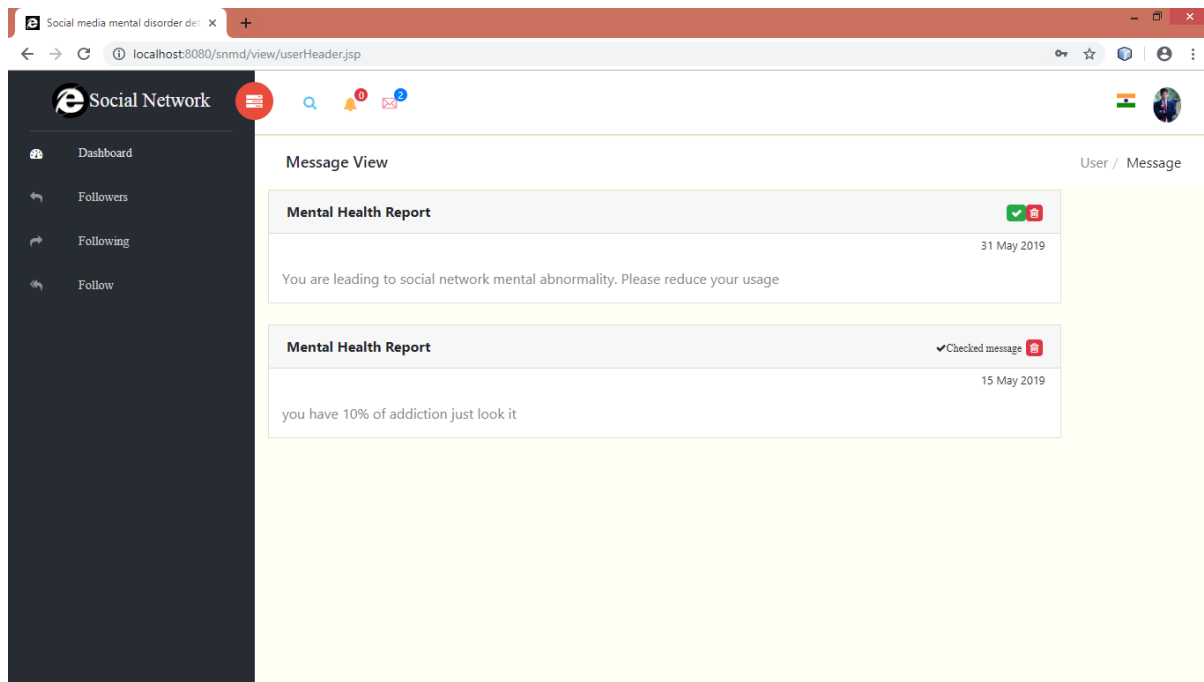


Fig 7.2.9 user view message

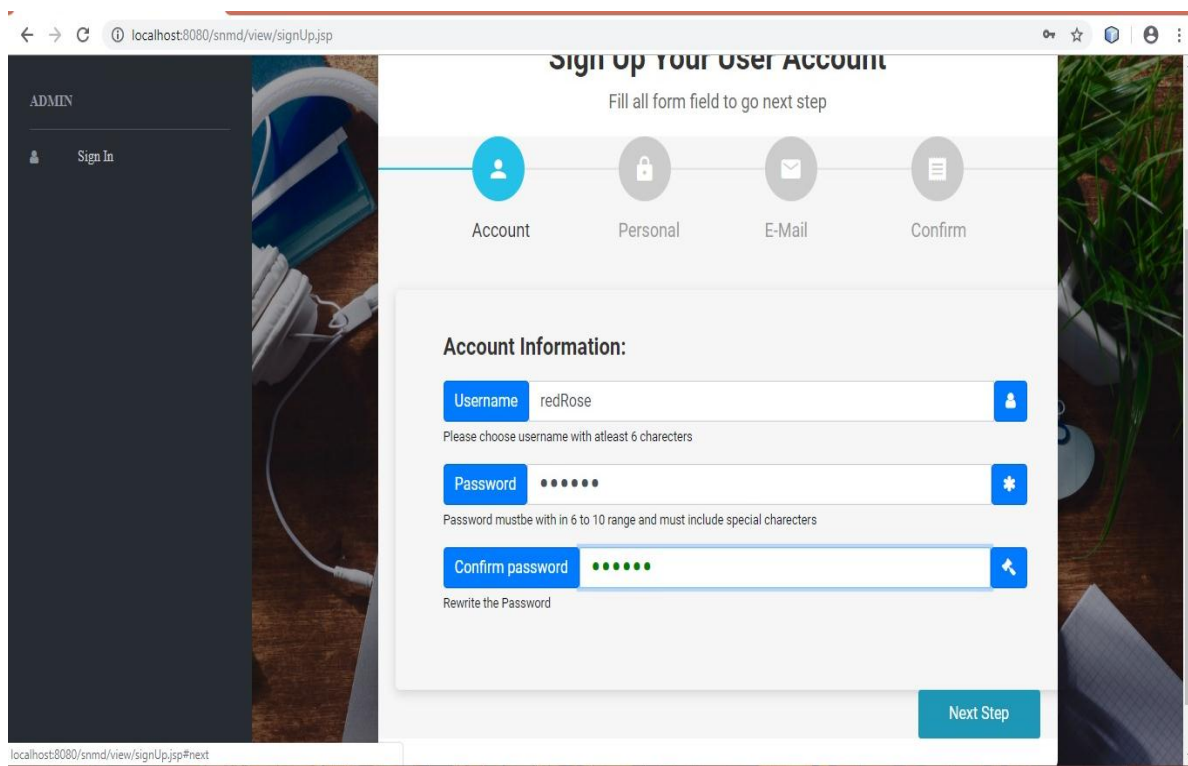


Fig 7.2.10 SignUpPage

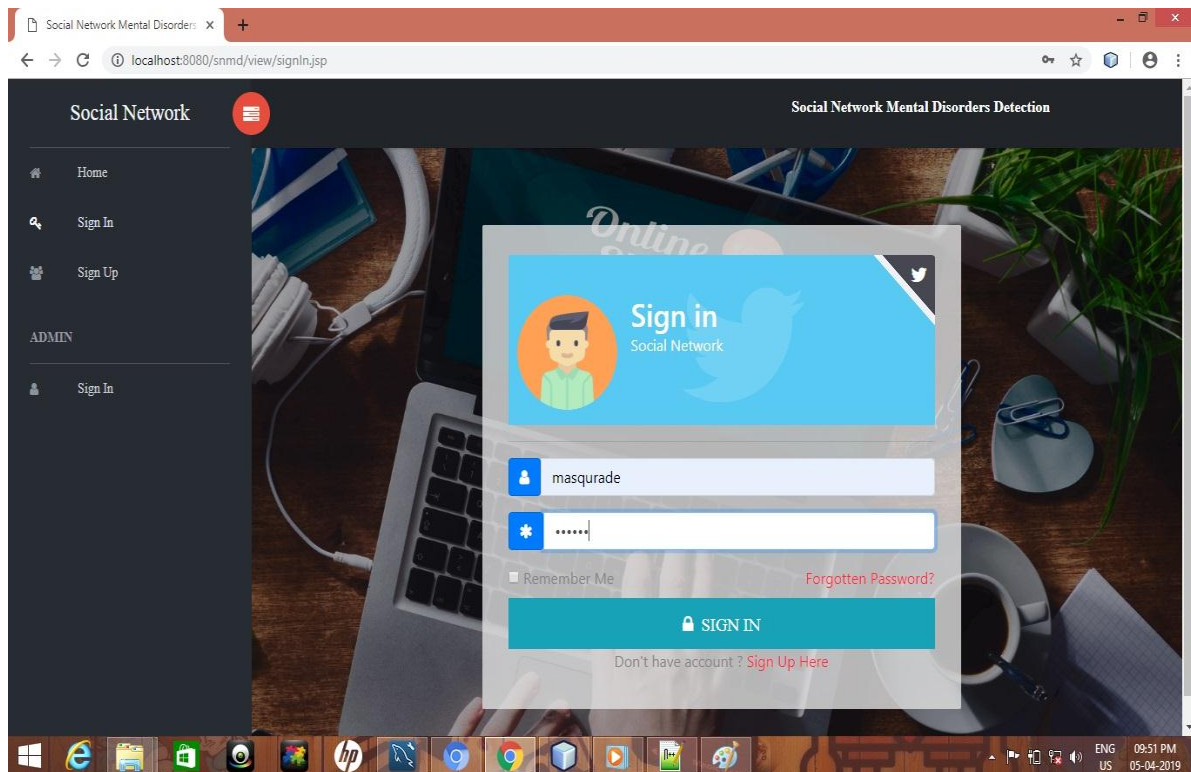


Fig 7.2.11 Sign In Page

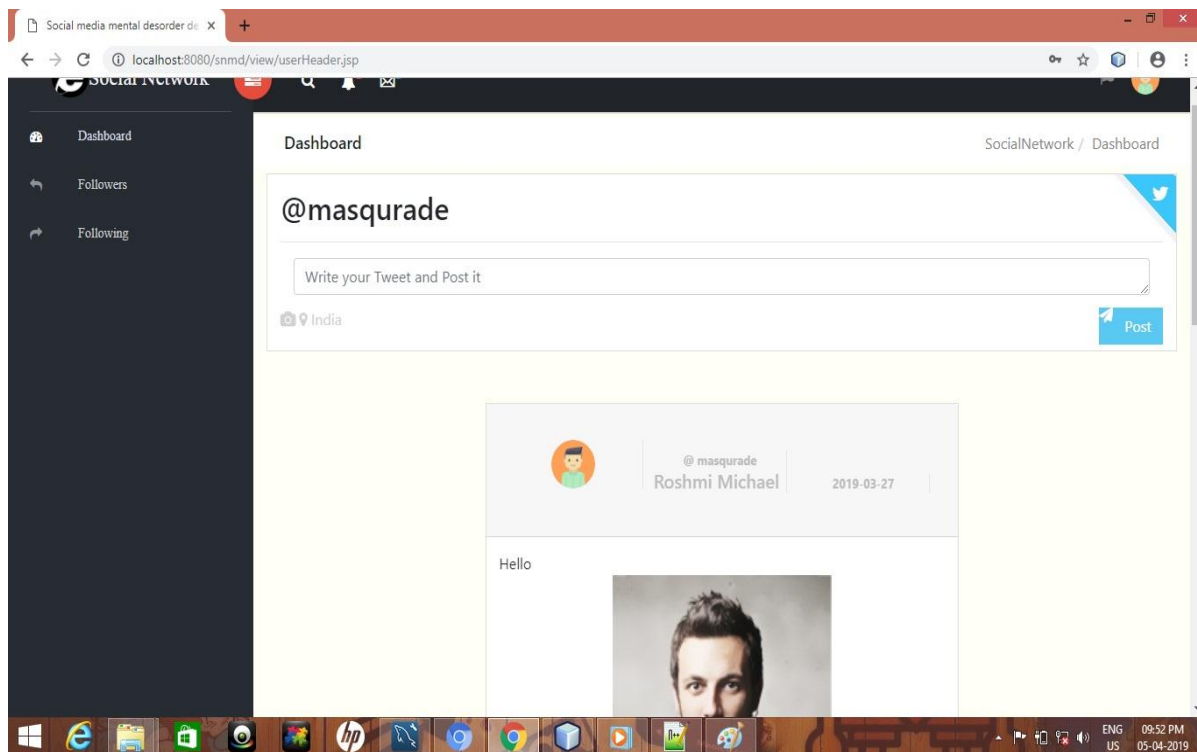


Fig 7.2.12 Dashboard Page

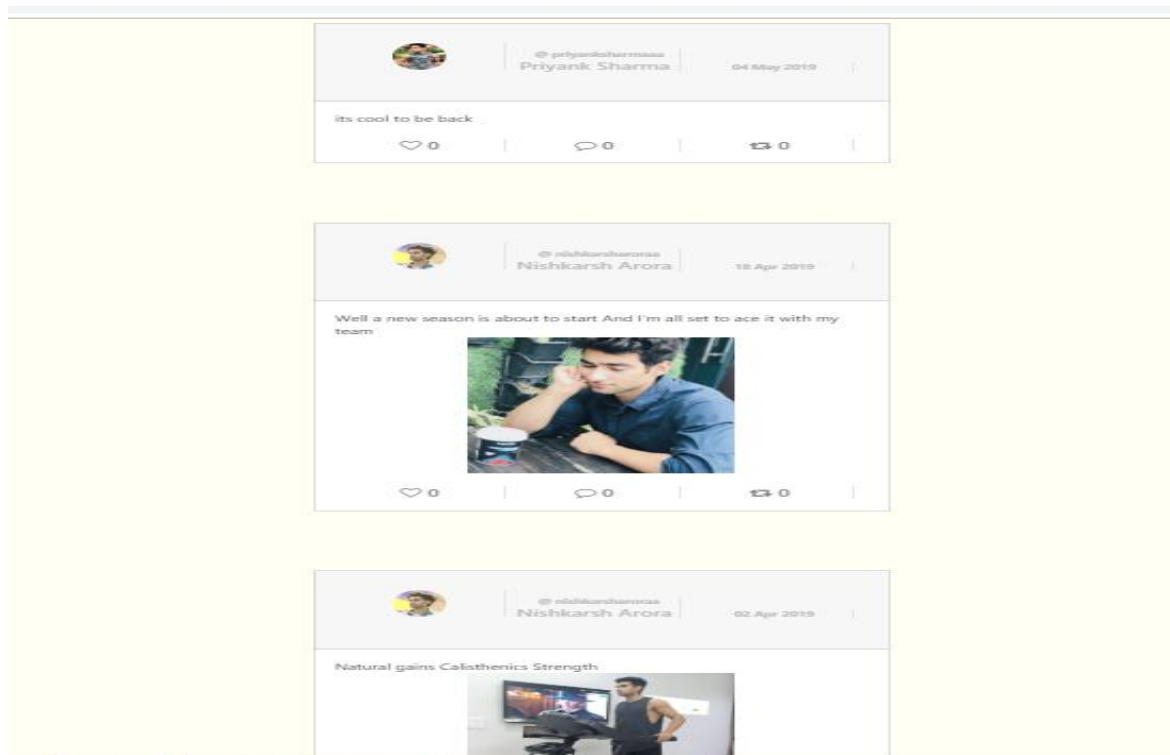


Fig 7.2.13 Tweets

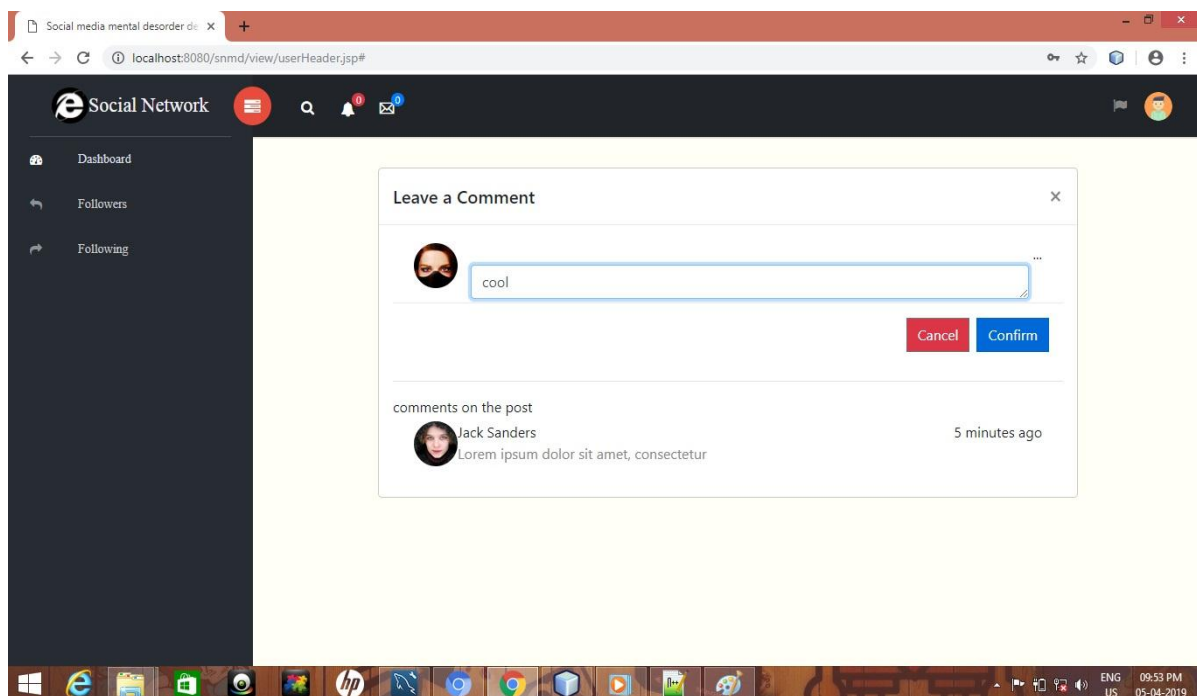
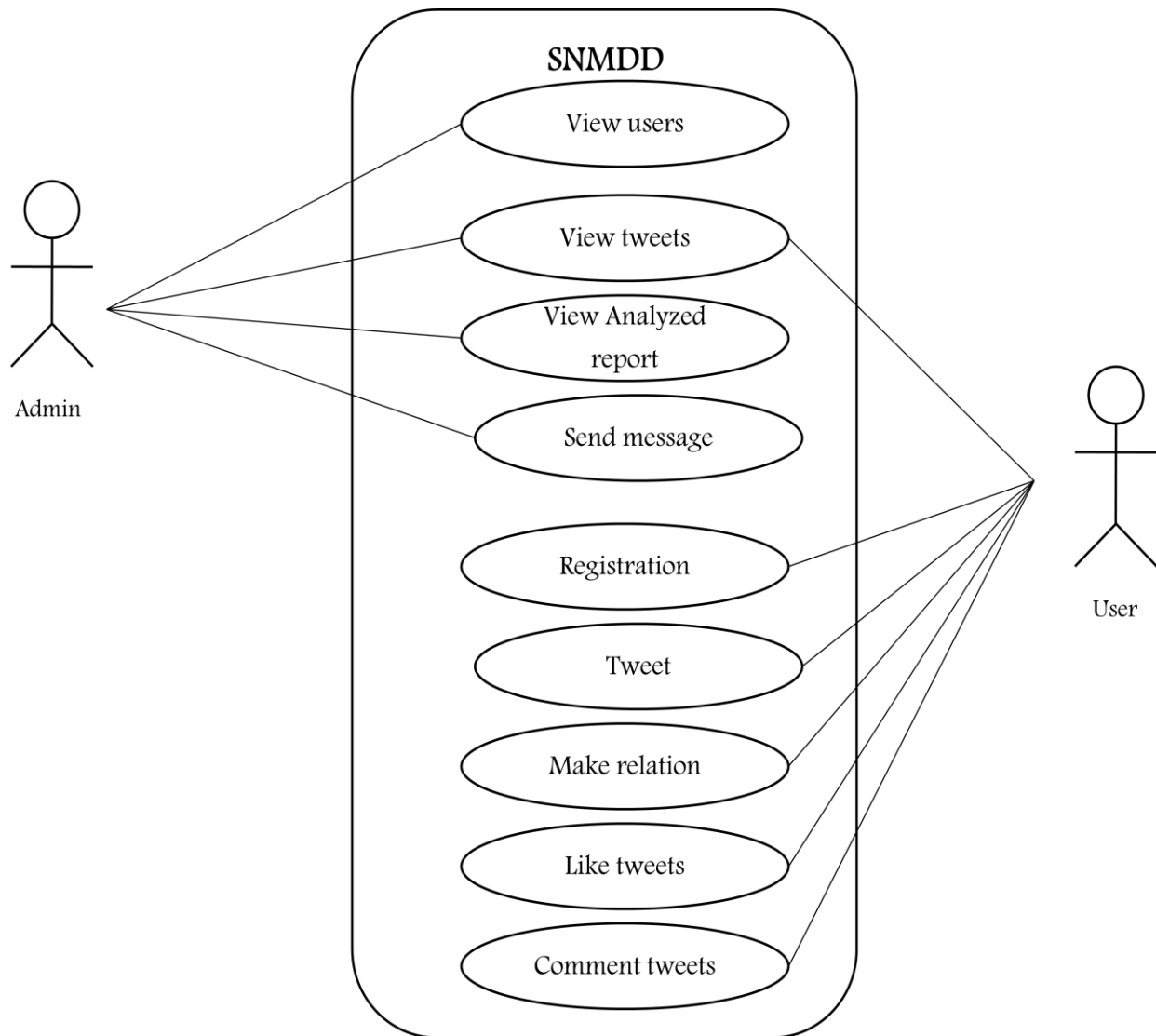


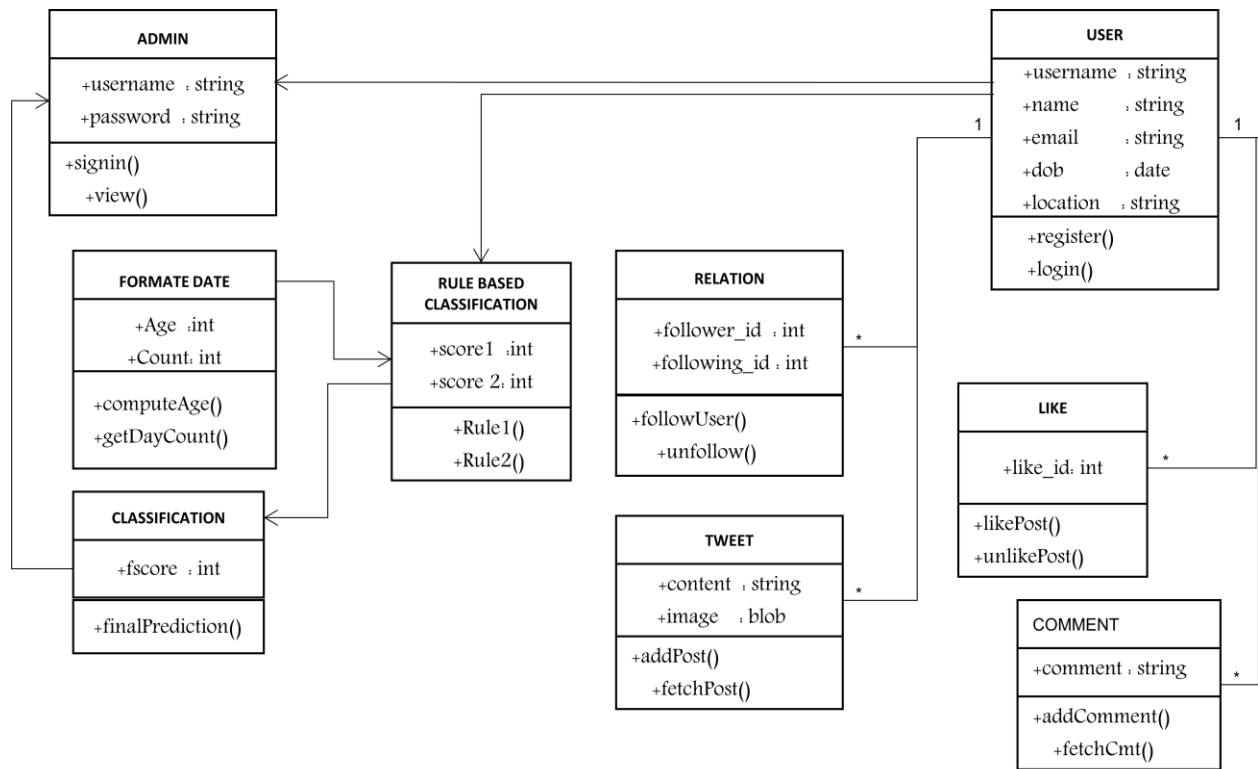
Fig 7.2.14 Comment Page

8. UML DIAGRAMS

8.1 Use case Diagram



8.2 Class Diagram



9.TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test .Software testing can also provide an objective ,independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs. Testing can be stated as the process of validating and verifying that the software product such as meets the requirements that guided its design and development, responds correctly to all kinds of inputs, performs its functions within an acceptable time

9.1 Test Cases

9.1.1 Login

SL.NO	INPUT	EXPECTED RESULT	OBSERVED RESULT
1	Click Login button with invalid username and password.	Display “invalid username and password”	Display “invalid Username and password”
2	Click Login button with each field as blank	Display “data required”	Display “data required”
3	Click Login button with valid username and invalid password.	Display “invalid password”	Display “invalid password”

9.1.2 Registration

SL.NO	INPUT	EXPECTED RESULT	OBSERVED RESULT
1	Click signup button without giving values to some field	Display “data required on blank field”	Display “data required “on blank field
2	Click signup button with invalid email	Display “invalid email address”	Display “invalid email address”
3	Click signup button with invalid password length	Display ”password mismatch”	Display ”password mismatch”

9.2 Test Report

Test Case No	No of Test Cases run	No of Test Cases Successful (%)	Pass/Fail	Expected Result	Actual Result
TC-01	8	100	PASS	The user details should be stored in the DB.	The user details are stored in the DB.
TC-02	10	100	PASS	Output should be displayed and tables are updated	Output is displayed and tables are updated
TC-03	5	90	PASS	Output should be displayed and tables must be updated	Output is displayed and tables are updated
TC-04	7	90	PASS	Output should be displayed and tables must be updated	Output is displayed and tables are updated

10. SYSTEM IMPLEMENTATION

This system is currently being developed by us. The following are modules are the ones that are already developed. Their working is as follows:

Registration: This is mainly for the users to register social network site. This is a demo source site for mining data for the mental disorder detection.

Login: Once the registration is done the user can login to the system. This module will ask the user to provide the username and password. This will be accepted in text. It will be validate whether the details are entered correctly or not. Once the entry is done correctly database will be checked for entry. If the user is authorized it will be directed to homepage.

Home Page: The user is redirected to this page once log in done successfully. From this page now the user can perform operations that the user wishes to perform. The options available are: 1. post 2. follower 3.following.

11. CONCLUSION & FUTURE ENHANCEMENT

In this paper, we make an attempt to automatically identify potential online users with SNMDs. We propose an SNMDD framework that explores various features from data logs of OSNs and a new tensor technique for deriving latent features from multiple OSNs for SNMD detection. This work represents a collaborative effort between computer scientists and mental healthcare researchers to address emerging issues in SNMDs. As for the next step, we plan to study the features extracted from multimedia contents by techniques on NLP and computer vision. We also plan to further explore new issues from the perspective of a social network service provider, e.g., Facebook or Instagram, to improve the well-beings of OSN users without compromising the user engagement.

12. APPENDIX

12.1 Appendix A

12.1.1 Inspection.jsp

```
<% @page import="classificationPrediction.NaiveBayesPrediction"%>
<% @page import="java.io.BufferedWriter"%>
<% @page import="classificationPrediction.ConversionToArff"%>
<% @page import="java.io.FileWriter"%>
<% @page import="java.util.Date"%>
<% @page import="pojo.User_tbl"%>
<% @page import="dao.LoginStatusDao"%>
<% @page import="java.util.Iterator"%>
<% @page import="java.util.ArrayList"%>
<% @page import="java.io.BufferedWriter"%>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<div id="contentBody">
<% String user_id=request.getParameter("user_id");
User_tblob =new User_tbl();
ob.setUser_id(Integer.parseInt(user_id));
ob = UserDao.getInstance().getUserById(ob);
String pattern="dd MMM yyyy";
SimpleDateFormatsimpleDateFormat =new SimpleDateFormat(pattern);
%>
<div class="container"> <!-- Top Navigation -->
<div class="animated fadeIn"><section class="card"><div class="twit-feed bg-dark ">
<div class="corner-ribon blue-ribon"><i class="fafa-twitter"></i></div>
<div class="media"><a href="#">
<%if(ob.getProfile_id()!=null){ %>
```

```

<imgsrc="../../DownloadUserImg?id=<%=ob.getUser_id()%>&time=<%=System.currentTimeMillis()%>" class="align-self-center rounded-circle mr-3" style="width:85px; height:85px;" alt="" />
<% }else{ %>



<% }%></a>

<div class="media-body">

<h2 class="text-white display-6"><%String fullname=ob.getFname()+" "+ob.getLname();
out.print(fullname); %></h2>

<p class="text-light">@ <%= (ob.getUsername())%></p>

<p class="text-light"><% Boolean g=ob.getGender();
if(g==false){ out.print("Female");}
else{ out.print("Male");}%>

Age:<% Date dob=ob.getDob();
Integer age=CalculateAge.getInstance().ageCalc(dob); out.print(age);%>

<% String loc=ob.getLocation();
Location_tbl t= new Location_tbl();
t.setLocName(loc);

t = LocationDao.getInstance().getLocation(t); %>

<i class="<%=t.getLocFlag()%>"></i>

<%= (ob.getLocation())%>SignUp On:

<% String SignUp_on=simpleDateFormat.format(ob.getSignUp_on()); out.print(SignUp_on);%>

</p></div></div></div></section></div>

<% Tweet_tbl tob1=new Tweet_tbl();
tob1.setUser_id(ob.getUser_id());

Integer tcount=TweetDao.getInstance().getTweetCount(tob1);

Relation rob1=new Relation();
rob1.setFollower_id(ob.getUser_id());

Integer flcount=RelationDao.getInstance().getFollowingCount(rob1);

```



```

Relation rob2=new Relation();
rob2.setFollowing_id(ob.getUser_id());
Integer f2count=RelationDao.getInstance().getFollowerCount(rob2);
Retweet_tblrto=new Retweet_tbl();
rtob.setUid(ob.getUser_id());
Integer rtcount=RetweetDao.getInstance().getCount(rtob);
Like_tbllkob=new Like_tbl();
lkob.setLike_userId(ob.getUser_id());
Integer lcount=LikeDao.getInstance().getCount(lkob);
Comment_tblcmtob=new Comment_tbl();
cmtob.setCmt_user(ob.getUser_id());
Integer cmtcount=CommentDao.getInstance().getCount(cmtob);
String c1="",c2="",k=""; Integer keyCount=0,anCount=0;
try{
User_tbl u=new User_tbl();
u.setUser_id(ob.getUser_id());
Negative nob=new Negative();
ArrayList<Negative> alt=NegativeDao.getInstance().getUserPosts(u);
Iterator it=alt.iterator();
while(it.hasNext()){
nob=(Negative) it.next();
c1=nob.getContent();
Negative nob2=new Negative();
ArrayList<Negative> alt2=NegativeDao.getInstance().getNegativeWords();
Iterator it2=alt2.iterator();
while(it2.hasNext()){
nob2=(Negative) it2.next();
k=nob2.getKeyword();

```

```

if((c1.contains(k))) {
    keyCount=keyCount + 1;
} } } catch (Exception e) {
    System.out.print(e); }
%>

<!-- /container --><!-- /tab begins -->

<div class="card-body"><div class="default-tab"><nav>

<div class="navnav-tabs" id="nav-tab" role="tablist">

<a class="nav-item nav-link active" id="nav-home-tab" data-toggle="tab" href="#nav-home"
role="tab" aria-controls="nav-home" aria-selected="true">Overview</a>

<a class="nav-item nav-link" id="nav-profile-tab" data-toggle="tab" href="#nav-profile"
role="tab" aria-controls="nav-profile" aria-selected="false">Anomalies</a>

<a class="nav-item nav-link" id="nav-contact-tab" data-toggle="tab" href="#nav-contact"
role="tab" aria-controls="nav-contact" aria-selected="false">Mental Health</a>

<a class="nav-item nav-link" id="nav-msg-tab" data-toggle="tab" href="#nav-msg" role="tab"
aria-controls="nav-msg" aria-selected="false">Send Message</a>

</div></nav>

<div class="tab-content pl-3 pt-2" id="nav-tabContent">

<div class="tab-pane fade" id="nav-profile" role="tabpanel" aria-labelledby="nav-profile-tab">

<% if(count !=0){

    Integer usage = totalDur / count;

    score1 = RuleBasedClass.getInstance().rule1(usage);

%>

<div class="col-md-4">

<div class="card" style="height:139px;">

<% if(score1==0){ %>

<div class="card-body" style="background-color: #c3e6cb;color: #155724;">

<div class="mx-auto d-block">

<h5 class="text-sm-center mt-2 mb-1"><% out.print("Normal In");%></h5></div></div>

<% }else if(score1>0 && score1<50){anCount=anCount+1; %>

```

```

<div class="card-body" style="background-color: #b8daff;color: #004085;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><%= (score1)%>%</h5>
<div class="location text-sm-center"><%= "Abnormality In"%></div></div></div>
<% }else{ anCount=anCount+1; %>
<div class="card-body" style="background-color: #f5c6cb;color: #491217;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><%= (score1)%>%</h5>
<div class="location text-sm-center"><%= "Abnormality In"%></div></div></div><%= %>
<div class="card-footer text-sm-center">
<strong class="card-title mb-3">Excessive Usage</strong></div></div></div>
<% } %>
<!-- /Excessive use ends -->
<% Integer dayCount = LoginStatusDao.getInstance().getDayCount(lob);
long daydiff = CalculateAge.getInstance().countDay(ob.getSignUp_on());
Integer score2 = RuleBasedClass.getInstance().rule2(daydiff, dayCount);
%>
<div class="col-md-4">
<div class="card" style="height: 139px;">
<% if(score2==0){ %>
<div class="card-body" style="background-color: #c3e6cb;color: #155724;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><%= out.print("Normal In");%></h5>
</div></div>
<% }else if(score2>0 && score2<50){ anCount=anCount+1;%>
<div class="card-body" style="background-color: #b8daff;color: #004085;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><%= (score2)%>%</h5>

```

```

<div class="location text-sm-center"><%= "Abnormality In"%></div></div></div>

<% }else{ anCount=anCount+1; %>

<div class="card-body" style="background-color: #f5c6cb;color: #491217;">

<div class="mx-auto d-block">

<h5 class="text-sm-center mt-2 mb-1"><%= (score2)%>%</h5>

<div class="location text-sm-center"><%= "Abnormality In"%></div></div></div>

<% }%>

<div class="card-footer text-sm-center">

<strong class="card-title mb-3">Addiction</strong></div></div></div>

<!-- /addiction ends -->

<% Relation robj=new Relation();

robj.setFollower_id(Integer.parseInt(user_id));

Integer rCount = RelationDao.getInstance().getRelationCount(robj);

long daydiff1 =CalculateAge.getInstance().countDay(ob.getSignUp_on());

Integer score3=RuleBasedClass.getInstance().rule2(daydiff1,rCount);

%>

<div class="col-md-4">

<div class="card" style="height:139px;">

<% if(score3==0){ %>

<div class="card-body" style="background-color: #c3e6cb;color: #155724;">

<div class="mx-auto d-block">

<h5 class="text-sm-center mt-2 mb-1"><% out.print("Normal In");%></h5></div></div>

<% }else if(score3>0 && score3<50){ anCount=anCount+1; %>

<div class="card-body" style="background-color: #b8daff;color: #004085;">

<div class="mx-auto d-block">

<h5 class="text-sm-center mt-2 mb-1"><%= (score3)%>%</h5>

<div class="location text-sm-center"><%= "Abnormality In"%></div></div></div>

<% }else{ anCount=anCount+1; %>

```

```

<div class="card-body" style="background-color: #f5c6cb;color: #491217;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><%= (score3)%>%</h5>
<div class="location text-sm-center"><%= "Abnormality In"%></div></div></div>
<% }%>
<div class="card-footer text-sm-center">
<strong class="card-title mb-3">Relationship Addiction</strong></div></div></div>
<!-- /Relationship addiction ends -->
<% Integer dprCount = keyCount;
long daydiff4 = CalculateAge.getInstance().countDay(ob.getSignUp_on());
Integer score4 = RuleBasedClass.getInstance().rule2(daydiff4, dprCount);
%>
<div class="col-md-4">
<div class="card" style="height: 139px;">
<% if(score4==0){ %>
<div class="card-body" style="background-color: #c3e6cb;color: #155724;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><% out.print("Normal In");%></h5></div></div>
<% }else if(score4>0 && score4<50){ anCount=anCount+1; %>
<div class="card-body" style="background-color: #b8daff;color: #004085;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><%= (score4)%>%</h5>
<div class="location text-sm-center"><%= "Abnormality In"%></div></div></div>
<% }else{ anCount=anCount+1; %>
<div class="card-body" style="background-color: #f5c6cb;color: #491217;">
<div class="mx-auto d-block">
<h5 class="text-sm-center mt-2 mb-1"><%= (score4)%>%</h5>
<div class="location text-sm-center"><%= "Abnormality In"%></div>

```

```

</div></div><%}%>

<div class="card-footer text-sm-center">

<strong class="card-title mb-3">Depression</strong>

</div></div></div>

<!-- /Depression addiction ends -->

<!-- /Annoying addiction ends -->

<% Update_statusuobj=new Update_status();
uobj.setUserId(Integer.parseInt(user_id));
Integer uCount = UpdateDao.getInstance().getUpdateCount(uobj);
long daydiff3 =CalculateAge.getInstance().countDay(ob.getSignUp_on());
Integer score5=RuleBasedClass.getInstance().rule2(daydiff3,uCount);
%>

<div class="col-md-4">

<div class="card" style="height:139px;">

<% if(score5==0){ %>

<div class="card-body" style="background-color: #c3e6cb;color: #155724;">

<div class="mx-auto d-block">

<h5 class="text-sm-center mt-2 mb-1"><% out.print("Normal In");%></h5></div></div>

<% }else if(score5>0 && score5<50){anCount=anCount+1; %>

<div class="card-body" style="background-color: #b8daff;color: #004085;">

<div class="mx-auto d-block">

<h5 class="text-sm-center mt-2 mb-1"><%= (score5)%>%</h5>

<div class="location text-sm-center"><%= "Abnormality In"%></div>

</div></div>

<% }else{ anCount=anCount+1; %>

<div class="card-body" style="background-color: #f5c6cb;color: #491217;">

<div class="mx-auto d-block">

<h5 class="text-sm-center mt-2 mb-1"><%= (score5)%>%</h5>

```

```

<div class="location text-sm-center"><%= "Abnormality In"%></div>

</div></div>

<% } %>

<div class="card-footer text-sm-center">

<strong class="card-title mb-3">Profile Update</strong>

</div></div></div>

<!-- /Profile Update addiction ends -->

<!-- ----- -->

</div>

<!-- /tab2 ends -->

<div class="tab-pane fade" id="nav-contact" role="tabpanel" aria-labelledby="nav-contact-tab">

<h3>Report</h3>

<% String opfilename="F:/MainProject/Project/snmd/web/data/test.arff";

ConversionToArff.getInstance().convertToArffFile(score2, score2, score3, score5, opfilename);

String predString=NaiveBayesPrediction.getInstance().classification();

%>

<div class="col-lg-6" style="margin: 8vh auto;margin-top: 8vh; margin-right:
auto;margin-bottom: 8vh;margin-left: auto;">

<div class="card" style="width:800px;font-family: Lucida Console;font-size: 12px;">

<div class="card-body card-block">

<div class="row form-group">

<div class="col col-md-8">Name:</div>

<div class="col col-md-4"><% out.print(fullname); %></div>

<!-- ----- -->

<div class="col col-md-8">Age:</div>

<div class="col col-md-4"><%=age%></div>

<!-- ----- -->

<div class="col col-md-8">Gender:</div>

```

```

<div class="col
col-md-4"><%if(g==false){ out.print("Female");}else{out.print("Male");}%></div>

<!-- ----- -->

<div class="col col-md-8">Location:</div>

<div class="col col-md-4"><%= (ob.getLocation())%></div>

<div class="col col-md-8"></div>

<div class="col col-md-4">-</div>

<!-- ----- -->

<div class="col col-md-8">Anomalies Present:</div>

<div class="col col-md-4"><%=anCount%></div>

<div class="col col-md-8"></div>

<div class="col col-md-4">

<div class="col col-md-8">Excessive use :<%=score1%>%</div>

<div class="col col-md-8">Addiction :<%=score2%>%</div>

<div class="col col-md-8">Relationship Addiction :<%=score3%>%</div>

<div class="col col-md-8">Depression :<%=score4%>%</div>

<div class="col col-md-8">Profile abnormality :<%=score5%>%</div></div>

<!-- ----- -->

<div></div>

<div class="col col-md-8">Overall Result:</div>

<div class="col col-md-4"><%=predString%></div></div></div></div></div>

<!-- /addiction ends -->

</div>

<!-- /tab3 ends -->

```


12.1.2 TestData.arff

@relation mental-health-data

@attribute excessive REAL

@attribute addiction REAL

@attribute relationship REAL

@attribute profile REAL

@attribute class {Normal,Abnormal,Leading}

@data

20,20,20,0,?

12.2 Appendix B

12.2.1 Bibliography

- [1] K. Young, M. Pistner, J. O'Mara, and J. Buchanan. Cyber-disorders: The mental health concern for the new millennium. *Cyberpsychol.Behav.*, 1999.
- [2] J. Block. Issues of DSM-V: internet addiction. *American Journal of Psychiatry*, 2008.
- [3] K. Young. Internet addiction: the emergence of a new clinical disorder, *Cyberpsychol. Behav.*, 1998.
- [4] I.-H. Lin, C.-H. Ko, Y.-P. Chang, T.-L. Liu, P.-W. Wang, H.-C. Lin, M.-F. Huang, Y.-C. Yeh, W.-J. Chou, and C.-F. Yen. The association between suicidality and Internet addiction and activities in Taiwanese adolescents. *Compr. Psychiat.*, 2014.
- [5] Y. Baek, Y. Bae, and H. Jang. Social and parasocial relationships on social network sites and their differential relationships with users' psychological well-being. *Cyberpsychol. Behav. Soc. Netw.*, 2013.
- [6] D. La Barbera, F. La Paglia, and R. Valsavoia. Social network and addiction. *Cyberpsychol. Behav.*, 2009.