TRASH KILLER

*A*

*PROJECT*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

By

**K. RESHMI REDDY (1602-17-737-093)**

**ADITYA RAMARAJU (1602-17-737-316)**

**A.SANDEEP KUMAR(1602-17-737-101)**

**Under the guidance of**

**K.SHYAM SUNDER REDDY**

**Assistant Professor**

 **Department of Information Technology**

**Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2019-2020**



**Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Hyderabad-500 031**

**Department of Information Technology** DECLARATION BY THE CANDIDATE



I, **RESHMI, ADITYA, SANDEEP** bearing hall ticket number **1602-17-737-093, 1602-17-737-316, 1602-17-737-101** hereby declare that the project report entitled **“TRASH KILLER”** under the guidance of **K.SHYAM SUNDER REDDY**, Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**K.RESHMI REDDY (1602-17-737-093)**

**ADITYA RAMARAJU (1602-17-737-316)**

1. **SANDEEP KUMAR (1602-17-737-101)**

**Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Hyderabad-500 031**

**Department of Information Technology**



# BONAFIDE CERTIFICATE

This is to certify that the project entitled “**TRASH KILLER”** being submitted by **RESHMI , ADITYA , SANDEEP** bearing **1602-17- 737-093, 1602-17-737-316, 1602-17-737-101** in partial

fulfilment of the requirements for the award of the degree of Bachelor of

Engineering in Information Technology is a record of bonafide work carried out by him/her under my guidance.

**K.SHYAM SUNDER Dr. K. Ram Mohan Rao**

**Assistant Professor HOD, IT**

# ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them.

We would like to take the opportunity to express our humble gratitude to **K.SHYAM SUNDER** sir under whom we executed this project.

We would also use this opportunity to thank our senior P.NARESH KUMAR(1602-16-737- 025). We are grateful to his guidance, and constructive suggestions that helped us in the preparation of this project. His constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.We would like to thank all faculty members and staff of the Department of Information Technology for their generous help in various ways for the completion of this project.

Finally, yet importantly, We would like to express our heartfelt thanks to our **HOD Dr. K. Ram Mohan Rao** Sir and classmates for their help and wishes for the successful completion of this project.

# ABSTRACT

One fundamental issue in today On-line Social Networks (OSNs) is to give users the ability to control the messages posted on their own private space to avoid that unwanted content is displayed. Up to now OSNs provide little support to this requirement. To fill the gap, in this project, we propose a system allowing OSN users to have a direct control on the messages posted on their walls. This is achieved through a flexible rule-based system, that allows users to customize the filtering criteria to be applied to their walls soft classifier automatically labeling messages in support of content-based filtering.

**Table of Contents**

[1. INTRODUCTION 8](#_Toc18377)

[1.1 Application Description 8](#_Toc31750)

[2.LITERATURE SURVEY 9](#_Toc6389)

[2.1 Introduction to Java: 9](#_Toc24290)

[2.2Technologies Used 11](#_Toc6615)

[2.2.1 HTML 11](#_Toc23028)

[2.2.2 JavaScript 15](#_Toc16666)

[2.2.3 SQL 17](#_Toc26084)

[2.3 SERVLETS 18](#_Toc14021)

[2.4 Java Server Pages (JSP) 19](#_Toc23699)

[2.5 Domain Description: 20](#_Toc24391)

[2.6 Data Mining Uses 21](#_Toc642)

[3.REQUIREMENT ANALYSIS 21](#_Toc16320)

[3.1 H/W System Configuration:- 21](#_Toc22630)

[3.2 S/W System Configuration:- 21](#_Toc3521)

[3.3 FunctiOnal Requirements 22](#_Toc24188)

[3.4 Non Functional Requirements 23](#_Toc3541)

[3.5 Existing System 24](#_Toc32601)

[3.6 Proposed System 24](#_Toc16276)

[4.SYSTEM DESIGN 25](#_Toc17984)

[4.1 UML Diagrams: 25](#_Toc2833)

[4.2 Use-Case diagram: 29](#_Toc25939)

[4.3 Class Diagram: 30](#_Toc729)

[4.4 Sequence Diagram 31](#_Toc29205)

[4.5 Activity Diagram: 32](#_Toc9959)

[4.6 Component 36](#_Toc13070)

[5.IMPLEMENTATION 36](#_Toc14389)

[5.1Modules: 36](#_Toc28570)

[5.1.1 Filtering rules 36](#_Toc27089)

[5.1.2 Online setup assistant for FRs thresholds: 37](#_Toc31527)

[5.1.3 Blacklists: 37](#_Toc25506)

[5.2 Filter Wall Architecture: 38](#_Toc16239)

[5.2.1 Content Based Filtering: 39](#_Toc27706)

[5.2.2 Policy Based Personalization of OSN Contents: 39](#_Toc25969)

[5.3 Short Text Classifier: 41](#_Toc31346)

[5.3.1 Text Representation: 42](#_Toc31538)

[5.4 Machine Learning-based Classification: 43](#_Toc13468)

[6.Coding 44](#_Toc9359)

[6.1 Databasecon.java 44](#_Toc21137)

[6.2 UserRegistration.jsp 45](#_Toc6061)

[6.3 UserLogin.jsp 53](#_Toc16614)

[6.4 Filtertext.jsp 55](#_Toc27074)

[7. TESTING 56](#_Toc5898)

[7.1 TYPES OF TESTS 56](#_Toc5965)

[7.1.1 Unit testing 56](#_Toc16160)

[7.1.2 Integration testing 56](#_Toc19199)

[7.1.3 Functional testing 57](#_Toc26060)

[7.1.4 System Testing 57](#_Toc16598)

[7.1.5 White Box Testing 57](#_Toc14673)

[7.1.6 Black Box Testing 58](#_Toc20293)

[8.RESULTS 60](#_Toc31407)

[8.1 RIOUS SNAPSHOTS 60](#_Toc11862)

[8.1.1 USER LOGIN PAGE: 60](#_Toc14497)

[8.1.2 New user Registration 61](#_Toc5421)

[8.1.3 Update Status page 61](#_Toc2994)

[8.1.4 Update status page to give the symptoms 62](#_Toc18520)

[8.1.5 Selecting the image from data base 63](#_Toc9147)

[8.1.6 Admin login page 64](#_Toc18371)

[9.GitHub links : 64](#_Toc14982)

[10.CONCLUSION AND FUTURE WORK 65](#_Toc345)

[11.References 65](#_Toc23669)

# 1. INTRODUCTION

## 1.1 Application Description

Online Social Networks (OSNs) are today one of the most popular interactive medium to communicate, share, and disseminate a considerable amount of human life information. Daily and continuous communications imply the exchange of several types of content, including free text, image, audio, and video data. According to Facebook statistics1 average user creates 90 pieces of content each month, whereas more than 30 billion pieces of content (web links, news stories, blog posts, notes, photo albums, etc.) are shared each month.

The huge and dynamic character of these data creates the premise for the employment of web content mining strategies aimed to automatically discover useful information dormant within the data. They are instrumental to provide an active support in complex and sophisticated tasks involved in OSN management, such as for instance access control or information filtering. Information filtering has been greatly explored for what concerns textual documents and, more recently, web content. However, the aim of the majority of these proposals is mainly to provide users a classification mechanism to avoid they are overwhelmed by useless data. In OSNs, information filtering can also be used for a different, more sensitive, purpose. This is due to the fact that in OSNs there is the possibility of posting or commenting other posts on particular public/private areas, called in general walls. Information filtering can therefore be used to give users the ability to automatically control the messages written on their own walls, by filtering out unwanted messages.

# **2.LITERATURE [SURVEY](http://www.blurtit.com/q876299.html)**

Literature [survey](http://www.blurtit.com/q876299.html) is the most important step in software development process. Before we are going to developing the tool it is important to determine the factors like time,econamy and srength .after satisfy thease actions we are mainly concentraing on the operating syste we used for this project we are used windows7 operating system and this project is developed under the platform java .the programers uses some hardware deevices by using thease devices we can easily develop our proposed system.

## 2.1 Introduction to Java:

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (\*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name \*7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

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

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

## 2.2Technologies Used

### 2.2.1 HTML

HTML, an initialism of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors.

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML 1.0 and later) or its form descended directly from SGML

**Hyper Text Markup Language**

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags:

<! -- --> specifies comments

<A>……….</A> Creates hypertext links

<B>……….</B> Formats text as bold

<BIG>……….</BIG> Formats text in large font.

<BODY>…</BODY> Contains all tags and text in the HTML document

<CENTER>...</CENTER> Creates text

<DD>…</DD> Definition of a term

<DL>...</DL> Creates definition list

<FONT>…</FONT> Formats text with a particular font

<FORM>...</FORM> Encloses a fill-out form

<FRAME>...</FRAME> Defines a particular frame in a set of frames

<H#>…</H#> Creates headings of different levels( 1 – 6 )

<HEAD>...</HEAD> Contains tags that specify information about a document

<HR>...</HR> Creates a horizontal rule

<HTML>…</HTML> Contains all other HTML tags

<META>...</META> Provides meta-information about a document

<SCRIPT>…</SCRIPT> Contains client-side or server-side script

<TABLE>…</TABLE> Creates a table

<TD>…</TD> Indicates table data in a table

<TR>…</TR> Designates a table row

<TH>…</TH> Creates a heading in a table

**Attributes**

The attributes of an element are name-value pairs, separated by "=", and written within the start label of an element, after the element's name. The value should be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML).Leaving attribute values unquoted is considered unsafe.

Most elements take any of several common attributes: id, class, style and title. Most also take language-related attributes: lang and dir.

The id attribute provides a document-wide unique identifier for an element. This can be used by stylesheets to provide presentational properties, by browsers to focus attention on the specific element or by scripts to alter the contents or presentation of an element. The class attribute provides a way of classifying similar elements for presentation purposes. For example, an HTML document (or a set of documents) may use the designation class="notation" to indicate that all elements with this class value are all subordinate to the main text of the document (or documents). Such notation classes of elements might be gathered together and presented as footnotes on a page, rather than appearing in the place where they appear in the source HTML.

An author may use the style non-attributal codes presentational properties to a particular element. It is considered better practice to use an element’s son- id page and select the element with a stylesheet, though sometimes this can be too cumbersome for a simple ad hoc application of styled properties. The title is used to attach subtextual explanation to an element. In most browsers this title attribute is displayed as what is often referred to as a tooltip. The generic inline span element can be used to demonstrate these various non-attributes.

The preceding displays as HTML (pointing the cursor at the abbreviation should display the title text in most browsers).

**Advantages**

* A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
* HTML is platform independent.
* HTML tags are not case-sensitive.

### **2.2.2 JavaScript**

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser’s display accordingly

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags

<SCRIPTS>.. </SCRIPT>.

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

Here are a few things we can do with JavaScript:

* Validate the contents of a form and make calculations.
* Add scrolling or changing messages to the Browser’s status line.
* Animate images or rotate images that change when we move the mouse over them.
* Detect the browser in use and display different content for different browsers.
* Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

**JavaScript Vs Java**

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

* Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
* While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

Advantages

* JavaScript can be used for Sever-side and Client-side scripting.
* It is more flexible than VBScript.
* JavaScript is the default scripting languages at Client-side since all the browsers supports it.

**Java Database Connectivity**

What Is JDBC?

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

What Does JDBC Do?

Simply put, JDBC makes it possible to do three things:

* Establish a connection with a database
* Send SQL statements
* Process the results.

### 2.2.3 SQL

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

In the relational model, data is stored in structures called relations or tables*.*

SQL statements are issued for the purpose of:

**Data definition:** Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes).

**Data manipulation:** Used to manipulate the data within those schema objects (DML Inserting, Updating, Deleting the data, and Querying the Database).

A schema is a collection of database objects that can include: tables, views, indexes and sequences

List of SQL statements that can be issued against an Oracle database schema are:

* **ALTER** - Change an existing table, view or index definition (DDL)
* **AUDIT** - Track the changes made to a table (DDL)
* **COMMENT** - Add a comment to a table or column in a table (DDL)
* **COMMIT** - Make all recent changes permanent (DML - transactional)
* **CREATE** - Create new database objects such as tables or views (DDL)
* **DELETE**- Delete rows from a database table (DML)
* **DROP** - Drop a database object such as a table, view or index (DDL)
* **GRANT** - Allow another user to access database objects such as tables or views (DDL)
* **INSERT** - Insert new data into a database table (DML)
* **No AUDIT** - Turn off the auditing function (DDL)
* **REVOKE** - Disallow a user access to database objects such as tables and views (DDL)
* **ROLLBACK** - Undo any recent changes to the database (DML - Transactional)
* **SELECT** - Retrieve data from a database table (DML)
* **TRUNCATE** - Delete all rows from a database table (can not be rolled back) (DML)
* **UPDATE**- Change the values of some data items in a database table (DML)

## 2.3 SERVLETS

**Introduction**

The Java web server is JavaSoft's own web Server. The Java web server is just a part of a larger framework, intended to provide you not just with a web server, but also with tools. To build customized network servers for any Internet or Intranet client/server system. Servlets are to a web server, how applets are to the browser.

**About Servlets**

Servlets provide a Java-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side - object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform independent, dynamically loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

For example, an HTTP Servlets can be used to generate dynamic HTML content. When you use Servlets to do dynamic content you get the following advantages:

* They’re faster and cleaner than CGI scripts
* They use a standard API (the Servlets API)
* They provide all the advantages of Java (run on a variety of servers without needing to be rewritten).

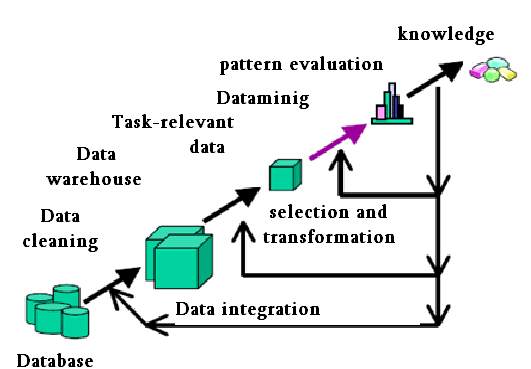
## 2.4 Java Server Pages (JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model .The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches; it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other’s work.

## **2.5 Domain Description:**

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods (algorithms that improve their performance automatically through experience, such as neural networks or decision trees). Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction.

Data mining can be performed on data represented in quantitative, textual, or multimedia forms. Data mining applications can use a variety of parameters to examine the data. They include association (patterns where one event is connected to another event, such as purchasing a pen and purchasing paper), sequence or path analysis (patterns where one event leads to another event, such as the birth of a child and purchasing diapers), classification (identification of new patterns, such as coincidences between duct tape purchases and plastic sheeting purchases), clustering (finding and visually documenting groups of previously unknown facts, such as geographic location and brand preferences), and forecasting (discovering patterns from which one can make reasonable predictions regarding future activities, such as the prediction that people who join an athletic club may take exercise classes)



**Figure 2.5 knowledge discovery process**

## **2.6 Data Mining Uses**

Data mining is used for a variety of purposes in both the private and public sectors.

* Industries such as banking, insurance, medicine, and retailing commonly use data mining to reduce costs, enhance research, and increase sales. For example, the insurance and banking industries use data mining applications to detect fraud and assist in risk assessment (e.g., credit scoring).
* Using customer data collected over several years, companies can develop models that predict whether a customer is a good credit risk, or whether an accident claim may be fraudulent and should be investigated more closely.

The medical community sometimes uses data mining to help predict the effectiveness of a procedure or medicine.

Pharmaceutical firms use data mining of chemical compounds and genetic material to help guide research on new treatments for diseases.

# **3.REQUIREMENT ANALYSIS**

# **3.1 H/W System Configuration:-**

Processor : i3

Ram : 1 GB

Hard disk : 80 GB

# **3.2 S/W System Configuration:-**

Operating System : Windows XP/7/8

Technology : jdbc, jsp, html, JavaScript

Database : MY SQL

Web Server : Apache Tomcat 7.0

Java : j.d.k 1.7

## 3.3 FunctiOnal Requirements

**Inputs:**

**a. Server:**

In this we will enter ip address for client system. In practice, the mechanism can be executed either by the end-user or by a centralized server. In the former case, the source node is responsible for collecting cost reports from individual SRs and computing the path vector and payments. Such an implementation is suitable for an end-user of sufficient power computation capability. A centralized server implementation is more appropriate for resource-constrained users.

**Operation:**

Routers are small physical devices that join multiple networks together. Technically, a router is a Layer3 gateway device, meaning that it connects two or more networks and that the router operates at the network layer of the OSI model. This module is used for removing the Hackers among the nodes. This module consists of a textbox field which is used to enter the destination IP address. After that select the node. Now click on start button in server module. With this the data is transferred to the destination through the selected node. To avoid jamming in that node click on particular avoid jammer node. If we want to know the time elapsed for sending the file click on Time elapsed button in Router .Now go to client module there you will get a message saying that file received.Defination of Hackers: a person who illegally gains access to and sometimes tampers with information in a computer system

**Output:**

By exaggerating its capacity, node may relay a larger fraction of traffic and receive a higher payment, leading to a possibly higher profit. To maintain the overall truthfulness of the mechanism, misreporting of capacity information should be eliminated by design.

## 3.4 Non Functional Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of users of the existing system to give the requirement specifications because they are the people who finally use the system.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system.
* The system should be accurate.
* Te system should be better than existing system.
* **Portability:** It should run on specified platforms successfully. To achieve this we should test the product on all platforms before launching the product. If our project runs successfully on different platforms then our system is portable in nature.
* **Reliability:** The system should perform its intended functions under specified conditions. If our system satisfies all the specified conditions then it is Reliable in nature.
* **Reusability:** The system should be extremely reusable as a whole or part. Make the system modularize and make sure that modules are loosely coupled. This project is having reusability nature because we can reuse whole or part of this project on other systems.
* **Robustness:** The system on the whole should be robust enough to perform well under different circumstances without any inconsistencies.
* **Testability:** The product of a given development phase should satisfy the conditions imposed at the start of that phase.
* **Usability:** It should be perfect and comfortable for users to work.
* **Security:** The system is completely based on the security. This system will provide security base on the password

## 3.5 Existing System

We believe that this is a key OSN service that has not been provided so far. Indeed, today OSNs provide very little support to prevent unwanted messages on user walls. For example, Face book allows users to state who is allowed to insert messages in their walls (i.e., friends, friends of friends, or defined groups of friends). However, no content-based preferences are supported and therefore it is not possible to prevent undesired messages, such as political or vulgar ones, no matter of the user who posts them. Providing this service is not only a matter of using previously defined web content mining techniques for a different application, rather it requires to design ad-hoc classification strategies. This is because wall messages are Constituted by short text for which traditional classification Methods have serious limitations since short texts do not Provide sufficient word occurrences.

## 3.6 Proposed System

The aim of the present work is therefore to propose and experimentally evaluate an automated system, called Filtered Wall (F1W), able to filter unwanted messages from OSN user walls. We exploit Machine Learning (ML) text categorization techniques to automatically assign with each short text message a set of categories based on its content. The major efforts in building a robust short text classifier are concentrated in the extraction and selection of a set of characterizing and discriminate features. The solutions investigated in this paper are an extension of those adopted in a previous work by us from which we inherit the learning model and the elicitation procedure for generating pre-classified data. The original set of features, derived from endogenous properties of short texts, is enlarged here including exogenous knowledge related to the context from which the messages originate. As far as the learning model is concerned, we confirm in the current paper the use of neural learning which is today recognized

# 4.SYSTEM DESIGN

## 4.1 UML Diagrams:

UML is a method for describing the system architecture in detail using the blueprint.

UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a very important part of developing objects oriented software and the software development process.

UML uses mostly graphical notations to express the design of software projects.

Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

**Definition:**

UML is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of the software system.

**UML is a language:**

It will provide vocabulary and rules for communications and function on conceptual and physical representation. So it is modeling language.

**UML Specifying:**

Specifying means building models that are precise, unambiguous and complete. In particular, the UML address the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

**UML Visualization:**

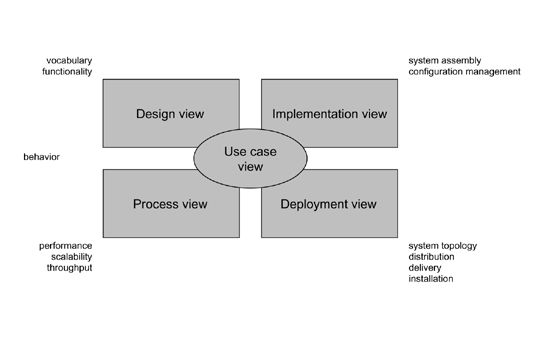
The UML includes both graphical and textual representation. It makes easy to visualize the system and for better understanding.

**UML Constructing:**

UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

**UML Documenting:**

UML provides variety of documents in addition raw executable codes.



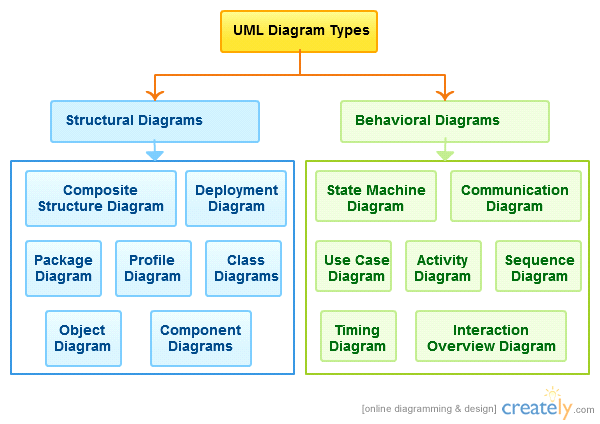
**Figure 4.1 Modeling a System Architecture using views of UML**

The use case view of a system encompasses the use cases that describe the behavior of the system as seen by its end users, analysts, and testers.

The *design view* of a system encompasses the classes, interfaces, and collaborations that form the vocabulary of the problem and its solution.

The *process view* of a system encompasses the threads and processes that form the system's concurrency and synchronization mechanisms.

The *implementation view* of a system encompasses the components and files that are used to assemble and release the physical system.The*deployment view* of a system encompasses the nodes that form the system's hardware topology on which the system executes.



**Figure 4.2 Types of UML Diagrams**

**Uses of UML:**

The UML is intended primarily for software intensive systems. It has been used effectively for such domain as

Enterprise Information System

Banking and Financial Services

Telecommunications

Transportation

Defense/Aerosp

Retails

Medical Electronics

Scientific Fields

Distributed Web

**Building blocks of UML:**

The vocabulary of the UML encompasses 3 kinds of building blocks

Things

Relationships

Diagrams

**Things:**

Things are the data abstractions that are first class citizens in a model. Things are of 4 types

Structural Things, Behavioral Things, Grouping Things, A notational Things

**Relationships:**

Relationships tie the things together. Relationships in the UML are

Dependency, Association, Generalization, Specialization

**UML Diagrams:**

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships).

There are two types of diagrams, they are:

Structural and Behavioral Diagrams

**Structural Diagrams:**

The UML‘s four structural diagrams exist to visualize, specify, construct and document the static aspects of a system. icon View the static parts of a system using one of the following diagrams. Structural diagrams consists of Class Diagram, Object Diagram, Component Diagram, Deployment Diagram.

**Behavioral Diagrams :**

The UML’s five behavioral diagrams are used to visualize, specify, construct, and document the dynamic aspects of a system. The UML’s behavioral diagrams are roughly organized around the major ways which can model the dynamics of a system.

Behavioral diagrams consists of

Use case Diagram, Sequence Diagram, Collaboration Diagram, State chart Diagram, Activity Diagram

## 4.2 Use-Case diagram:

A use case is a set of scenarios that describing an interaction between a user and a system.  A use case diagram displays the relationship among actors and use cases.  The two main components of a use case diagram are use cases and actors.

An actor is represents a user or another system that will interact with the system you are modeling.  A use case is an external view of the system that represents some action the user might perform in order to complete a task.

**Contents:**

* Use cases
* Actors
* Dependency, Generalization, and association relationships
* System boundary



## 4.3 Class Diagram:

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Class diagrams are arguably the most used UML diagram type. It is the main building block of any object oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class. In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many classes related classes are grouped together to to create class diagrams. Different relationships between diagrams are show by different types of Arrows. Below is a image of a class diagram. Follow the link for more class diagram examples.



## 4.4 Sequence Diagram

Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are show as arrows. This article explains the purpose and the basics of Sequence diagrams.

****

## 4.5 Activity Diagram:

Activity diagrams describe the workflow behavior of a system.  Activity diagrams are similar to state diagrams because activities are the state of doing something.  The diagrams describe the state of activities by showing the sequence of activities performed.  Activity diagrams can show activities that are conditional or parallel.

**How to Draw: Activity Diagrams**

Activity diagrams show the flow of activities through the system.  Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities.  A fork is used when multiple activities are occurring at the same time.  The diagram below shows a fork after activity1.  This indicates that both activity2 and activity3 are occurring at the same time.  After activity2 there is a branch.  The branch describes what activities will take place based on a set of conditions.  All branches at some point are followed by a merge to indicate the end of the conditional behavior started by that branch.   After the merge all of the parallel activities must be combined by a join before transitioning into the final activity state.

**When to Use: Activity Diagrams**

Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams.  The main reason to use activity diagrams is to model the workflow behind the system being designed.  Activity Diagrams are also useful for: analyzing a use case by describing what actions need to take place and when they should occur; describing a complicated sequential algorithm; and modeling applications with parallel processes.

User Activity diagram

****

Admin Activity diagram:

****

## 4.6 Component

****

# 5.IMPLEMENTATION

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

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

## **5.1Modules**:

### 5.1.1 Filtering rules

In defining the language for FRs specification, we consider three main issues that, in our opinion, should affect a message filtering decision. First of all, in OSNs like in everyday life, the same message may have different meanings and relevance based on who writes it. As a consequence, FRs should allow users to state constraints on message creators. Creators on which a FR applies can be selected on the basis of several different criteria; one of the most relevant is by imposing conditions on their profile’s attributes. In such a way it is, for instance, possible to define rules applying only to young creators or to creators with a given religious/political view. Given the social network scenario, creators may also be identified by exploiting information on their social graph. This implies to state conditions on type, depth and trust values of the relationship(s) creators should be involved in order to apply them the specified rules. All these options are formalizedby the notion of creator specification, defined as follows.

### **5.1.2 Online setup assistant for FRs thresholds**:

In Online Setup Assistant (OSA) presents the user with a set of messages selected from the dataset discussed in Section VI-A. For each message, the user tells the system the decision to accept or reject the message. The collection and processing of user decisions on an adequate set of messages distributed over all the classes allows to compute customized thresholds representing the user attitude in accepting or rejecting certain contents. Such messages are selected according to the following process. A certain amount of non neutral messages taken from a fraction of the dataset and not belonging to the training/test sets, are classified by the ML in order to have, for each message, the second level class membership values.

### **5.1.3 Blacklists**:

A further component of our system is a BL mechanism to avoid messages from undesired creators, independent from their contents. BLs are directly managed by the system, which should be able to determine who are the users to be inserted in the BL and decide when users retention in the BL is finished. To enhance flexibility, such informationare given to the system through a set of rules, hereafter called BL rules. Such rules are not defined by the SNM, therefore they are not meant as general high level directives to be applied to the whole community. Rather, we decide to let the users themselves, i.e., the wall’s owners to specify BL rules regulating who has to be banned from their walls and for how long. Therefore, a user might be banned from a wall, by, at the same time, being able to post in other walls.

Similar to FRs, our BL rules make the wall owner able to identify users to be blocked according to their profiles as well as their relationships in the OSN. Therefore, by means of a BL rule, wall owners are for example able to ban from their walls users they do not directly know (i.e., with which they have only indirect relationships), or users that are friend of a given person as they may have a bad opinion of this person. This banning can be adopted for an undetermined time period or for a specific time window. Moreover, banning criteria may also take into account users’ behavior in the OSN. More precisely, among possible information denoting users’ bad behavior we have focused on two main measures.

The first is related to the principle that if within a given time interval a user has been inserted into a BL for several times, say greater than a given threshold, he/she might deserve to stay in the BL for another while, as his/her behavior is not improved. This principle works for those users that have been already inserted in the considered BL at least one time. In contrast, to catch new bad behaviors, we use the Relative Frequency (RF) that let the system be able to detect those users whose messages continue to fail the FRs. The two measures can be computed either locally, that is, by considering only the messages and/or the BL of the user specifying the BL rule or globally, that is, by considering all OSN users.

## **5.2 Filter Wall Architecture**:

The architecture in support of OSN services is a three-tier structure . The first layer, called Social NetworkManager (SNM), commonly aims to provide the basic OSN functionalities (i.e., profand relationship management),whereas the second layer provides the support for external SoNetwork Applications (SNAs).4 The supported SNAs may in turn require an additionaforneededGraphical User Interfaces (GUIs). According to this reference architecture, the proposed system is placed in thesecond and third layers. The core components of the OSN system are theContent-Based Messages Filtering (CBMF) and the ShortText Classifier (STC) modules.

### **5.2.1 Content Based Filtering**:

Information filtering systems are designed to classify a stream of dynamically generainformation dispatched asynchronously by an information producer and present to the user those information that are likely to satisfy his/her requirements

In content-based filtering each user is assumed to operate independently. As a resultacontent-based filtering system selects information items based on the correlation between the content the items and the user preferences as opposed to a collaborative filtering system that chooses it based

on the correlation between people with similar preferences.

Content-based filtering is mainly based on the use of the ML paradigm according to which a classifier is automatically induced by learning from a set of pre-classified examples. The feature extraction procedure maps text into a compact representation of its content and is uniformly applied to training and generalization phases.

### **5.2.2 Policy Based Personalization of OSN Contents**:

Recently, there have been some proposals exploiting classification mechanisms for personalizing access in OSNs. For instance, in a classification method has been proposed to categorize short text messages in order to avoid overwhelming users of microblogging services by raw data. The system described in focuses on Twitter2 and associates a set of categories with each tweet describing its content. The user can then view only certain types of tweets based on his/her interests. In contrast, Golbeck and Kuter propose an application, called FilmTrust, that exploits OSN trust relationships and provenance information to personalize access to the website. However, such systems do not provide a filtering policy layer by which the user can exploit the result of the classification process to decide how and to which extent filtering out unwanted information. In contrast, our filtering policy language allows the setting of FRs according to a variety of criteria, that do not consider only the results of the classification process but also the relationships of the wall owner with other OSN users as well as information on the user profile. Moreover, our system is complemented by a flexible mechanism for BL management that provides a further opportunity of customization to the filtering procedure.

The only social networking service we are aware of providing filtering abilities to its users is MyWOT,3 a 2http://www.twitter.com 3http://www.mywot.com social networking service which gives its subscribers the ability to:

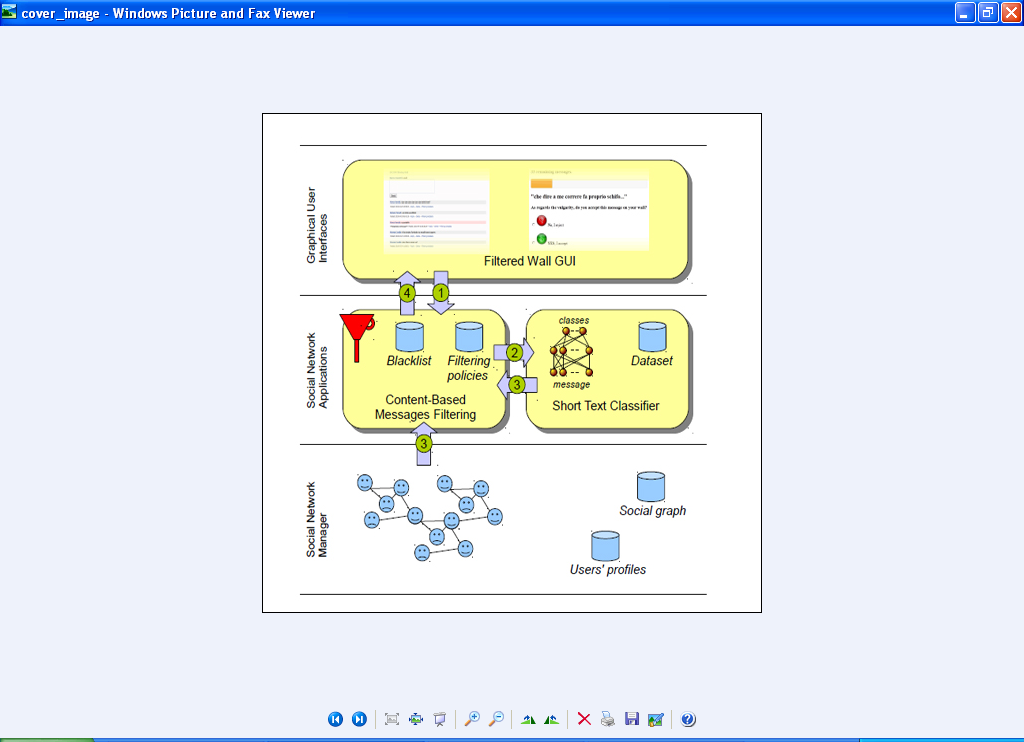
1) rate resources with respect to four criteria: trustworthiness, vendor reliability, privacy, and child safety;

2) specify preferences determining whether the browser should block access to a given resource, or should simply return a warning message on the basis of the specified rating. Despite the existence of some similarities, the approach

adopted by MyWOT is quite different from ours. In particular, it supports filtering criteria which are far less flexible than the ones of Filtered Wall since they are only based on the four above-mentioned criteria. Moreover, no automatic classification mechanism is provided to the end user.

Our work is also inspired by the many access control models and related policy languages and enforcement mechanisms that have been proposed so far for OSNs (see for a survey), since filtering shares several similarities with access control. Actually, content filtering can be considered as an extension of access control, since it can be used both to protect objects from unauthorized subjects, and subjects from inappropriate objects. In the field of OSNs, the majority of access control models proposed so far enforce topology-based access control, according to which access control requirements are expressed in terms of relationships that the requester should have with the resource owner. We use a similar idea to identify the users to which a FR applies. However, our filtering policy language extends the languages proposed for access control policy specification in OSNs to cope with the extended requirements of the filtering domain. Indeed, since we are dealing with filtering of unwanted contents rather than with access control, one of the key ingredients of our system is

the availability of a description for the message contents to be exploited by the filtering mechanism. In contrast, no one of the access control models previously cited exploit the content of the resources to enforce access control. Moreover, the notion of BLs and their management are notonsidered by any of the above-mentioned access control models.



**Fig. 5.1. Filtered Wall Conceptual Architecture**

## **5.3 Short Text Classifier**:

Established techniques used for text classification work well on datasets with large documents such as newswires corpora , but suffer when the documents in the corpus are short. In thi context, critical aspects are the definition of a set of characterizing and discriminant features allowing the representation of underlying concepts and the collection of a complete and consistent supervised examples.Our study is aimed at designing and evaluating various representation combinationwith a neural learning strategy to sematically categorize short texts. From a ML point of vieapproach the task by defining a hierarchical two level strategy assuming that it is better to identifeliminate “neutral” sentences, then classify “non neutral” sentences by the class of interest insteadof doing everything in one step. This choice is motivatedby related work showing advantages iclassifying textand/or short texts using a hierarchical strategy .

The firstlevel task is conceived as a hard classification in whichshort texts are labeled with crisp Neutral and Non-Neutral labels. second level soft classifier acts on the crisp set of non-neutral short texts and, for each of them, “simply” produces estimated appropriateness or “gradual membership” for each of the conceived classes, without taking any “hard” decision on any of them. Such a list of grades is then used subsequent phases of the filtering process.

### **5.3.1 Text Representation**:

The extraction of an appropriate set of features by which representing the text of a given document is a crucial task strongly affecting the performance of the overall classificatiostrategy. we consider three types of features, BoW, Document properties (Dp) and Contextual Featur(CF). The first two types of features, already used in [5], are endogenous, that is, they are entirederived from the information contained within the text of the message. Text representation usendogenous

Knowledge has a good general applicability, however in operational settings it is legitimate to use also exogenous. knowledge, i.e., any source of information outside the message body but directly or indirectly related to the message itself. We introduce CF modeling information thacharacterize

The environment where the user is posting. The underlying model for text representation is Vector Space Model (VSM) [37] according to which a textdocument dj is represented as a vector of binary or real weights dj = w1j ; : : : ;wjT jj , where T is the set of terms(sometimes also called features) that occur at least once in at least one document of the collection T r, and wkj 2 [0; 1]

Represents how much term tk contributes to the semantics of document dj . In the representation, terms are identified with words. In the case of non-binary weighting,the weight wkj of term tk in document dj is computed according to the standard term frequency - inverse document.

## **5.4 Machine Learning-based Classification**:

We address short text categorization as a hierarchical two-level classification process. The first-level classifier performs a binary hard categorization that labels messages as Neutral and Non-Neutral. The first-level filtering task facilitates the subsequent second-level task in which a finer-grained classification is performed. The second-level classifier performs a soft-partition of Non-neutral messages assigning a given message a gradual membership to each of the non neutral classes.

Among the variety of multi-clasML models well-suited for text classification, we choose the RBFN model for the experimented competibehavior with respect to other state of the art classifiers. RFBNs have a single hidden layer of processing unitswith local, restricted activation domain: a Gaussian function is commonly used, but any other locally tunablefunction can be used. They were introduced as a neural netevolution of exact interpolation , and are demonstrated to have the universal approximation property.

May produce confidence values and it may be robust to outliers; drawbacks are the pot sensitivity to input parameters, and potential overtraining sensitivity. The first level classifier is tstructured as a regular RBFN. In the second level of the classification stage we introducea modification of the standard use of RBFN. Its regular use in classification includes a hard decision on the outputvalues: according to the winner-take-all rule, a given input pattern is assigned withcorresponding to the winner output neuron which has the highest value. In our approach, we consider all values of the output neurons as a result of the classification task and we interpret them gradual estimation of multi-membership to classes.

# 6.Coding

## 6.1 Databasecon.java

package databaseconnection;

import java.io.PrintStream;

import java.sql.Connection;

import java.sql.DriverManager;

public class databasecon

{

public databasecon()

{

}

public static Connection getconnection()

{

try

{ Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/message\_filter", "root", "");

}

catch(Exception e)

{ System.out.println("class error");

}

return con;

}

static Connection con;

}

## 6.2 UserRegistration.jsp

<%@ page import="java.util.date.\*,java.util.text.DateFormat.\*,java.text.ParseException.\*"%>

<%@page import="com.oreilly.servlet.\*,java.sql.\*,java.lang.\*,databaseconnection.\*,java.text.SimpleDateFormat,java.util.\*,java.io.\*,javax.servlet.\*, javax.servlet.http.\*" %>

<%@ page import = "java.util.Date,java.text.SimpleDateFormat,java.text.ParseException"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link rel="stylesheet" type="text/css" href="style.css" />

<title>A-System-to-Filter-Unwanted-Messages-from-OSN-User-Walls</title>

<script type="text/javascript">

function valid()

{

var aa=document.s.un.value;

if(aa=="")

{

alert("Enter Name");

document.s.un.focus();

return false;

}

if((document.s.gen[0].checked==false)&&(document.s.gen[1].checked==false))

{

alert("Select Gender");

document.s.un.focus();

return false;

}

var db=document.s.dob.value;

if(db=="")

{

alert("Enter The Date Of Birth");

document.s.dob.focus();

return false;

}

var k = document.s.mob.value;

if(k=="")

{

alert("Enter mobile number");

document.s.mob.focus();

return false;

}

if(isNaN(k))

{

alert("Enter mobile number in numbers");

document.s.mob.focus();

return false;

}

if(k.charAt(0)!=9)

{

alert("Enter the correct mobile no");

document.s.mob.focus();

return false;

}

if(k.length!=10)

{

alert("Enter 10 digits");

document.s.mob.focus();

return false;

}

var emailfilter=/^\w+[\+\.\w-]\*@([\w-]+\.)\*\w+[\w-]\*\.([a-z]{2,4}|\d+)$/i;

var m=emailfilter.test(document.s.eid.value);

if(m==false)

{

alert("Please enter a valid Email Id");

document.s.eid.focus();

return false;

}

var ad = document.s.ad.value;

if(ad=="")

{

alert("Enter Address");

document.s.ad.focus();

return false;

}

var a=document.s.uid.value;

if(a=="")

{

alert("Enter User ID");

document.s.uid.focus();

return false;

}

var b=document.s.pass.value;

if(b=="")

{

alert("Enter Password");

document.s.pass.focus();

return false;

}

}

</script>

</head>

<body>

<%

java.util.Date now = new java.util.Date();

String DATE\_FORMAT1 = "dd/MM/yyyy";

SimpleDateFormat sdf1 = new SimpleDateFormat(DATE\_FORMAT1);

String strDateNew1 = sdf1.format(now);

String alphabet="MOBVCXZPNWIEURYTKSLAHFJDGQ";

int character=(int)(Math.random()\*26);

String s=alphabet.substring(character, character+4);

//System.out.println(s);

int y=100+(int)(Math.random()\*1000);

String z=s+y;

session.setAttribute("z1",z);

%>

<div id="container">

<div id="mainpic">

<h1>A-System-to-Filter-Unwanted-Messages-from-OSN-User-Walls</h1>

</div>

<div id="menu">

<ul>

<li class="menuitem"><a href="index.html">Back</a></li>

<!-- <li class="menuitem"><a href="#">About</a></li>

<li class="menuitem"><a href="#">Products</a></li>

<li class="menuitem"><a href="#">Services</a></li>

<li class="menuitem"><a href="#">Design</a></li>

<li class="menuitem"><a href="#">Contact</a></li>-->

</ul>

</div>

<div id="content">

<h2>&nbsp; </h2>

<h3>&nbsp;</h3>

<table width="800">

<tr>

<td>

<fieldset>

<table width="500" height="300">

<tr>

<td>

<form name="s" action="register1\_user.jsp" method="post" onsubmit="return valid()" enctype="multipart/form-data">

<table align="center" cellpadding="5" cellspacing="5" width="420" border="0">

<tr>

<tr>

<td colspan="2" align="center"><strong><font color="#FF3300" size="+1" face="Courier New">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Newuser Registration</font></strong></td>

</tr>

<td colspan="2" align="center">&nbsp;</td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Name</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="text" name="un" class="b"></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Gender</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="radio" name="gen" value="male" class="b">Male&nbsp;&nbsp;&nbsp;&nbsp;

<input type="radio" name="gen" value="female" class="b">Female</td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>DOB</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="text" name="dob" class="b"></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Mobile</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="text" name="mob" class="b"></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Email ID</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="text" name="eid" class="b"></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Address</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <textarea name="ad"></textarea></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>User ID</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="text" name="uid" class="b"></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Password</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="password" name="pass"></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Profile Image</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="file" name="fil" class="b"></td>

</tr>

<tr>

<td><font face="Courier New" size="+1" color="#000000"><strong>Date</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="text" name="date" class="b" value="<%=strDateNew1%>"></td>

</tr>

<tr>

<tr>

<td></td>

<td><input type="submit" name="s" value="submit" class="b1" >

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type="reset" name="r" value="clear" class="b1"></td>

</tr>

</table>

</form>

</td>

</tr>

</table>

</fieldset>

<td>&nbsp;</td>

</td>

<td>

<fieldset>

<table width="200" height="300" border="0">

<tr>

<td><font size="2"><b>

<%

String message=request.getParameter("message");

if(message!=null && message.equalsIgnoreCase("success"))

{

out.println("<font color='red'><blink>Registered Successfully !</blink></font>");

}

%>

</b></font></td>

</tr>

<tr>

<td align="justify"><p align="center"><img src="images/reg.jpg" />

</td>

</tr>

</table>

</fieldset>

</td>

</tr>

</table>

<br />

<br />

</div>

</div>

</body>

</html>

## 6.3 UserLogin.jsp

<%@ page import="java.sql.\*" import="databaseconnection.\*"%>

<%

int a=0;

String b=null,c=null,d=null,e=null,f=null,g=null,h=null,i=null,j=null,k=null,l=null,m=null;

String idu = request.getParameter("uname");

String passu= request.getParameter("passw");

try

{

Connection con1 = databasecon.getconnection();

Statement st1 = con1.createStatement();

//String sql="select uid,name,userAES\_DECRYPT(email,'key') from user";

String sss1 = "select \* from user\_reg where user\_id='"+idu+"' && pass='"+passu+"'";

ResultSet rs1=st1.executeQuery(sss1);

if(rs1.next())

{

a=rs1.getInt(1);

session.setAttribute("a1",a);

b=rs1.getString(2);

session.setAttribute("b1",b);

c=rs1.getString(3);

session.setAttribute("c1",c);

d=rs1.getString(4);

session.setAttribute("d1",d);

e=rs1.getString(5);

session.setAttribute("e1",e);

f=rs1.getString(6);

session.setAttribute("f1",f);

g=rs1.getString(7);

session.setAttribute("g1",g);

h=rs1.getString(8);

session.setAttribute("h1",h);

i=rs1.getString(9);

session.setAttribute("i1",i);

j=rs1.getString(10);

session.setAttribute("j1",j);

k=rs1.getString(11);

session.setAttribute("k1",k);

l=rs1.getString(12);

session.setAttribute("l1",l);

m=rs1.getString(13);

session.setAttribute("m1",m);

response.sendRedirect("user\_home.jsp?message=success");

}

else

{

response.sendRedirect("index.html?message=fail");

}

}

catch(Exception e1)

{

out.println(e1.getMessage());

}

%>

## 6.4 Filtertext.jsp

<%@ page language="java" import="java.sql.\*,databaseconnection.\*" %>

<%

try

{

String a = request.getParameter("wctry");

String b = request.getParameter("wrd");

Connection con = databasecon.getconnection();

Statement st = con.createStatement();

String sss = "insert into filter\_message(detail1,detail2) values('"+a+"','"+b+"')";

int x = st.executeUpdate(sss);

if(x!=0)

{

%>

<jsp:forward page="admin\_home.jsp" />

<%

}

}

catch(Exception e)

{

out.print(e);

}

%>

# 7. TESTING

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

## 7.1 TYPES OF TESTS

### 7.1.1 Unit testing

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

### 7.1.2 Integration testing

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

### 7.1.3 Functional testing

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

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

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

### 7.1.4 System Testing

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

### 7.1.5 White Box Testing

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

### 7.1.6 Black Box Testing

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

**1 Unit Testing:**

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

**Test strategy and approach**

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

**Test objectives**

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

**Features to be tested**

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

**2 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case number** | **Test case** | **Input** | **Expected output** | **Obtained output** |
| 1 | User Login | Give user name and password | User page open | User page open |
| 2 | Admin  Login | Give the user name and password | Admin  page is open | Admin  page is open |
| 3 | User Registration | Give all the details | Login page will open | Login page open |
| 4 | Profile | View profile | Add filter word | View filter performance |
| 5 | Image | Search image | View image | Upload image |
| 6 | comments | Write comments | Add friends | View other friends |

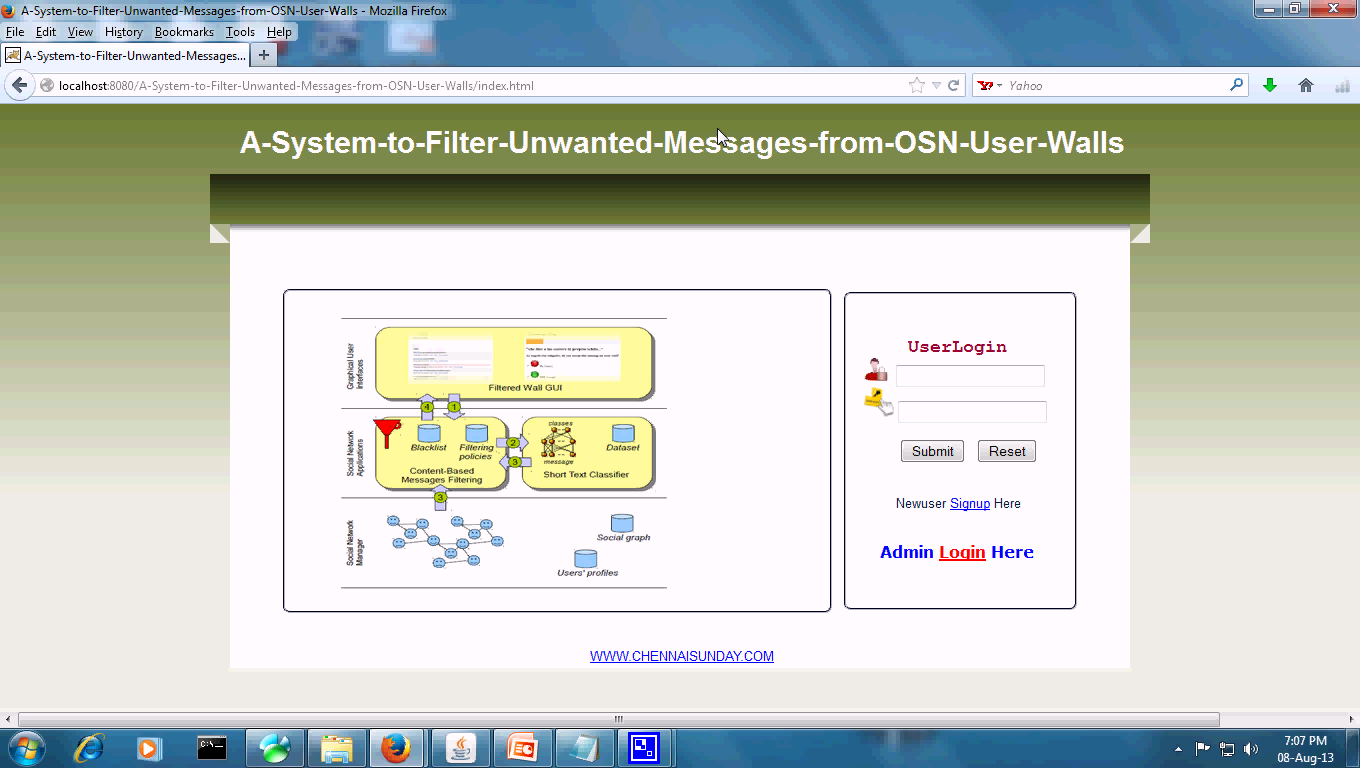
**Figure: 7.1.6 Test result table**

# 8.RESULTS

Snapshot is nothing but every moment of application while running.It gives the clear elaboratedof application.It will be usefull for the new user to understand for the future steps.

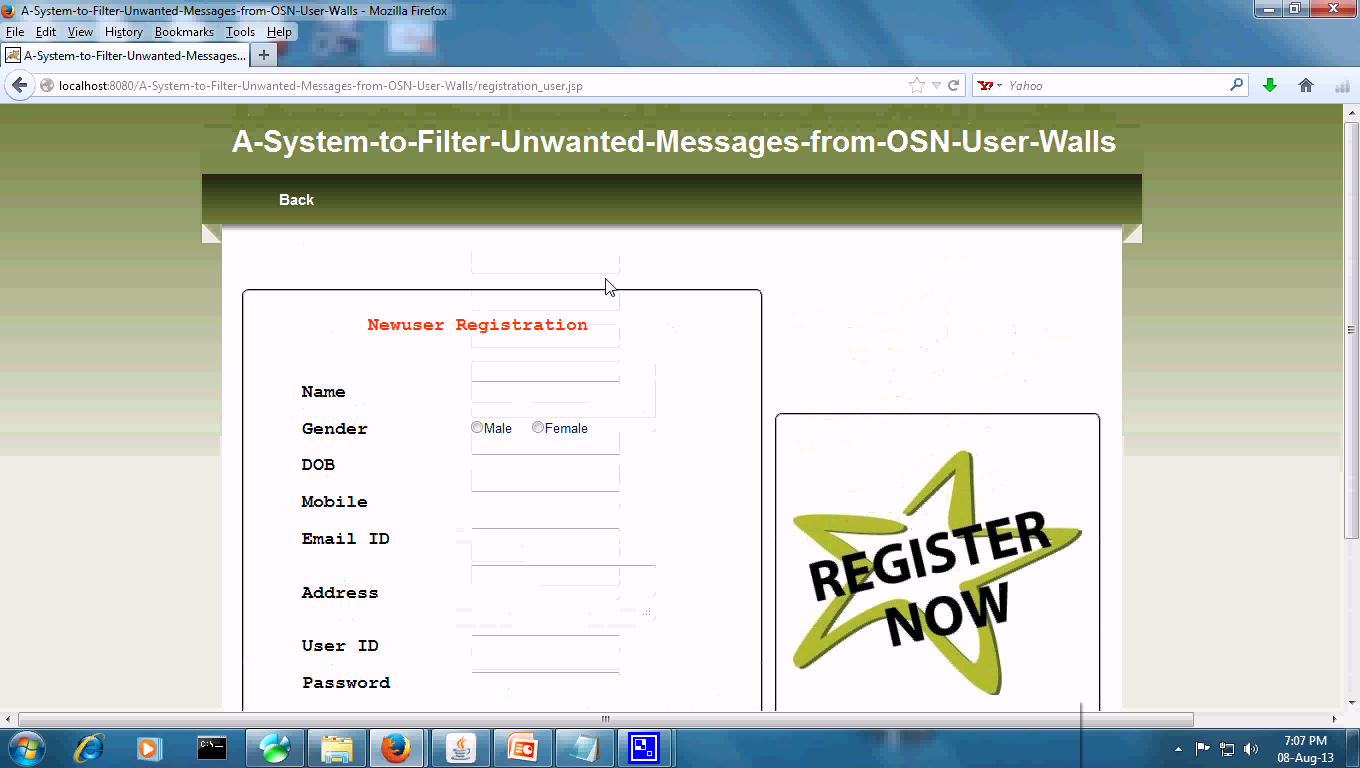
## 8.1 RIOUS SNAPSHOTS

### 8.1.1 USER LOGIN PAGE:

****

**Figure 8.1.1 user login page**

### 8.1.2 New user Registration

****

**Figure:8.1.2 New user Registration**

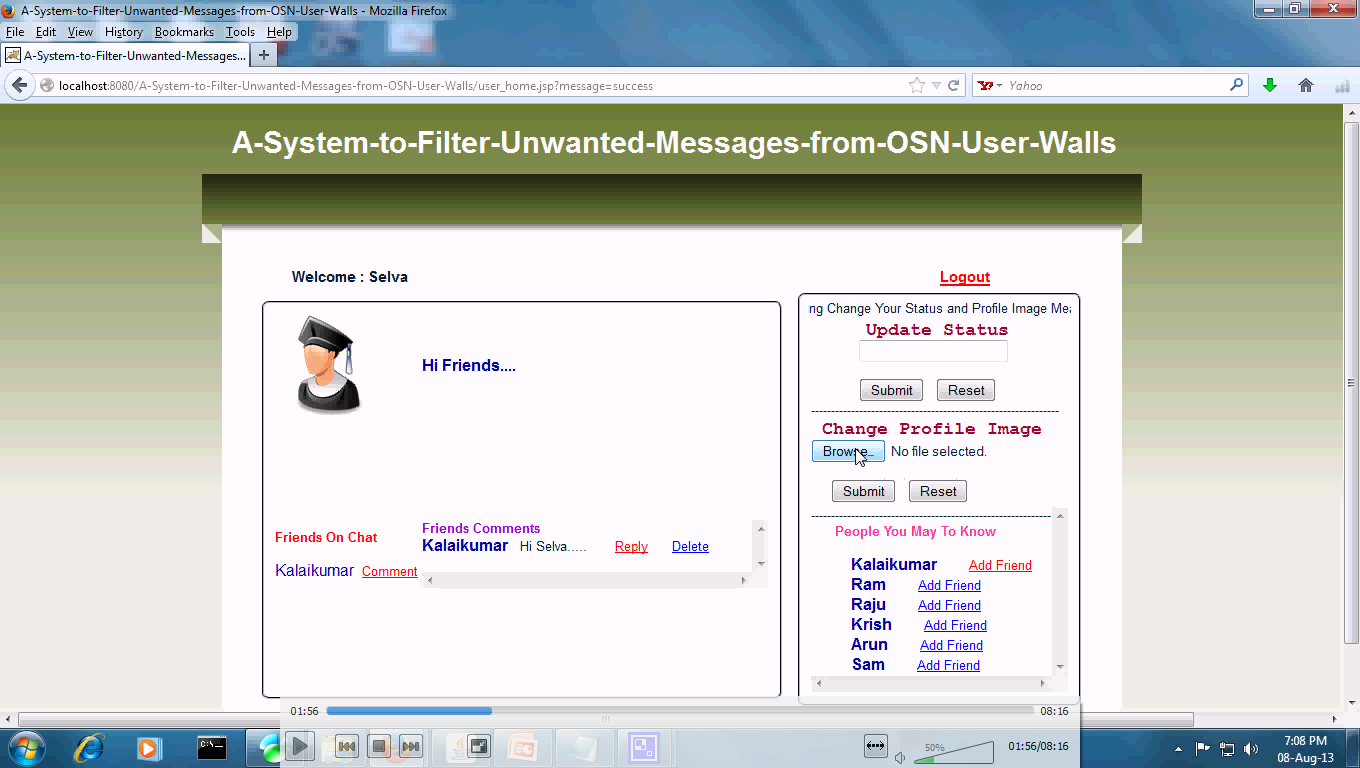
### 8.1.3 Update Status page

****

**Figure:8.1.3 Update status page**

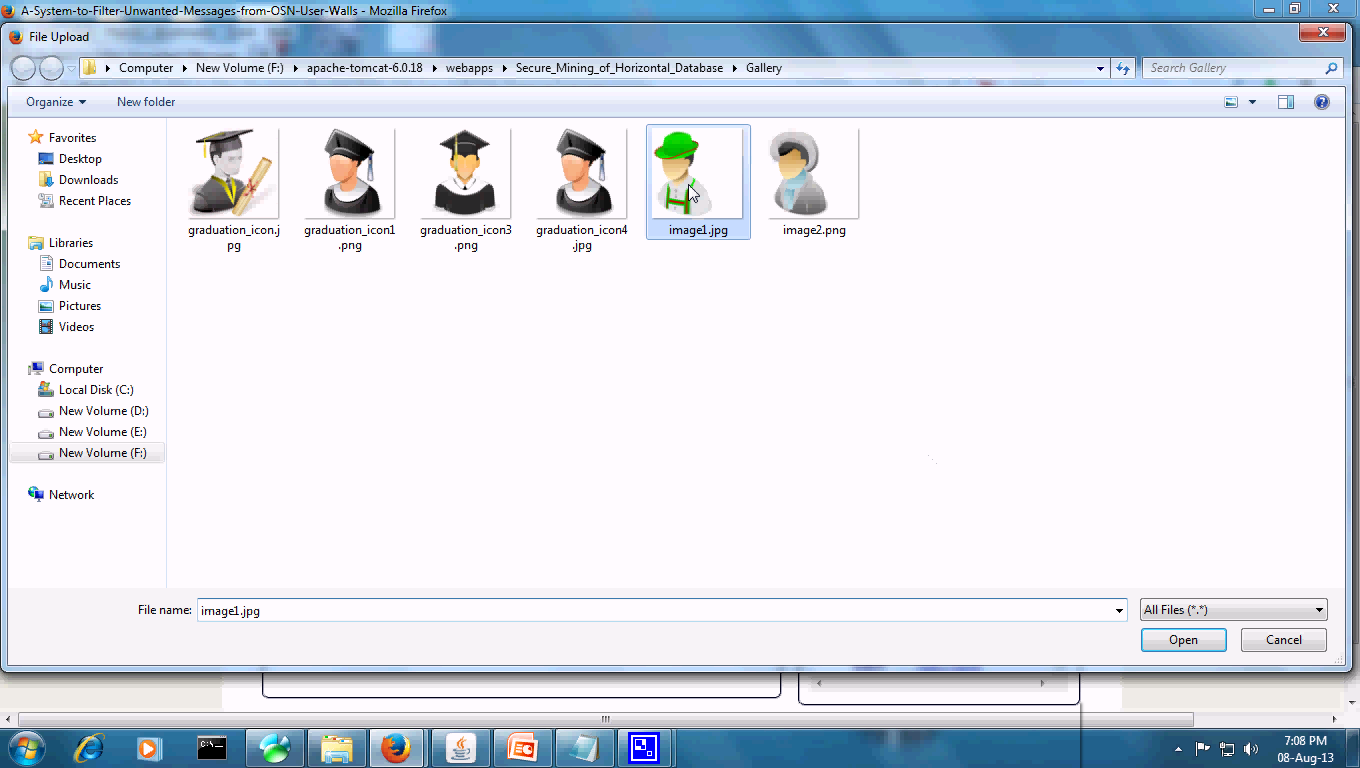
### 8.1.4 Update status page to give the symptoms

****

****

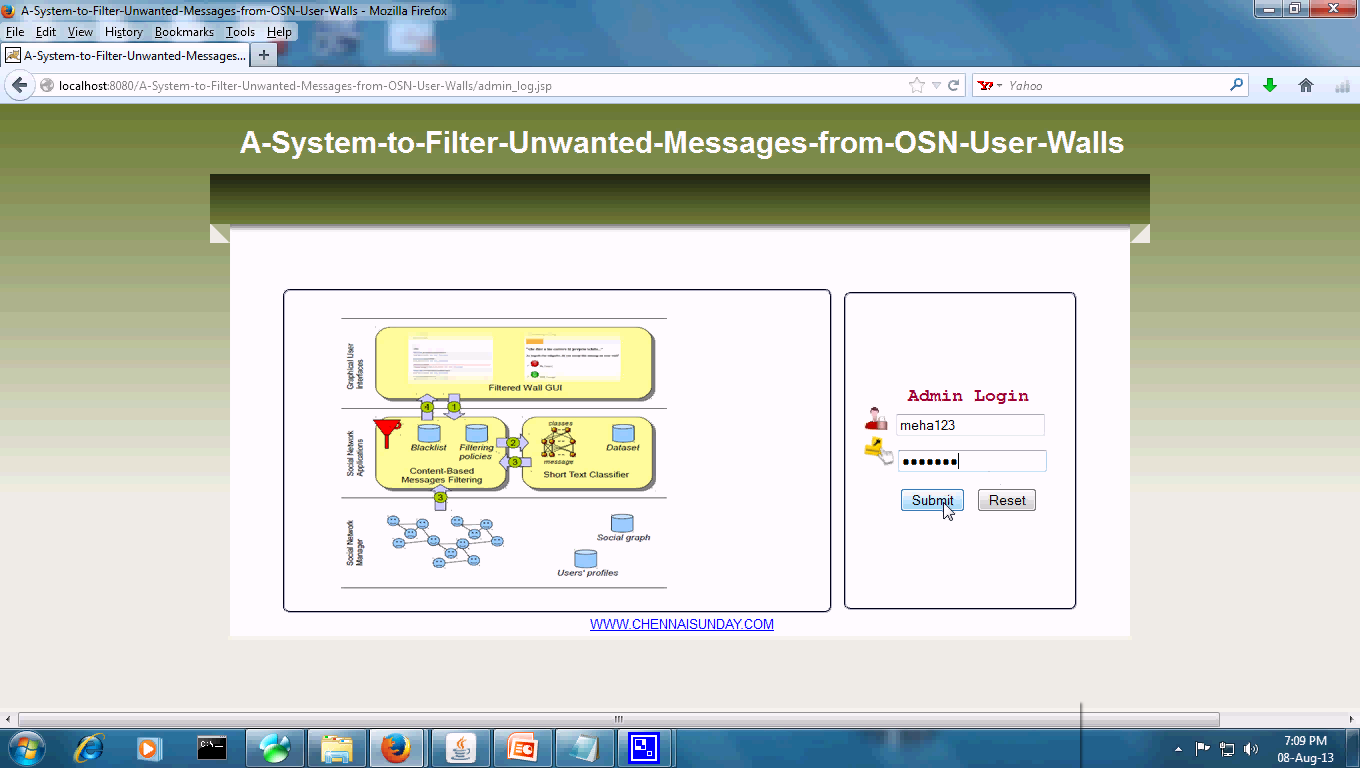
**Figure: 8.1.4 diagram for Update status page to give the symptoms**

### 8.1.5 Selecting the image from data base

****

**Figure: 8.1.5 selecting the image from data base**

### 8.1.6 Admin login page

****

**Figure: 8.1.6 admin login page**

# **9.GitHub links :**

**GitHub Link:**

**https://github.com/ReshmiReddy9848/PROJECT-CODE**

# 10.CONCLUSION AND FUTURE WORK

In this project, we have presented a system to filter undesired messages from OSN walls. The system exploits a ML Soft classifier to enforce customizable content-dependent FRs. The early encouraging results we have obtained on the classification procedure prompt us to continue with other work that will aim to improve the quality of classification in pictures and videos which helps in stoping the spreading of unqualified news outside.

# 11.References

[1] D. Banodkar, K. K. Ramakrishnan, S. Kalyanaraman, A. Gerber, and O. Spatscheck, “Multicast instant channel change in IPTV system,” in *Proceedings of IEEE COMSWARE*, January 2008.

[2] “Microsoft tv: Iptv edition,” <http://www.microsoft.com/tv/IPTVEdition.mspx>.

[3] H. A. Lagar-Cavilla, J. A. Whitney, A. Scannell, R. B. P. Patchin, S. M. Rumble, E. de Lara, M. Brudno, and M. Satyanarayanan, “SnowFlock: Virtual Machine Cloning as a First Class Cloud Primitive,” *ACM Transactions on Computer Systems (TOCS)*, 2011.

[4] V. Aggarwal, V. Gopalakrishnan, R. Jana, K. K. Ramakrishnan, and V. Vaishampayan, “Exploiting Virtualization for Delivering Cloud-based IPTV Services,” in *Proc. of IEEE INFOCOM (mini-conference)*, Shanghai,

April 2011.

[5] J. A. Stankovic, M. Spuri, K. Ramamritham, and G. C. Buttazzo, *Deadline Scheduling for Real-Time Systems: Edf and Related Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[6] N. V. Thoai and H. Tuy, “Convergent algorithms for minimizing a concave function,” in *Mathematics of operations Research*, vol. 5, 1980.

[7] R. Urgaonkar, U. Kozat, K. Igarashi, and M. J. Neely, “Dynamic resource allocation and power management in virtualized data centers,” in *Proceedings of IEEE IFIP NOMS*, March 2010.

[8] C. L. Liu and J. W. Layland, “Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment,” *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.