

Optimizing Spam Filtering with Machine Learning

Team id: NM2023TMID31961

	Name of the Student	NM ID
TEAM LEADER	V.Rajalakshmi	EA5DB3F51581109950746FB44D9D2FD1
TEAM MEMBER	P.Manjula	F8E4AEE7F55A2C2A5764D8C3D7A80867
	T.Malathi	1C891A2C6EA31F7B15B4EA529F83659D
	B.Muthulakshmi	84D2ABBF91C9E668A0F45F26532ACA16

Government Arts And Science

College, Kadayanallur-627751

INDEX

S.No	Contents	Page No.
1	Introduction	3
2	Problem definition And design thinking	4
3	System Analysis	5
4	Advantages and Disadvantages	33
5	Conclusion	35
6	Future Enhancement	35
7	Appendix	36

1.INTRODUCTION

Internet is a major source of spreading terrorism through speeches and videos. Terrorist organizations use internet especially social networks to brain wash individuals and also promote terrorist activities through provocative web pages that inspire helpless people to join terrorist organizations. So here we propose an efficient web data mining system to detect such web properties and flag them automatically for human review.

Investing in cyber-threat intelligence is an important first step in understanding where high-risk vulnerabilities exist in an organisation. These are likely the same areas being targeted by mass distributed hacking platforms, and shoring up their defences can stop low-risk, high-reward threat actors before they have a chance to gain a foothold. Here the cybercrime underground has a highly professional business model that supports its own underground economy. This business model, known as CaaS (Crimeware-as-a-Service) is “a business model used in the underground market where illegal services are provided to help underground buyers conduct cybercrimes, such as attacks, infections, and money laundering in an automated manner,”. Thus, CaaS is referred to as a do-it-for-me service, unlike crimeware which is a do-it-yourself product. Because CaaS is designed for novices, its customers do not need to run a hacking server or have high-level hacking skills.

2.PROBLEM DEFINITION AND DESIGN THINKING

A Denial of Service (DoS) attack can be characterized as an attack with the purpose of preventing legitimate users from using a victim computing system or network resource . A Distributed Denial of Service (DDoS) attack is a large-scale, coordinated attack on the availability of services of a victim system or network resource, launched indirectly through many compromised computers on the Internet. The services under attack are those of the “primary victim”, while the compromised systems used to launch the attack are often called the “secondary victims.” The use of secondary victims in performing a DDoS attack provides the attacker with the ability to wage a much larger and more disruptive attack, while making it more difficult to track down the original attacker. As defined by the World Wide Web Security FAQ: A Distributed Denial of Service (DDoS) attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the Denial of Service significantly by harnessing the resources of multiple unwitting accomplice computers which serve as attack.

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In an Existing system, detection of terrorism was presented by using Web traffic content as the audit information. After that the typical behavior of terrorists by applying a data mining algorithm to the textual content of terror-related Web sites. The resulting profile was used by the system to perform detection of users suspected of being engaged in terrorist activities. And this algorithm should be based on the content of existing terrorist sites and known terrorist traffic on the Web.

3.2 PROPOSED SYSTEM

There are two features used in this system that is data mining and web mining. Data mining is a technique used to mine out patterns of useful data from large data sets. Web mining also consists of text mining methodologies that allow us to scan and extract useful content from unstructured data. This system will check the sender messages and whether the message is promoting terrorism. Data mining as well as web mining are used together at times for efficient system development. System will find the unwanted messages that are more susceptible to terrorism and will send directly to the receiver's spam account. It will give more awareness to the users.

DETAIL DESCRIPTION OF TECHNOLOGY

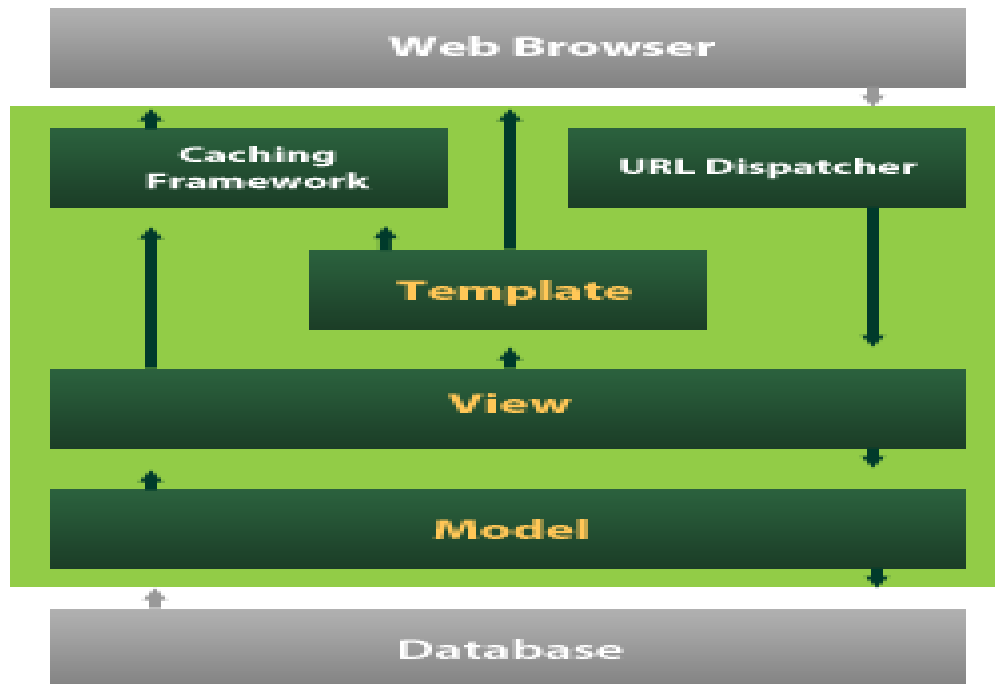
AN INTRODUCTION TO PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

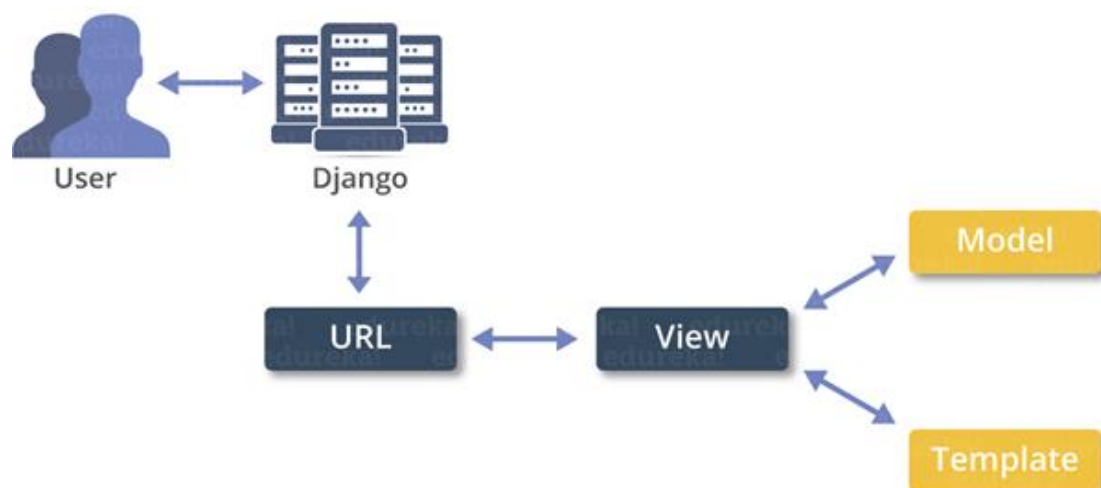
DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.



Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models



INTRODUCTION TO MYSQL

MySQL (<http://www.mysql.com>) is a robust SQL database server developed and maintained by T.c.X DataKonsultAB of Stockholm, Sweden. Publically available since 1995, MySQL has risen to become one of the most popular database servers in the world, this popularity due in part to the server's speed, robustness, and flexible licensing policy. (See note for more information regarding MySQL's licensing strategy.)

Given the merits of MySQL's characteristics, coupled with a vast and extremely easy-to-use set of predefined interfacing functions, MySQL has arguably become PHP's most-popular database counterpart.

Installation

MySQL is so popular among PHP users that support for the db server is automatically built into the PHP distribution. Therefore, the only task that you are left to deal with is the proper installation of the MySQL package. MySQL is compatible with practically every major operating system, including, among others, FreeBSD, Solaris, UNIX, Linux, and the various Windows versions. While the licensing policy is considerably more flexible than that of other database servers, I strongly suggest taking some time to read through the licensing information found at the MySQL site (<http://www.mysql.com>).

You can download the latest version of MySQL from one of the many worldwide mirrors. A complete listing of these mirrors is at <http://www.mysql.com/downloads/mirrors.html>. At the time of this writing the latest stable version of MySQL was 3.22.32, with version 3.23 in beta. It is in your best interest to always download the latest stable version. Go to the mirror closest to you and download the

version that corresponds with your operating system platform. You'll see links at the top of the page pointing to the most recent versions. Be sure to read through the entire page, as several OS-specific downloads are at the conclusion. The MySQL development team has done a great job putting together extensive documentation regarding the installation process.

Configuring MySQL

After a successful installation, it is time to configure the MySQL server. This process largely consists of creating new databases and configuring the MySQL *privilege tables*. The privilege tables control the MySQL database access permissions. Correct configuration of these tables is pivotal to securing your database system, and therefore it is imperative that you fully understand the details of the privilege system before launching your site into a production environment. Although a chore to learn at first, the MySQL privilege tables are extremely easy to maintain once you understand them. A complete introduction to these tables is certainly out of the scope of this book. However, a number of resources available on the Web are geared toward bringing MySQL users up to speed. Check out the MySQL site (<http://www.mysql.com>) for further information. Once you have correctly installed and configured the MySQL distribution, it's time to begin experimenting with Web-based databasing! The next section turns our attention towards exactly this matter, starting with an introduction of PHP's MySQL functionality.

PHP's Predefined MySQL Functions

Once you have created and successfully tested the necessary permissions, you are ready to begin using the MySQL server. In this section, I introduce the predefined MySQL functions, enabling you to easily interface your PHP scripts with a MySQL server. Here is the general order of events that take place during the MySQL server

communications process: 1. Establish a connection with the MySQL server. If the connection attempt fails, display an appropriate message and exit process. 2. Select a database on the MySQL server. If you cannot select the database, display an appropriate message and exit process. It's possible to simultaneously have several databases open for querying. 3. Perform necessary queries on selected database(s). 4. Once the querying is complete, close the database server connection. The example tables (products, customers, orders) in Figure 11-1 are used as the basis for the examples in the remainder of this section. If you would like to follow along with these examples, I suggest going back and creating them now. Alternatively, make a copy of the pages so you do not have to continuously flip back and forth. With that said, let's begin at the beginning, that is, how to connect to the MySQL database server.

mysql_connect()

The function `mysql_connect()` is used to establish an initial connection with the MySQL server. Once a successful connection is established, a database residing on that server can be selected.

The syntax is:

```
int mysql_connect([string hostname [:port] [:path/to/socket] [, string  
username] [, string password])
```

The *hostname* is the name of the host as listed in the MySQL server privilege tables. Of course, it is also used to direct the request to the Web server hosting the MySQL server, since it is possible to connect to a remote MySQL server. An optional *port* number can be included along with the host, in addition to an optional path to a socket when a local host is specified. Both the *username* and *password* input parameters should correspond to the username and password, respectively, as

specified in the MySQL server privilege tables. Note that all of the input parameters are optional,

since the privilege tables can be loosely configured to accept a nonauthenticated connection. If the *hostname* parameter is empty, `mysql_connect()` attempts to connect to the local host.

An example connection call follows: `@mysql_connect("localhost", "web", "4tf9zzzf")` or `die("Could not connect to MySQL server!")`; In this case, `localhost` is the server host, `web` is the username, and `4tf9zzzf` is the password. The `@` preceding the `mysql_connect()` function will suppress any error message that results from a failed attempt, instead producing the custom one specified in the `die()` call. Note that no value is returned from the `mysql_connect()` call.

Now, `$link1` and `$link2` can be called as needed in subsequent queries. You will soon learn exactly how these link IDs are used in queries to specify the

intended server.

mysql_select_db()

Once a successful connection is established with the MySQL server, a database residing on that server can be selected. This is accomplished with `mysql_select_db()`.

NOTE The function `mysql_pconnect()` offers persistent connection support. In multiuser environments, `mysql_pconnect()` is recommended over `mysql_connect()` as a means for conserving system resources. The `mysql_pconnect()` input and return parameters are exactly the same as in `mysql_connect()`.

The input parameter `database_name` should be selected and assigned an identification handle (returned by `mysql_select_db()`). Note that the input parameter `link_id` is optional. This is true only when just a single MySQL server connection is open. When

multiple connections are open, `link_id` must be specified. An example of how a database is selected using `mysql_select_db()` follows:

If there is only one database selection, there is no need to return a database ID. However, as with `mysql_connect()`, when multiple databases are open, the database ID must be returned so there is a way to specify exactly which database you would like to perform a query on; otherwise the most recently *opened* link is used.

mysql_close()

Once you have finished querying the MySQL server, you should close the connection. The function `mysql_close()` will close the connection corresponding to the optional input parameter `link_id`. If the `link_id` input parameter is not specified, `mysql_close()` will close the most recently opened link. The syntax is:

The input parameter `query` corresponds to an SQL query. This query is sent either to the server connection corresponding to the last opened link or to the connection specified by the optional input parameter `link_id`. People often mistakenly think that the `mysql_query()` function returns the results of the query. This is not the case. Depending on the type of query, `mysql_query()` has different outcomes. In a successful SELECT SQL statement, a result ID is returned that can subsequently be passed to `mysql_result()` so the selected data can be formatted and displayed to the screen. If the query fails, `FALSE` is returned. The function `mysql_result()` is introduced later in this section. Furthermore, the number of rows that have been selected can be determined by executing `mysql_num_rows()`. This function is also introduced later in this section. In the case of SQL statements involving INSERT, UPDATE, REPLACE, or DELETE, the function `mysql_affected_rows()` can be called to determine how many rows were affected by the query.

(The function `mysql_affected_rows()` is introduced next.) With that said, I will delay presenting an example until the `mysql_result()` and `mysql_affected_rows()` functions are introduced.

`mysql_pconnect()`.

TIP If you are concerned that you are using up too much memory when making various query calls, call the predefined PHP function `mysql_free_result()`. This function, which takes as input the `result_id` returned from `mysql_query()`, will free up all memory associated with that query call.

Notice that the input parameter `link_id` is optional. If it is not included, `mysql_affected_rows()` attempts to use the last opened `link_id`. Consider the following. This will not work for queries involving a `SELECT` statement. To determine the number of rows returned from a `SELECT`, use the function `mysql_num_rows()` instead. This function is introduced next.

`mysql_num_rows()`

The function `mysql_num_rows()` is used to determine the number of rows returned from a `SELECT` query statement.

its syntax is:

```
int mysql_num_rows (int result)
```

A usage example of `mysql_num_rows()` follows:

`mysql_result()`

The function `mysql_result()` is used in conjunction with `mysql_query()` (when a `SELECT` query is involved) to produce a data set.

Its syntax is:

```
int mysql_result (int result_id, int row [, mixed field])
```

The input parameter `result_id` refers to a value returned by `mysql_query()`.

Following:

- Field offset in the table.
- Field name
- Field name specified in dot format. Dot format is simply another way to specify the field name, specified as `fieldname. tablename`.

SYSTEM DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be

free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

DATABASE DESIGN

User Authentication

Column	Type
Id	int(11)
Password	varchar(128)
last_login	datetime(6)
is_superuser	tinyint(1)
Username	varchar(150)
first_name	varchar(30)
last_name	varchar(150)
Email	varchar(254)
is_staff	tinyint(1)
is_active	tinyint(1)
date_joined	datetime(6)

Email Spam Model

Column	Type
Id	int(11)

Column	Type
spam_category	varchar(250)
spam_list	varchar(250)

User Feedback

Column	Type
Id	int(11)
Feedback	varchar(300)
username_id	int(11)

User Registration

Column	Type
Id	int(11)
Firstname	varchar(300)
Lastname	varchar(200)
Userid	varchar(200)
Password	int(11)
Mblenum	bigint(20)
Email	varchar(400)
Gender	varchar(200)

User Send Mail

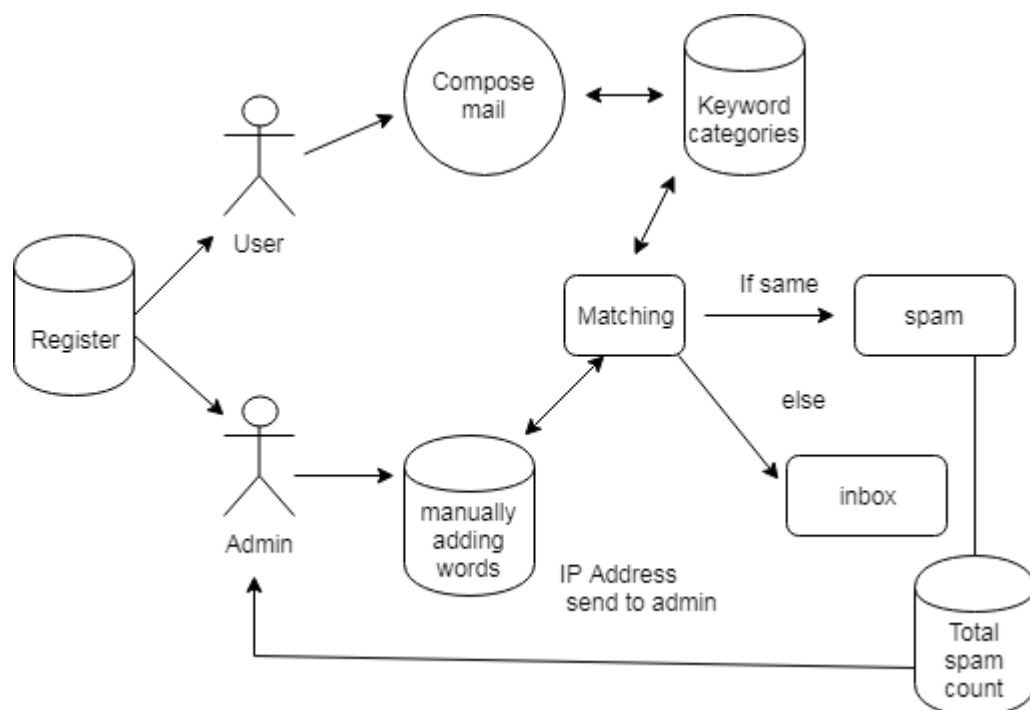
Column	Type
Id	int(11)
Sendermail	varchar(50)
To	varchar(100)
Subject	varchar(100)
Chat	varchar(200)
Spam	varchar(200)
Category	varchar(100)

SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE

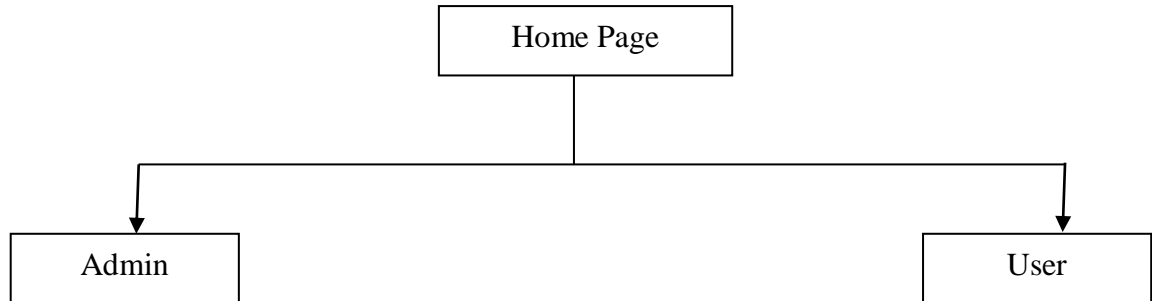
A system architecture or systems architecture is the computational design that defines the structure and/or behavior of a system.

An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed , that will work together to implement the overall system.

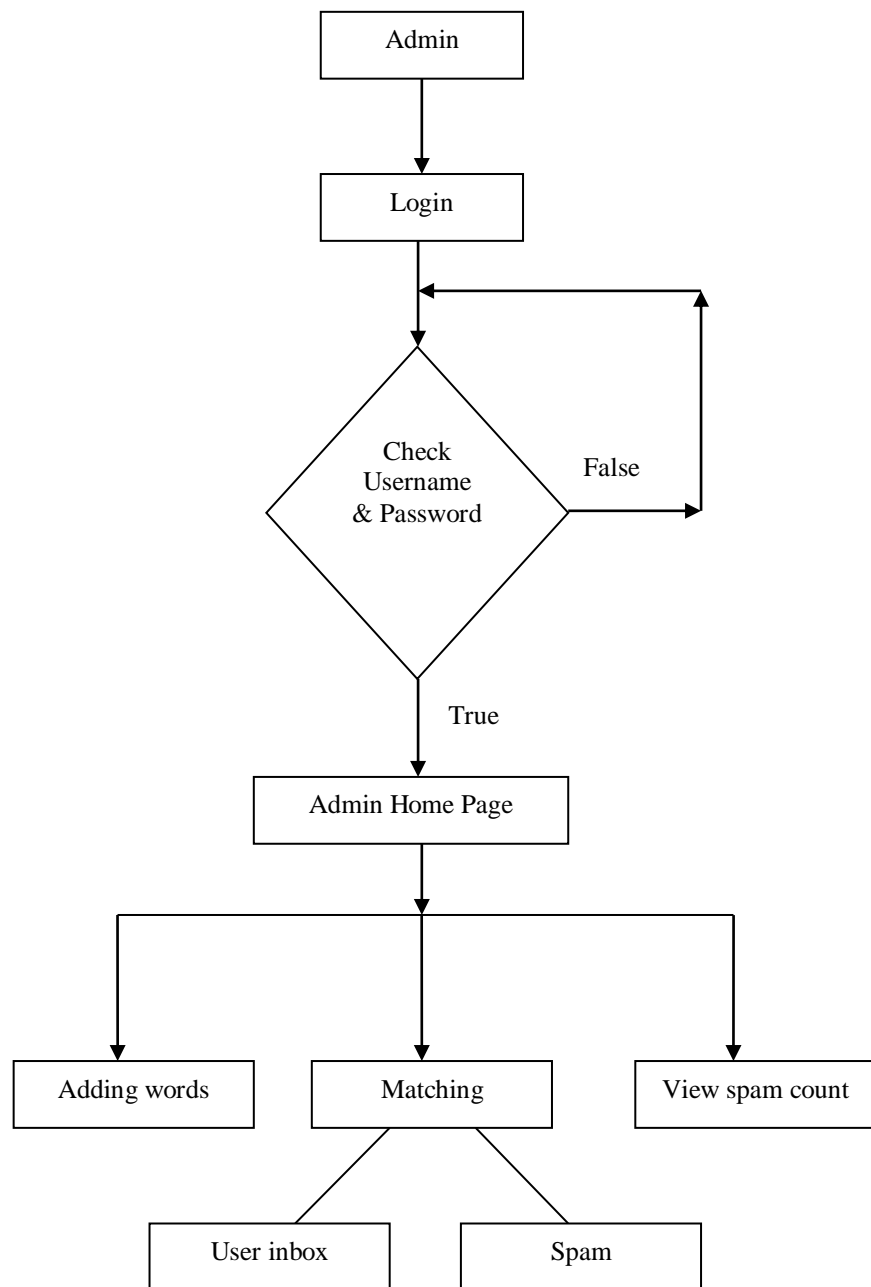


DATA FLOW DIAGRAM

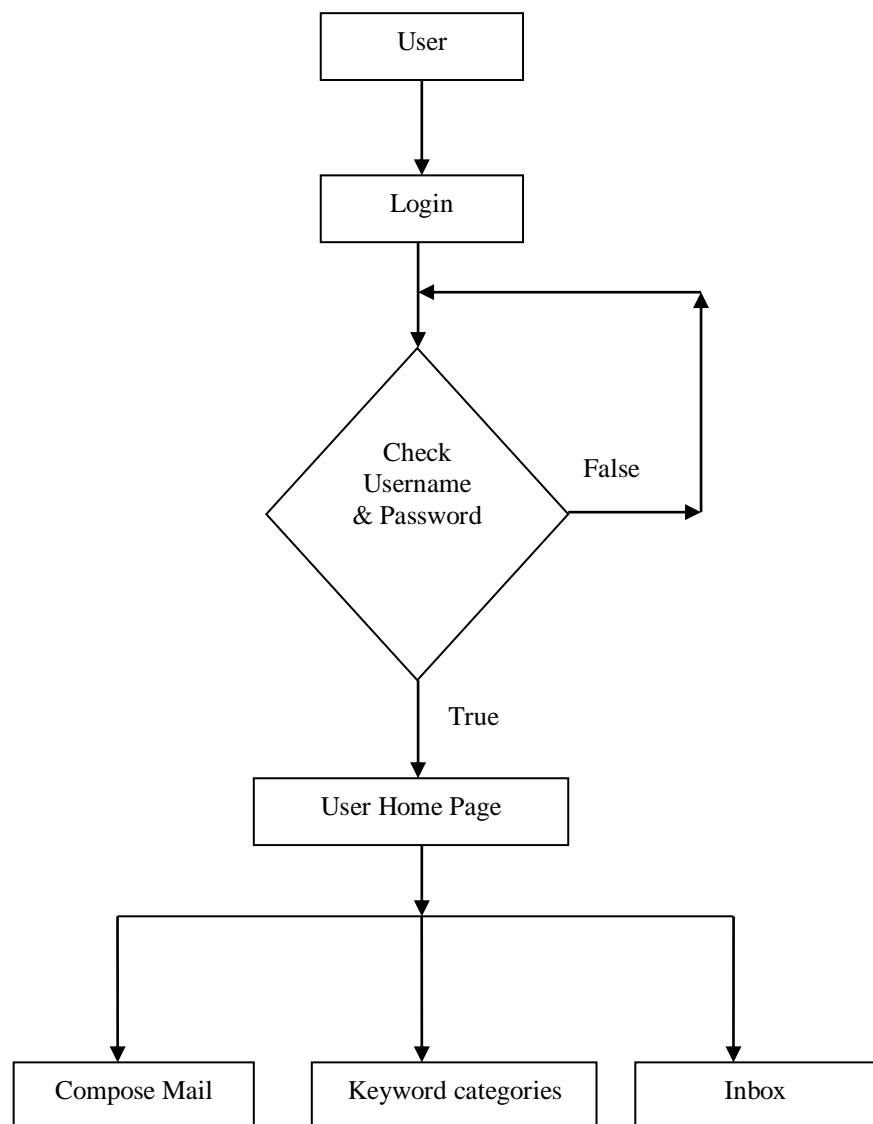
Home Page



Admin Login

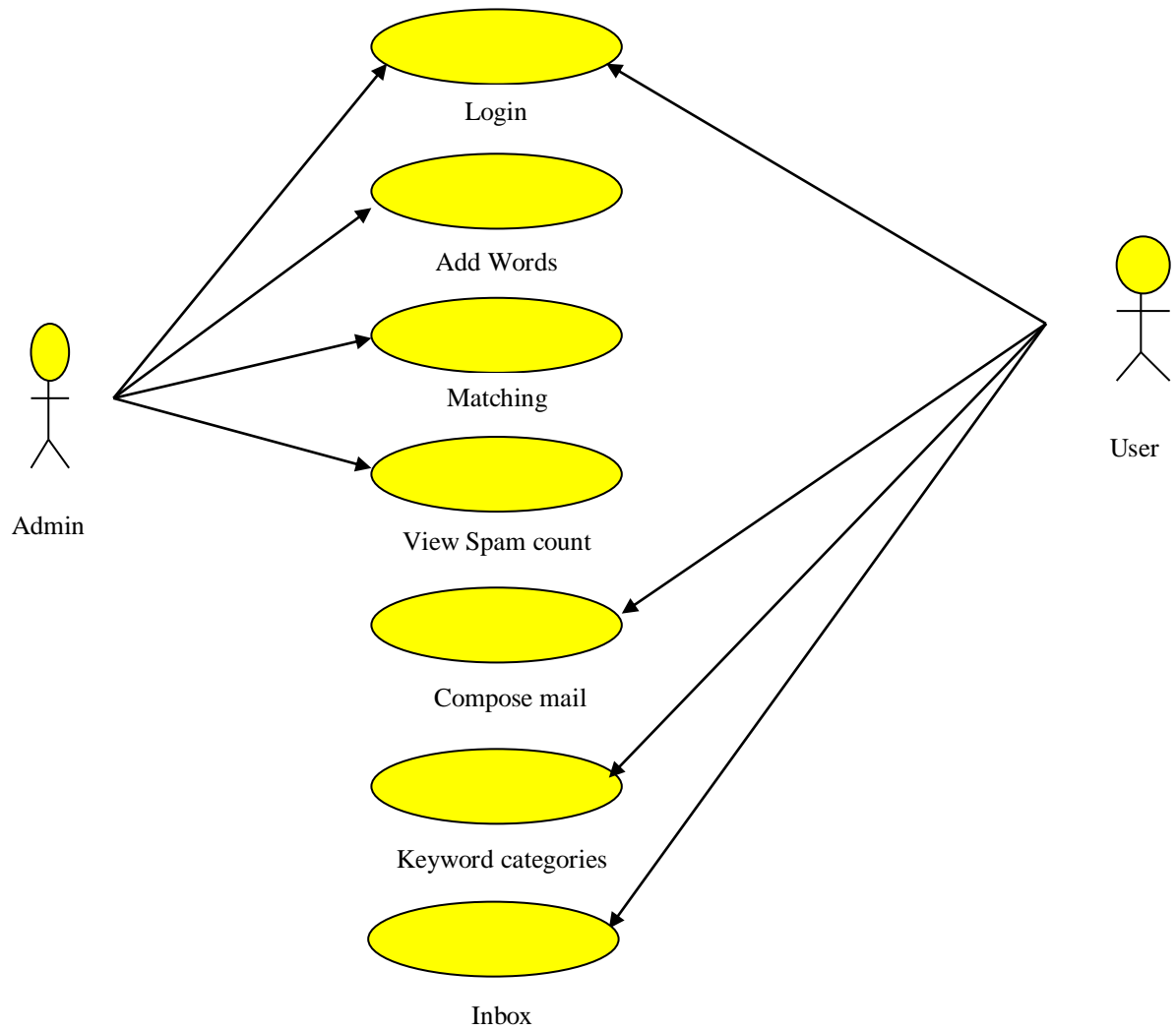


User Login



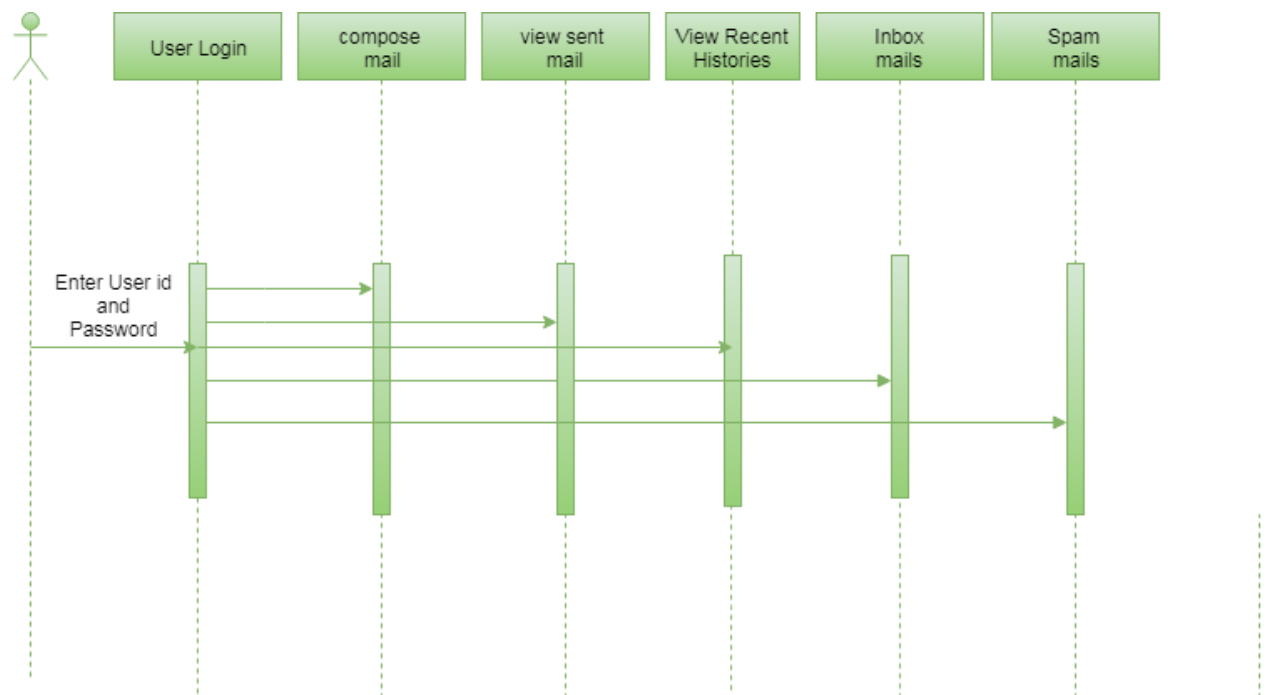
UML DIAGRAM

USE CASE DIAGRAM

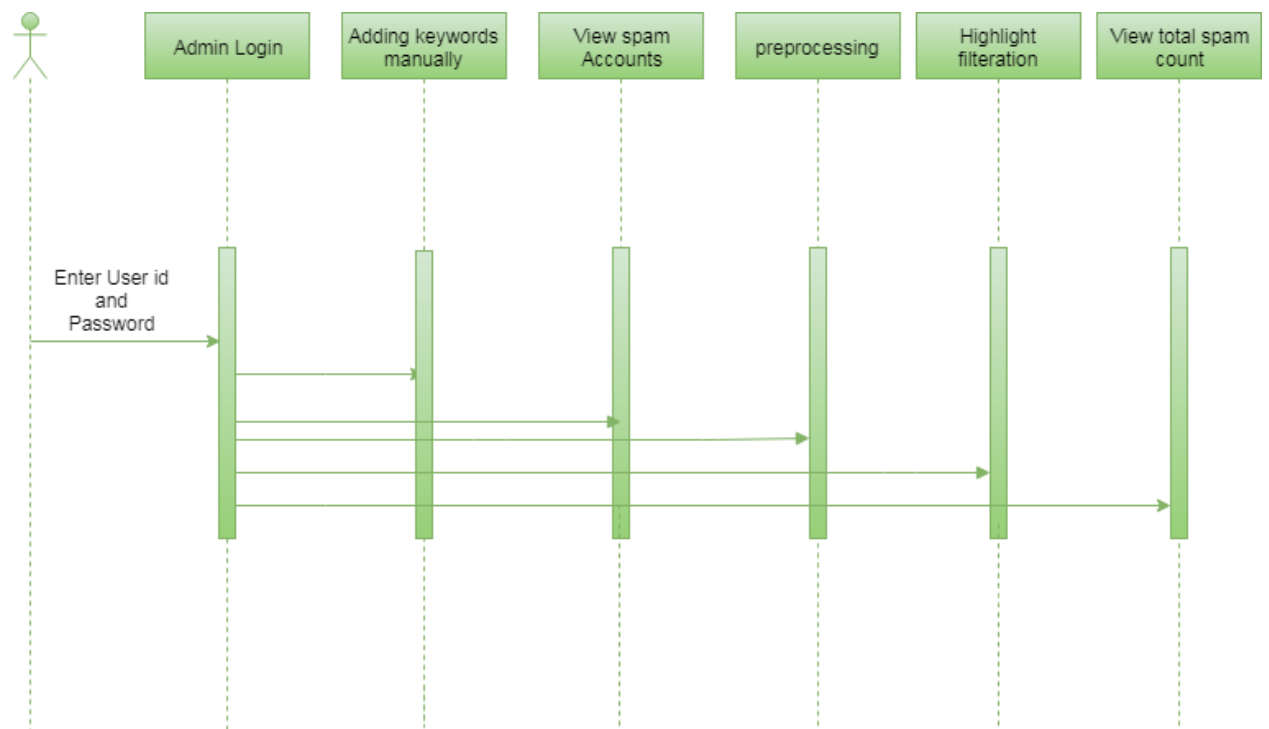


SEQUENCE DIAGRAM

User

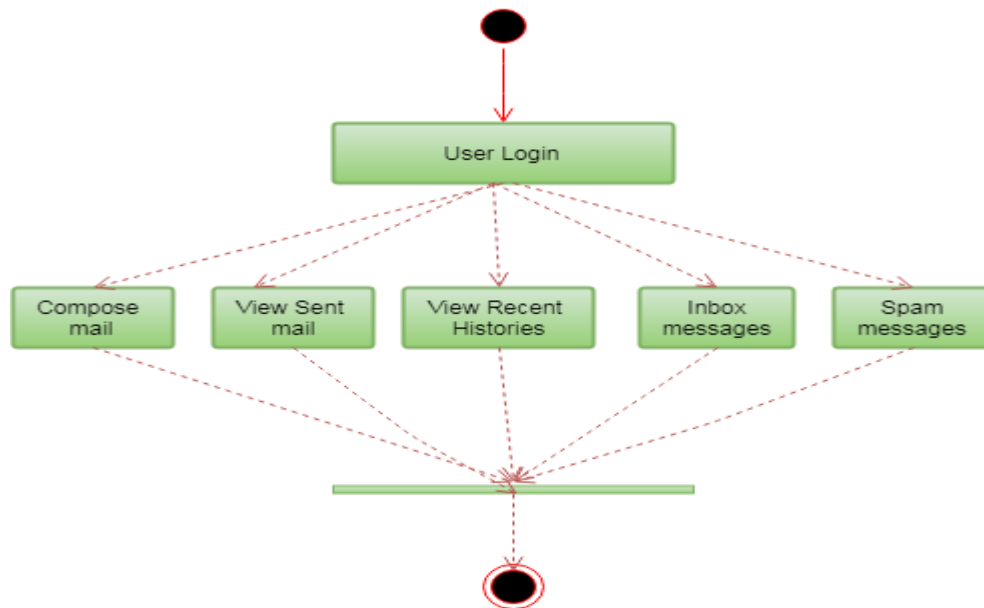


Admin

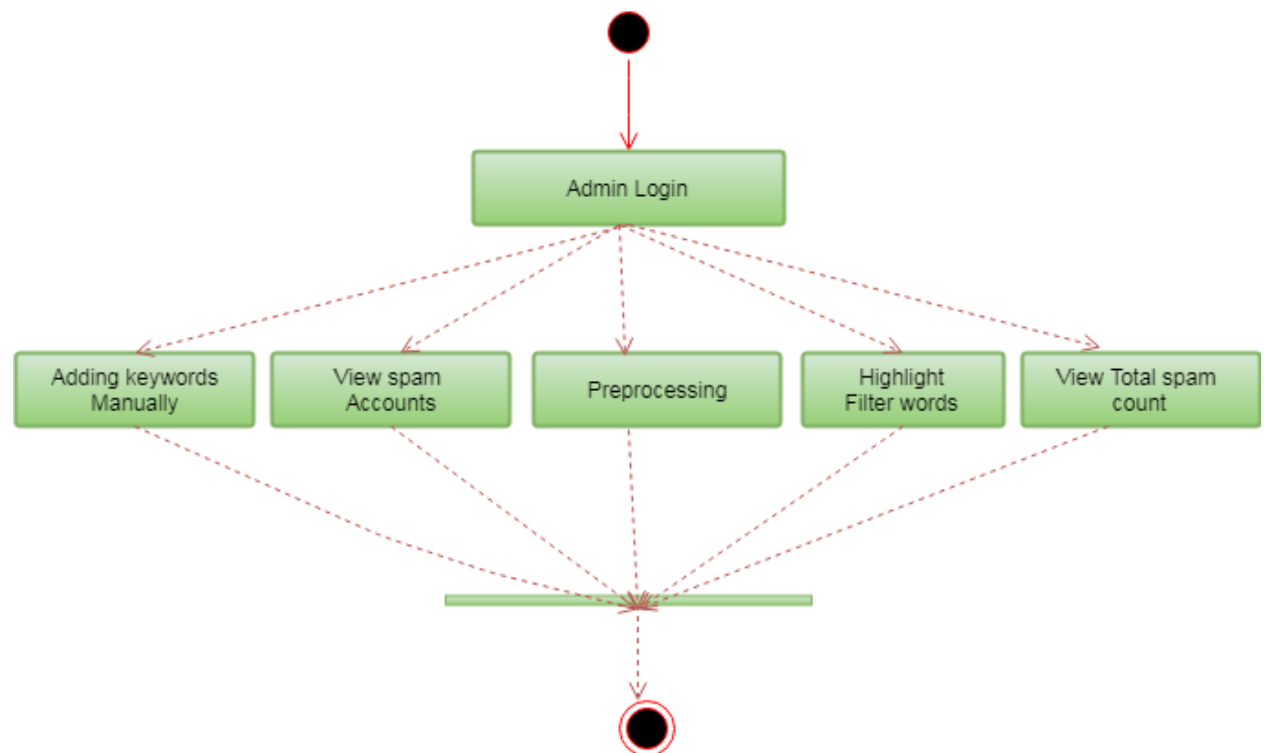


ACTIVITY DIAGRAM

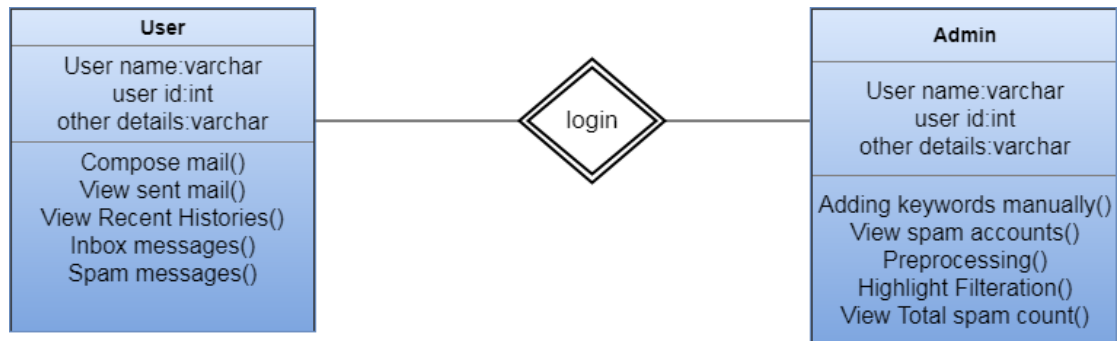
User



Admin



CLASS DIAGRAM



SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the users, what it will work efficient and effectively. It involves careful planning, investing of the current system, and its constraints on implementation, design of methods to achieve the change over methods.

The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out in these plans; discussion has been made regarding the equipment, resources and how to test activities.

The coding step translates a detail design representation into a programming language realization. Programming languages are vehicles for communication between human and computers programming language characteristics and coding style can profoundly affect software quality and maintainability. The coding is done with the following characteristics in mind.

- Ease of design to code translation.
- Code efficiency.
- Memory efficiency.
- Maintainability.

The user should be very careful while implementing a project to ensure what they have planned is properly implemented. The user should not change the purpose of project while implementing. The user should not go in a roundabout way to achieve a solution; it should be direct, crisp and clear and up to the point.

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 its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

MODULES:

Mailing

First, User should Register with their basic details through create an account link. By using that details they need to Login for enter into the system. Then they will receive the message of “success”. Here, we are using the system like E-mail. Hence, it contain the features of inbox, sent mail, spam, recent histories, etc., The user can compose the mail with whom to sent. It may be related to terrorism or may something related to common things. Here, the recent history denotes the person who is doing mail recently.

Filtering

In this Module, I have a few data's in my Dataset. With that, I will check whether the sent message have contain the filtration words about terrorism or not? I have using Data mining technique to mine out text data from large data sets and make the most use of obtained results. Web mining consists of text mining methodologies. Through that text mining, we can extract the text or content what are all related to

terrorism. If the filtration words are match with the sent message means, the receiver receives the mail in his/her spam box or else inbox.

Spam Detection

In this Module, Admin should login first. It will contain the predefined user name and password. Admin side, it will have the features of keywords, spam, analysis, chart. By using Mining concepts Administrator can add few terrorism related words manually in few parameters/ categories. That keywords will also going to add with the existing dataset . In spam, we can see what are all spam messages from starting. In analysis, It contains a mail having how many words in those keyword categories and their total count per each mail.

Preprocessing

In this Module, Admin can see all the spam mail sent and receive in this system, whereas, Spam Detection will contain preprocessing which means it will remove all the common words/stop words such as the, and, or, here, there, etc., Here. I have used the Naïve Bayes algorithm. After preprocessing I have highlight the filtration words in mails. Then it contains every categories count as total spam Detection count. Finally by make use of the total spam Detection count, did the chart.

SYSTEM TESTING

System Testing is an important stage in any system development life cycle. Testing is a process of executing a program with the intention of finding errors. The importance of software testing and its implications with respect to software quality cannot be overemphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. A good test case is one that has a high probability of finding a yet undiscovered error.

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.

Testing is the set of activities that can be planned in advance and conducted systematically. Different test conditions should be thoroughly checked and the bugs detected should be fixed. The testing strategies formed by the user are performed to prove that the software is free and clear from errors. To do this, there are many ways of testing the system's reliability, completeness and maintainability.

The important phase of software development is concerned with translating the design specification into the error-free source code. Testing is carried out to ensure that the system does not fail, that it meets the specification and it satisfies the user. The system testing was carried out in a systematic manner with a test data containing all possible combinations of data to check the features of the system. A test data was prepared for each module, which took care of all the modules of the program.

System Testing is an important stage where the system developed is tested with duplicate or original data. It is a process of executing a program with the intent of finding an error. It is a critical process that can consume fifty percent of the development time.

The following are the attributes of good test:

- A good test is not redundant.
- A good test should be "best of breed".
- A good test should be neither simple nor too complex.

UNIT TESTING

In the unit testing the analyst tests the program making up a system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system, many modules on different levels are needed.

Unit testing can be performed from the bottom up starting with the smallest and lowest level modules and proceeding one at a time. For each module in a bottom-up testing, a short program executes the module and provides the needed data.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associate with interfacing. Objectives are used to take unit test modules and built program structure that has been directed by design.

The integration testing is performed for this Project when all the modules where to make it a complete system. After integration the project works successfully.

VALIDATION TESTING

Validation testing can be defined in many ways, but a simple definition is that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists.

- The functions or performance characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered and a deficiency list is created.

Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

For example, in this project validation testing is performed against module. This module is tested with the following valid and invalid inputs for the field id.

WHITE BOX TESTING

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds and
- Exercise internal data structure to assure their validity.

For example in this project white box testing is performed against patient module. Without entering text if we apply it displays the message “First add record then save it” else it should be saved.

BLACK BOX TESTING

This method treats the coded module as a black box. The module runs with inputs that are likely to cause errors. Then the output is checked to see if any error occurred. This method cannot be used to test all errors, because some errors may depend on the code or algorithm used to implement the module.

4.ADVANTAGES AND DISADVANTAGES :

ADVANTAGES

1. Increased Efficiency : Spam Filtering in machine learning can automate the task of filtering spam emails or messages, thereby increasing the efficiency of the process.
2. Improved and Accuracy : Machine learning algorithms can learn from previous data and improve their accuracy overtime. This improves the accuracy of spam filtering which can reduce the number of false positive and negatives.
3. Cost-effective : Manual spam filtering can be very costly as its requires huge manpower and knowledge of the latest phishing techniques. Automated machine learning spam filtering can be a cost effective solution, especially for business.
4. Enhance User Experience : A spam-free inbox is better user experience. By using machine learning algorithms for spam filtering , users can focus more on important emails which enhance their productivity .

DISADVANTAGES :

1. Possibility of Misclassification : Machine learning algorithms may sometimes categorized genuine or important emails as spam, causing them to be filtered out. This can be frustrating for users and can result in a loss of important communication.
2. Dependence on Data Availability : Machine learning algorithm depend solely on data availability. If there isn't enough data available or the data is flawed, the algorithms will not be able to learn properly, leading to inefficient spam filtering.
3. Complicated Configuration : A high level of technical expertise is required to configure, train, and maintain machine learning algorithms. This can be a disadvantage for organizations that don't have skilled resources or the budget to hire expensive IT professionals.
4. Susceptibility to Algorithm Bias : Machine learning algorithms can sometimes introduce a bias based on the data they were trained on. For example, If an algorithm was trained using biased data, it may be biased against certain demographics or groups, which can be harmful if used for spam filtering.

5.CONCLUSION

To curb and destroy the terrorism and spreading of their activities through online social media through unwanted messages and images to cover the helpless people, we need to use the powerful method or system. That system should be useful to the cops for easily give awareness to common people and find the person who are spreading the harmful words as well as who are all involved in terrorism.

6.FUTURE ENHANCEMENT

We plan to write a simulator that allows accurate and parameterizable simulation of attacks. We also plan to develop analytical models of cyber behavior. These simulations and modeling techniques can then be used to investigate new countermeasures and more comprehensive solutions.

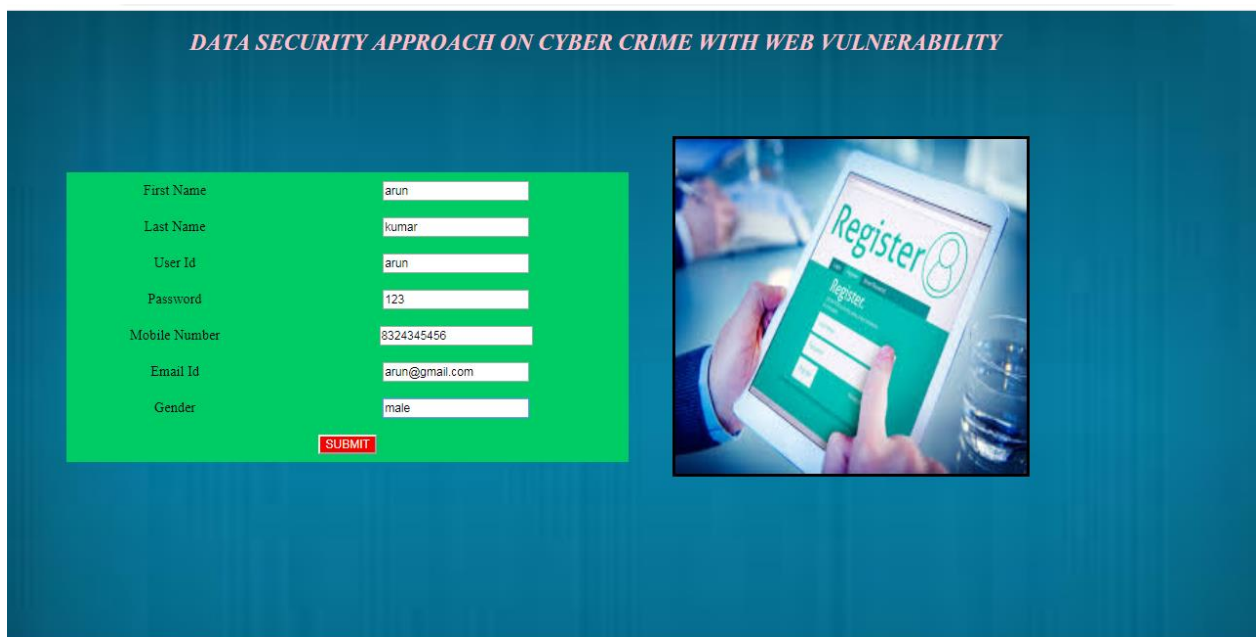
7.APPENDIX -1.1

SCREENSHOTS

Home Page



User Registration



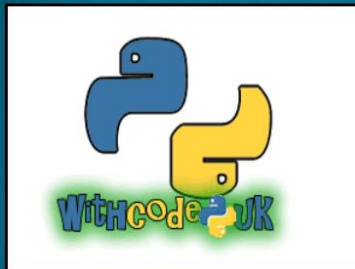
DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

First Name	<input type="text" value="mano"/>
Last Name	<input type="text" value="bala"/>
User Id	<input type="text" value="bala"/>
Password	<input type="text" value="345"/>
Mobile Number	<input type="text" value="9823746343"/>
Email Id	<input type="text" value="bala@gmail.com"/>
Gender	<input type="text" value="male"/>
<input type="button" value="SUBMIT"/>	



1User Login

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY



<input type="text" value="arun"/>	<input type="button" value="✉"/>
<input type="password"/>	<input type="button" value="🔒"/>
<input type="button" value="LOGIN HERE"/>	
<input type="button" value="NEW USER"/>	



User Page

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

MY DETAILSHOME PAGEUPDATE DETAILSFEEDBACKLOGOUT

COMPOSE

INBOX

SPAM



User Profile

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

MY DETAILSHOME PAGEUPDATE DETAILSFEEDBACKLOGOUT

First Name

arun

Last Name

kumar

Userid

arun

Mobile Number

8324345456

Email

arun@gmail.com

Gender

male



Update Profile

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

MY DETAILS HOME PAGE UPDATE DETAILS FEEDBACK LOGOUT

FIRST NAME	<input type="text" value="arun"/>
LAST NAME	<input type="text" value="kumar"/>
USER ID	<input type="text" value="arun"/>
PASSWORD	<input type="text" value="123"/>
MOBILE NUMBER	<input type="text" value="8324345450"/>
EMAIL ID	<input type="text" value="arun@gmail.com"/>
GENDER	<input type="text" value="male"/>



A green speech bubble graphic with the text "UPDATE YOUR DETAILS!" in white. Inside the bubble is a small illustration of a notepad with a pencil and an '@' symbol.

Feedback

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

MY DETAILS HOME PAGE UPDATE DETAILS FEEDBACK LOGOUT

FEEDBACK



A blue rectangular graphic with the word "feedback" in white. It features several black arrows of varying sizes and directions, some pointing towards the text and others away from it, creating a dynamic visual effect.

Admin Login



Analysis Pack

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY					
USER DETAILS	ANALYSIS PAGE	FEEDBACK	CHART PAGE	LOGOUT	
siva12@gmail.com	sanjai12@gmail.com	ultimatum	An ultimatum is a demand whose fulfillment is requested in a specified period of time and which is backed up by a threat to be followed through in case of noncompliance	Threat	Delete
sanjai12@gmail.com	siva12@gmail.com	demoniac	a goddess with both divine and demoniac qualities	Violence	Delete
sanjai12@gmail.com	siva12@gmail.com	plunder	the commander refused to maintain his troops through pillage and plunder	trafficking	Delete
siva12@gmail.com	siva12@gmail.com	subject	An immigration violation is considered a civil offense. Advocates and police say keeping civil immigration enforcement and criminal police enforcement separate helps police fight crime, because immigr	other	Delete
siva12@gmail.com	siva12@gmail.com	project	we drfthb rftj	other	Delete
sanjai12@gmail.com	siva12@gmail.com	project	hii how are you	other	Delete
siva12@gmail.com	siva12@gmail.com	project	next week start the battle	war	Delete
arun@gmail.com	bala@gmail.com	welcome	hello	other	Delete

Feedback

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

USER DETAILSANALYSIS PAGE**FEEDBACK**CHART PAGELOGOUT

USER NAME	FEEDBACK
sabari	thank you sir
siva	improve the designing process

A blue square graphic with the word "feedback" in white lowercase letters. Four black arrows point towards the center of the square from the top, bottom, left, and right edges.

User Page

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

MY DETAILSHOME PAGEUPDATE DETAILS**FEEDBACK**LOGOUT

COMPOSE

bala@gmail.com

INBOX

war

battle war in pakistan

SPAM

send

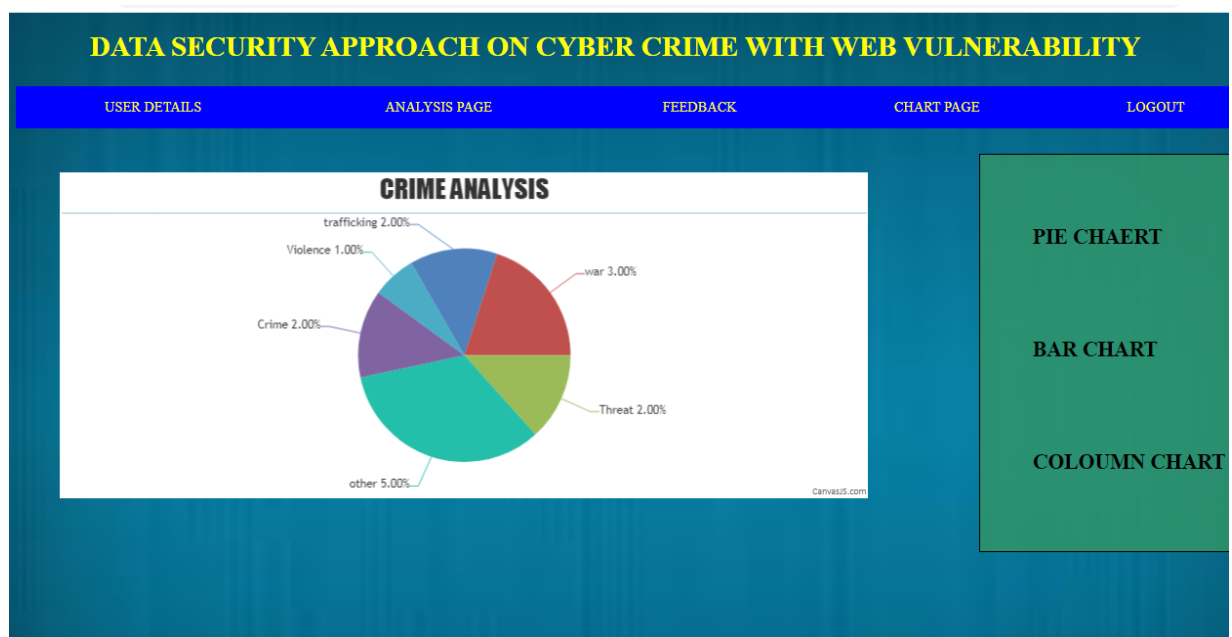
A photograph of a person's hand typing on a laptop keyboard. The laptop screen displays a blue background with several white envelope icons, representing email. The background of the entire page is a dark blue gradient.

41

Admin Page(Analysis Page)

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY						
USER DETAILS	ANALYSIS PAGE		FEEDBACK	CHART PAGE	LOGOUT	
siva12@gmail.com	sanjai12@gmail.com	ultimatum	specified period of time and which is backed up by a threat to be followed through in case of noncompliance	Threat	Delete	
sanjai12@gmail.com	siva12@gmail.com	demoniac	a goddess with both divine and demoniac qualities	Violence	Delete	
sanjai12@gmail.com	siva12@gmail.com	plunder	the commander refused to maintain his troops through pillage and plunder	trafficking	Delete	
siva12@gmail.com	siva12@gmail.com	subject	An immigration violation is considered a civil offense. Advocates and police say keeping civil immigration enforcement and criminal police enforcement separate helps police fight crime, because immigr	other	Delete	
siva12@gmail.com	siva12@gmail.com	project	we drfthb rftj	other	Delete	
sanjai12@gmail.com	siva12@gmail.com	project	hii how are you	other	Delete	
siva12@gmail.com	siva12@gmail.com	project	next week start the battle	war	Delete	
arun@gmail.com	bala@gmail.com	welcome	hello	other	Delete	
arun@gmail.com	bala@gmail.com	war	battle war in pakistan	war	Delete	

Admin Page(chart)



DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

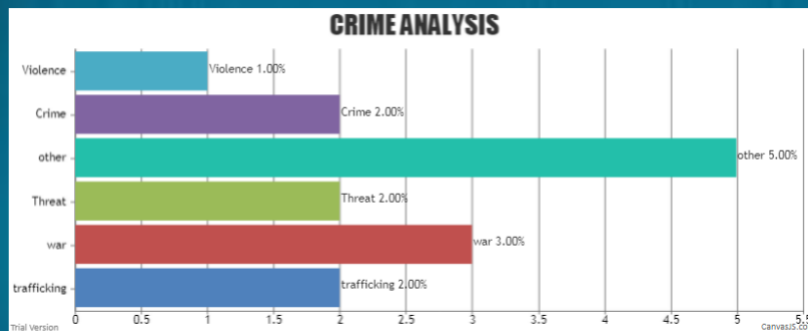
USER DETAILS

ANALYSIS PAGE

FEEDBACK

CHART PAGE

LOGOUT



PIE CHAERT

BAR CHART

COLOUMN CHART

DATA SECURITY APPROACH ON CYBER CRIME WITH WEB VULNERABILITY

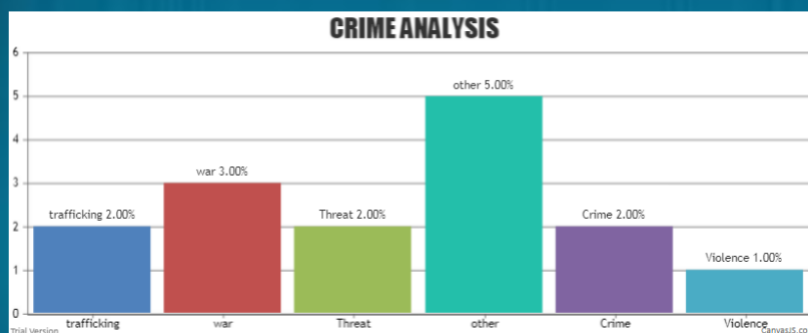
USER DETAILS

ANALYSIS PAGE

FEEDBACK

CHART PAGE

LOGOUT



PIE CHAERT

BAR CHART

COLOUMN CHART

APPENDIX -1.2

CODING

```
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/


# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '%@9-_7#_bluk_7%b-18(ltl-q!(+xv)p-=q)rircbb9%dl*501'


# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True


ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',
```

```
'django.contrib.messages',  
'django.contrib.staticfiles',  
'user',  
'email_admin',  
]
```

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

```
ROOT_URLCONF = 'email_spam_detection.urls'
```

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [(os.path.join(BASE_DIR, 'assets/templates'))],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',
```

```

'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]

WSGI_APPLICATION = 'email_spam_detection.wsgi.application'

# Database

# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'gmail_spam',

'USER': 'root',

'PASSWORD': '',

'HOST': '127.0.0.1',

'PORT': '3306',

}

}

# Password validation

# https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',

```

```
},  
  
{  
    'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',  
},  
  
{  
    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',  
},  
  
{  
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',  
},  
}
```

Internationalization

<https://docs.djangoproject.com/en/1.11/topics/i18n/>

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

Static files (CSS, JavaScript, Images)

<https://docs.djangoproject.com/en/1.11/howto/static-files/>

```
STATIC_URL = '/static/'
```

```
STATICFILES_DIRS=[os.path.join(BASE_DIR,'assets/static'),]
```

```
MEDIA_URL='/media/'
```

```
MEDIA_DIR=os.path.join(BASE_DIR,'assets/media')
```

"""email_spam_detection URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/1.11/topics/http/urls/>

Examples:

Function views

1. Add an import: from my_app import views
2. Add a URL to urlpatterns: `url(r'^$', views.home, name='home')`

Class-based views

1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: `url(r'^$', Home.as_view(), name='home')`

Including another URLconf

1. Import the `include()` function: from `django.conf.urls` import `url`, `include`
2. Add a URL to urlpatterns: `url(r'^blog/', include('blog.urls'))`

"""

```
from django.conf.urls import url
```

```
from django.contrib import admin
```

```
from user import views as user_views
```



```

from email_admin import views as admin_views

urlpatterns = [

url(r'^admin/', admin.site.urls),

url('^$',user_views.index,name="index"),

url('user/register', user_views.register, name="register"),

url('user/userpage',user_views.userpage,name="userpage"),

url('user/viewmailpage',user_views.viewmailpage,name="viewmailpage"),

url('user/spampage',user_views.spampage,name="spampage"),

url('user/logout/$', user_views.logout, name='logout'),

url('user/delete/(?P<pk>\d+)/$',user_views.deleteobj,name="deleteobj"),

url('user/spamdelete/(?P<pk>\d+)/$',user_views.spamdeleteobj,name="spamdeleteobj

"),

url('user/mydetails',user_views.mydetails,name="mydetails"),

url('user/updatemydetails',user_views.updatemydetails,name="updatemydetails"),

url('user/feedback',user_views.feedback,name="feedback"),

url('login',admin_views.login,name="login"),

url('admin_page',admin_views.admin_page,name="admin_page"),

url('analysis_page',admin_views.analysis_page,name="analysis_page"),

url('categoryanalysis/(?P<chart_type>\w+)',admin_views.categoryanalysis_chart,nam

e="categoryanalysis_chart"),

url('admin/delete/(?P<pk>\d+)/$',admin_views.analysisdelete,name="analysisdelete"),

url('admin/viewfeedback', admin_views.viewfeedback, name="viewfeedback"),

]

```