Software Design Document

for

<Language Filter and Abuse Detection System>

Sehrish Aslam(EP-191901073)
Kiron Ismail(Ep-19101031)
Syeda Fatima Ahsan(EP-19101085)

CONTENTS

1 Introduction

The purpose is to create a language filter and abusive word detection software for online social media platforms like Facebook, Instagram etc. The system will be able to filter out abusive words and post thereby creating a safe and clean environment for the online users.

This design document presents the designs used or intended to be used in implementing the project. The designs described, follow the requirements specified in the Software Requirements Specifications document prepared for the project.

1.1 Purpose

The purpose of this document is to present a detailed description of the designs of the Language Filter and Abuse Detector System created for the online social media platforms. The document is intended for the developers, designers and the programming team to use the designs as guidelines to implement the project. Developers are supposed to use this document in the development phase for the structure and design of each component. Lastly this document could be used for designers and the maintenance team who try to upgrade or modify the present design of the inventory system.

1.2 Scope

This document gives a detailed description of the software architecture of the language filter and abuse detection system. It specifies the structure and design of some of the modules discussed in the SRS. It also displays the class diagram and use cases that are transformed into sequential and activity diagrams. The class diagram shows how the language filter modules and report generation modules are linked and would be later implemented.

1.3 References

The user of the SDD may need the following documents for the reference:

IEEE Standard 1016-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

The User Requirement document, URD. Last modified: Apr. 11, 2020

The Software Requirement Specification document, SRS. Last modifies Apr. 11, 2020.

1.4 Definitions and Acronyms

SDD	Software Design Document
SRS	Software Requirement Specification

URD	User Requirement Document
LFADS	Language Filter and Abuse Detection System

1.5 Overview

This document is written according to the standards for Software Design Documentation explained in "IEEE Recommended Practice for Software Design Documentation". Sections 3 – 5 contain discussions of the designs for the project with diagrams, section 6 shows samples of UI from the system, and section 7 contains the class diagrams. The appendices contain the setup and configuration needed for the LFAD, a list of functions that are implemented in this version, and that are to be implemented in future version, and a list of tools and environment used in the entire project.

2 System Overview

The LFADS is a software that will use the Graph API (Facebook for developers) to generate user access tokens which will be used to get posts from the token-specific account. Those posts will then be filtered out by matching the content of the post against a list of abusive words.

The IDE used for the software is NetBeans IDE 8.2 version. It will use JAVA Database Connectivity (JDBC) which is an Application Programming Interface (API) for the programming language java which helps a client access a database.

3 Architecture

3.1 Architecture Design

The architectural design of the LFAD consists of two types of architectural design patters. These are **MVC-Model View Controller** and **Pipe and Filter Architecture.** The MVC divides the system into the following modules to achieve the complete functionality.

- **Model**
  The model is linked with the database that contains a list of the abusive words. The database also contains user account information along with the list of posts with abusive words detected.
- **View**
  The view shows the user interface. It allows user to sign in and sign up. The user can input the pre-generated token which is sent to the controller.
- **Controller**

The controller uses the token to fetch the posts from Facebook. These posts are then compared to the abusive-word list which is fetched from the model. The controller further contains the pipe and filter architecture. The pipes send the posts which are filtered out by comparison of the post-content with the abusive word list. The model is updated by the controller to update the abusive word count against the token-specific account.

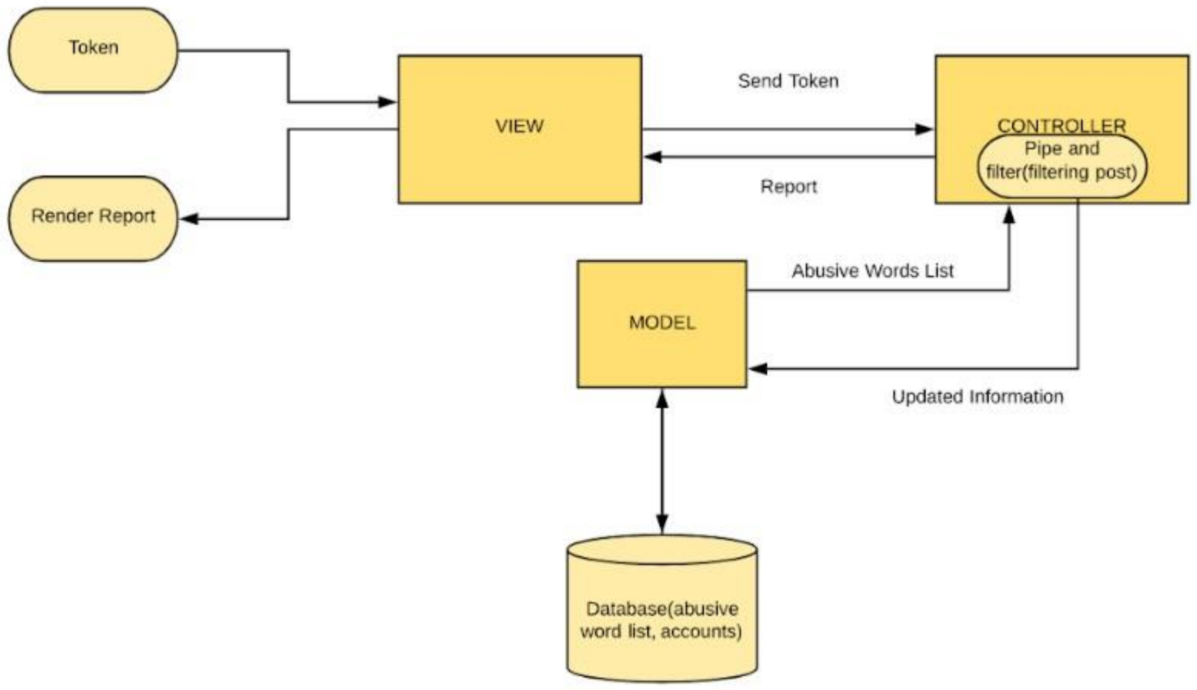


Figure 1: MVC architecture of LFAD

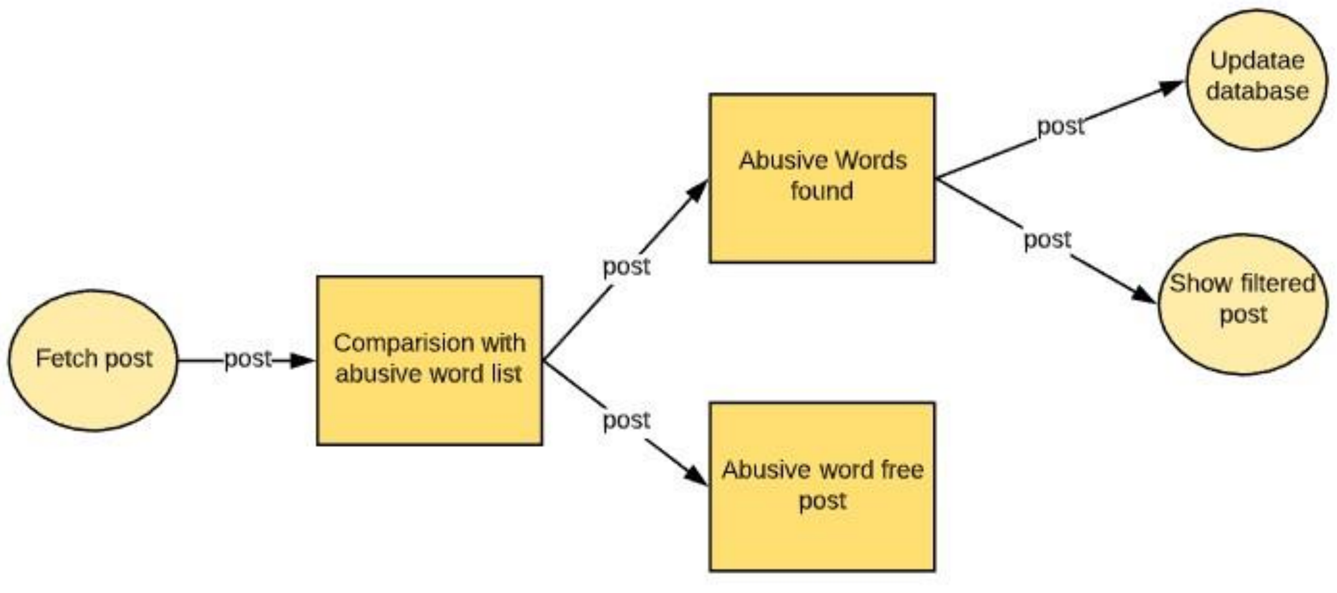


Figure 2 Pipe and Filter

3.2 DECOMPOSITION DESCRIPTION

The decomposition of the system is explained in the following two ways.

3.2.1 Module Decomposition

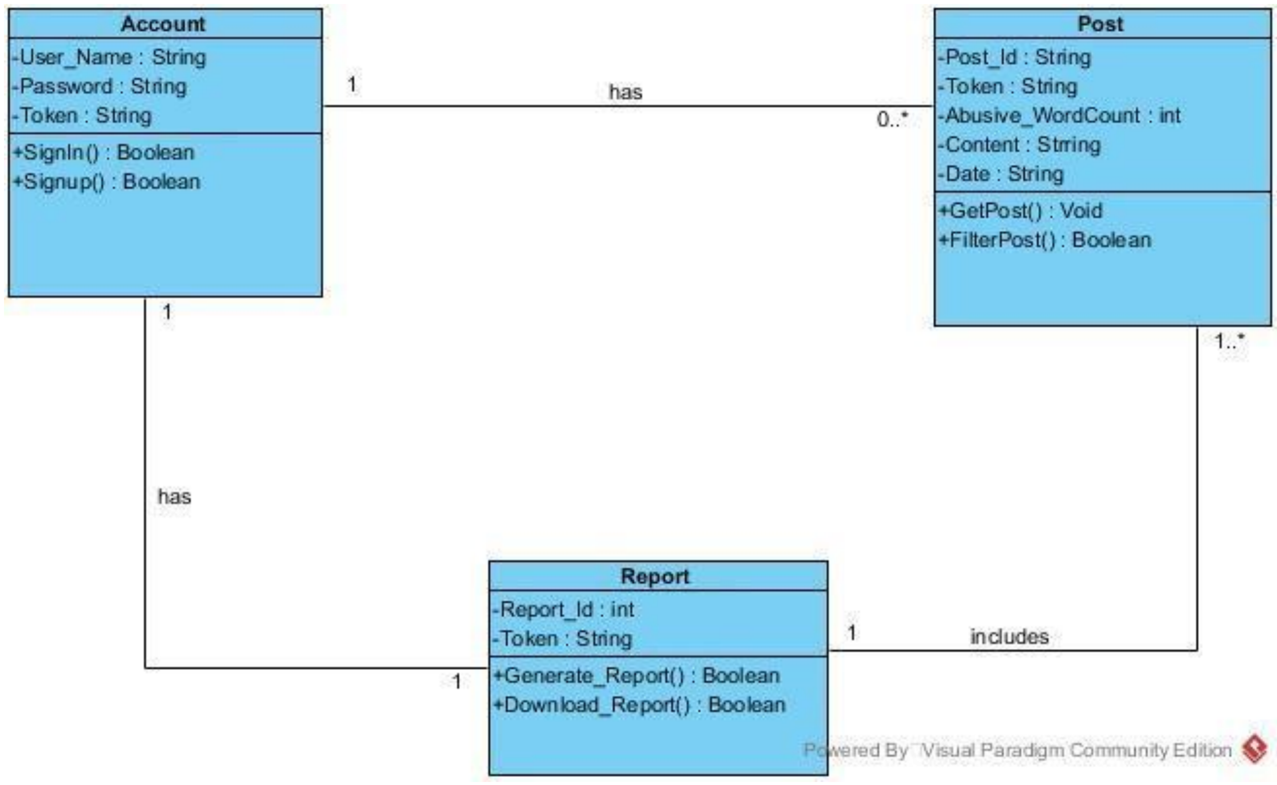


Figure 3 Class Diagram

There are three modules/classes namely Account, Post and Report.

Account allows the user to sign into the system. From there the user can enter a pre-generated token. This token allows the user to access the posts linked to the token-specific account.

Post refers to the posts that are linked to the Facebook account. An account has many posts. If the post is filtered out on the basis of abusive-content. Its abusive – word count and content is noted and saved to the database.

The user can generate a **Report** which will contain report id and information like the abusive content and list of abusive words. The posts will also be displayed to the user in its updated filtered form with abusive -free content.

3.2.2 Process Decomposition

The process decomposition is explained through sequence and use case diagrams which decomposes the system into well-defined and cohesive processes. The use cases explain the set of actions that a user undertakes while dealing with the LFADS.

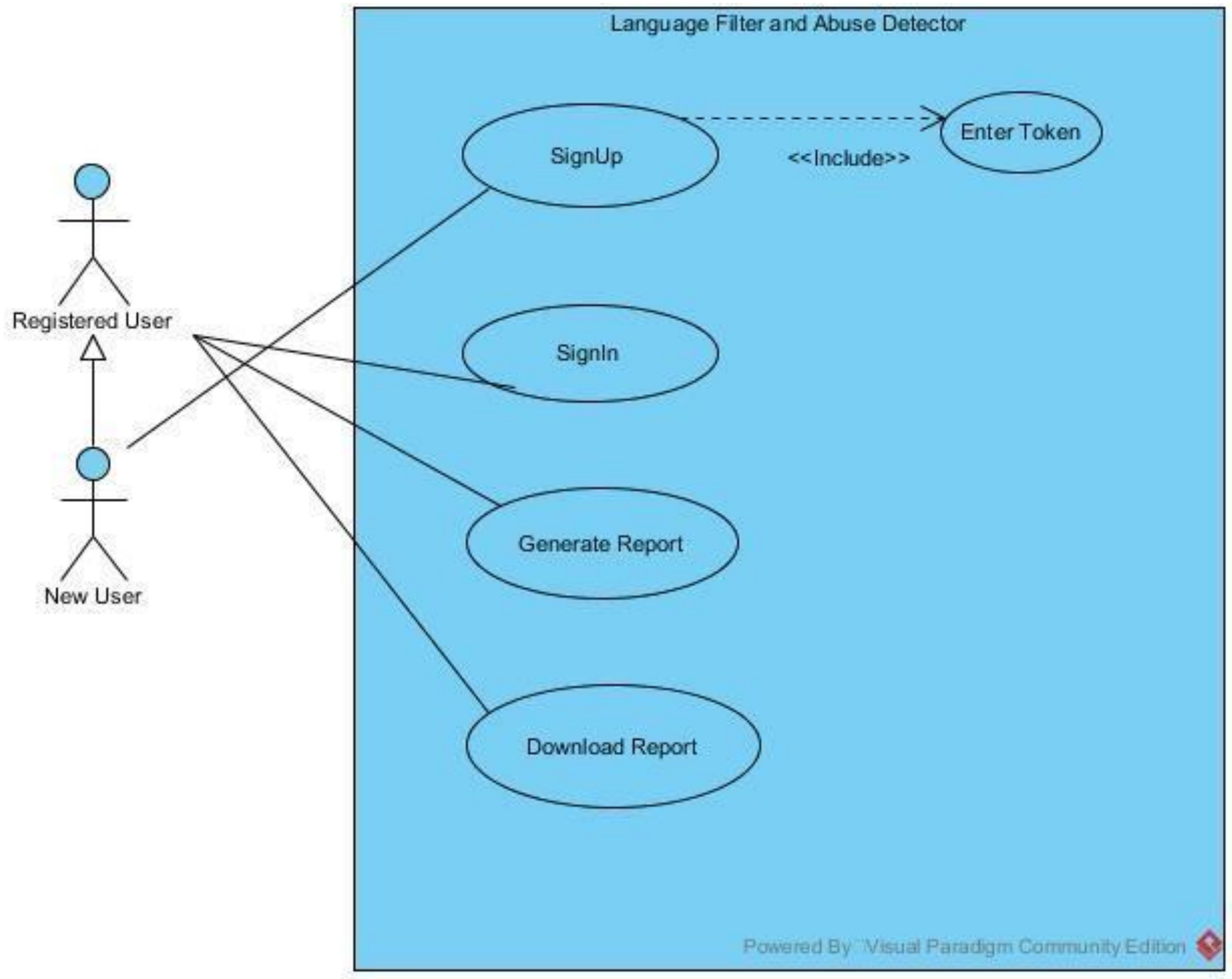
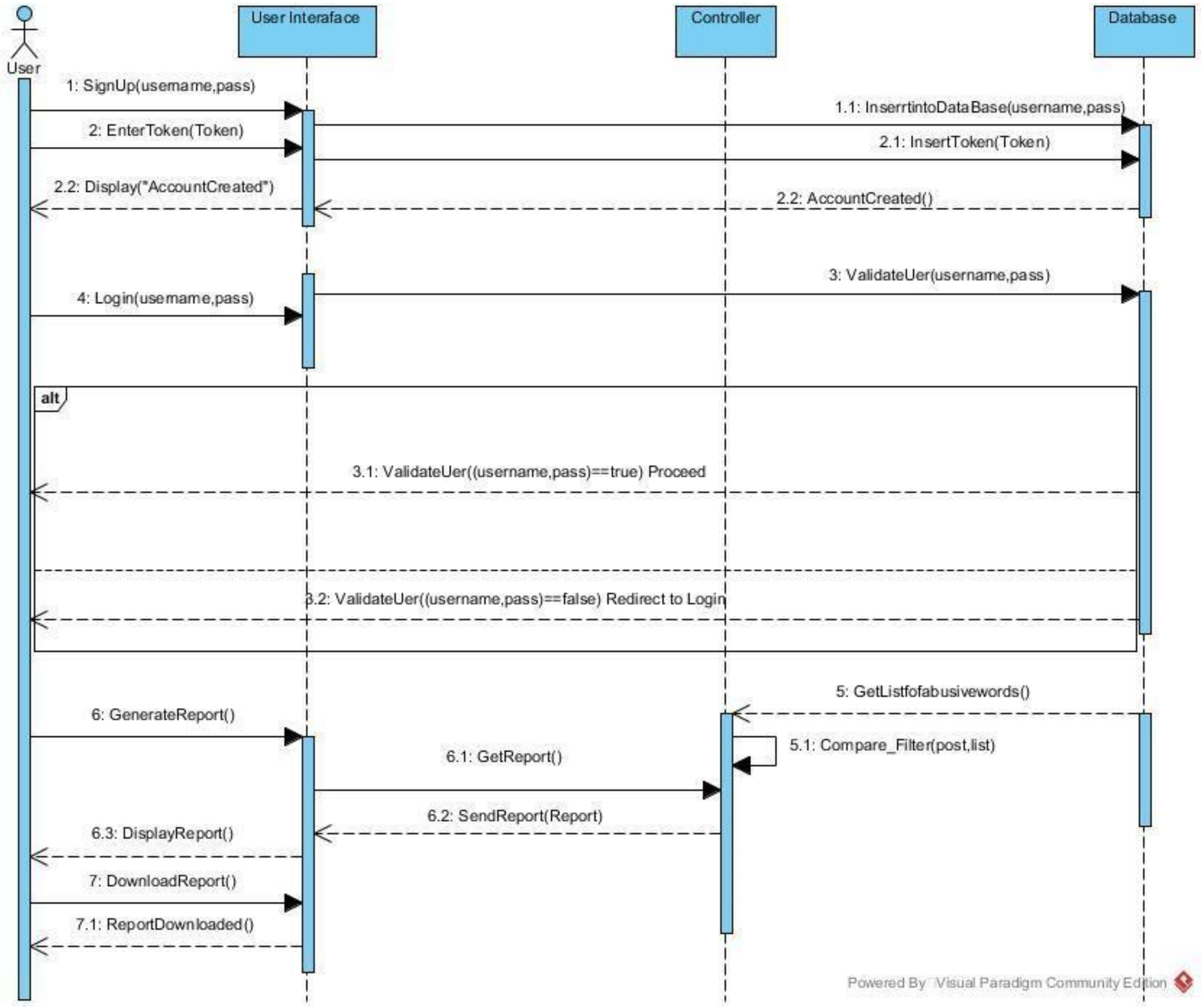


Figure 4 Use Case Diagram

There are two primary actors. Registered and unregistered user. Once registered, the user can sign in. The user can also Generate and download report.

A new and unregistered user must sign up first. After signing up, the user must enter a token. This token gets saved along with the user account. The next time, the user doesn't have to enter the token again and again. This pre-generated token gets linked with the Facebook account.

Figure 5 Sequence Diagram


User
User Interaface
Controller
Database
1: SignUp(usemame,pass)
1.1: InserrtintoDataBase(username,pass)
2: EnterToken(Token)
2.1: InsertToken(Token)
2.2: Display("AccountCreated")
2.2: AccountCreated()
3: ValidateUer(username,pass)
4: Login(usemame,pass)
alt
3.1: ValidateUer((username,pass)==true) Proceed
3.2: ValidateUer((username,pass)==false) Redirect to Login
5: GetListfofabusivewords()
6: GenerateReport()
5.1: Compare_Filter(post,list)
6.1: GetReport()
6.2: SendReport(Report)
6.3: DisplayReport()
7: DownloadReport()
7.1: ReportDownloaded()
Powered By Visual Paradigm Community Edition

The sequence diagram shows the sequence of events and object interactions arranged in time sequence. The use interacts with the user interface to sign up, login and enter the token number. The controller deals with the database and the user interface. The controller gets information from the database and compares the post content. After the data processing, a report is generated and the report is sent to the user interface to be viewed by the user. The report can also be downloaded on the host machine upon the user's request.

3.3 Design Rationale

The architecture chosen for the LFADS is a **Hybrid architecture** consisting of MVC**- Model View Controller** and **Pipe and Filter.**

MVC has been chosen because of separation of concern. Separation of concern means we divide the system into Model, Controller and View. The model only deals with the database and is separated from the view and controller. It does not have to deal with user interface or data processing. The only two things that a View has to do is to show data to the user on User Interface and to respond to the events. In LFADS the view allows the user to enter the username, password and token. It also displays the generated report to the user. The controller makes it easier to perform data-processing and comparison of the post content with the information in the database. It processes the fetched posts and filters them using the pipe and filter architecture.

The pipe and filter architecture is the part of the controller. The rationale behind it is that we input a post into the filter and in output we filter out the posts containing the abusive content. Each filter can be implemented as a separate task and can be executed in parallel with other filters.

4 Data Design

4.1 Data Description

The database used is JAVA DB provided in the Net Beans IDE version 8.2. Java DB is Sun's supported distribution of the open source Apache Derby 100% Java technology database The JDBC driver is used to connect the Java application to the Database.

The database contains:

- Account (User id, password, token)
- Posts (Post id, content, date, Abusive word count)
- Abusive word list

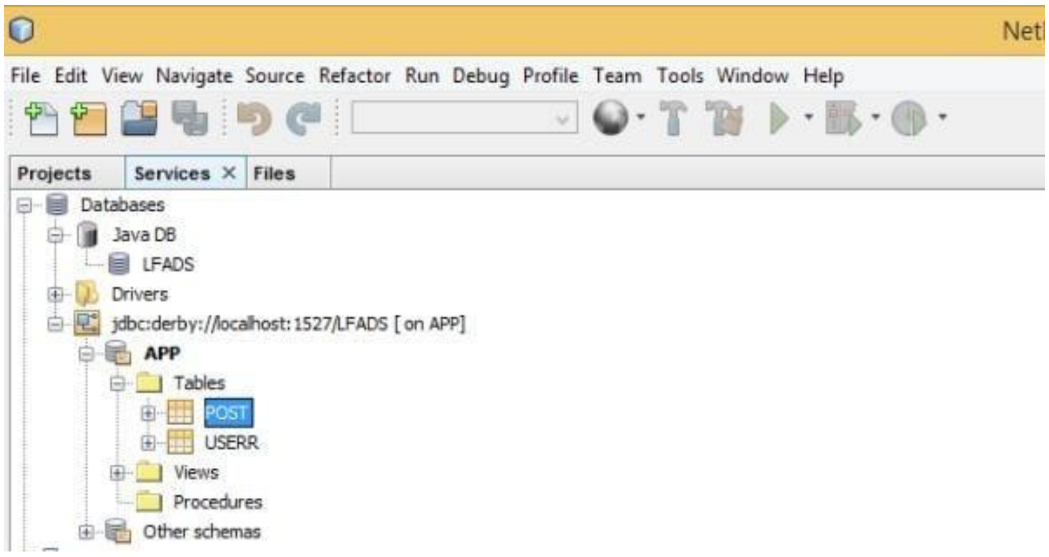

Net
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Services
Files
Databases
Java DB
LFADS
Drivers
jdbc:derby://localhost:1527/LFADS [on APP]
APP
Tables
POST
USERR
Views
Procedures
Other schemas

Figure 6 Net Beans JDB

4.2 DATA DICTIONARY

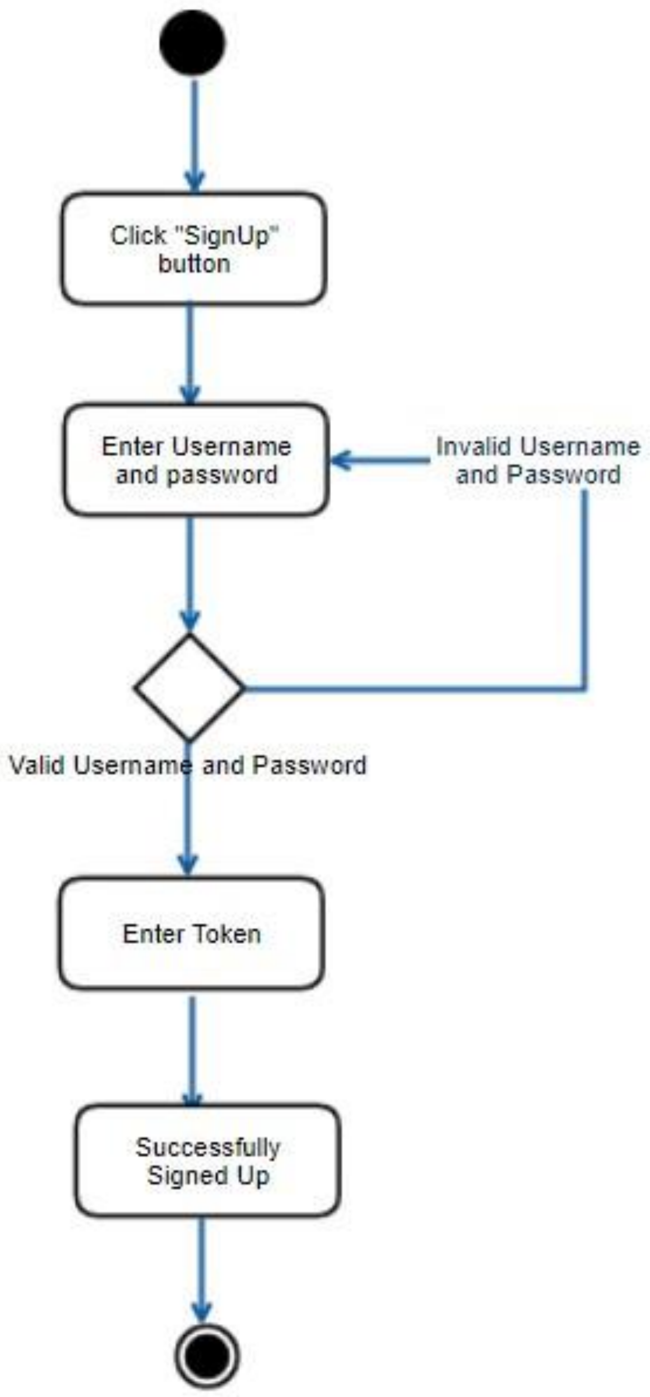
Table	**Field**	**Type**	**Null**
Abusive_words	Words	Varchar (100)	No
Account	Account_id	int	No
	User_Name	Varchar(100)	No
	Password	Varchar(100)	No
	Token	Varchar(150)	No
Post	Post_id	int	No
	Token	Varchar(150)	No
	Content	Long Varchar	Yes
	Abusive_word_count	int	Yes
	Date	date	No

5 Component Design

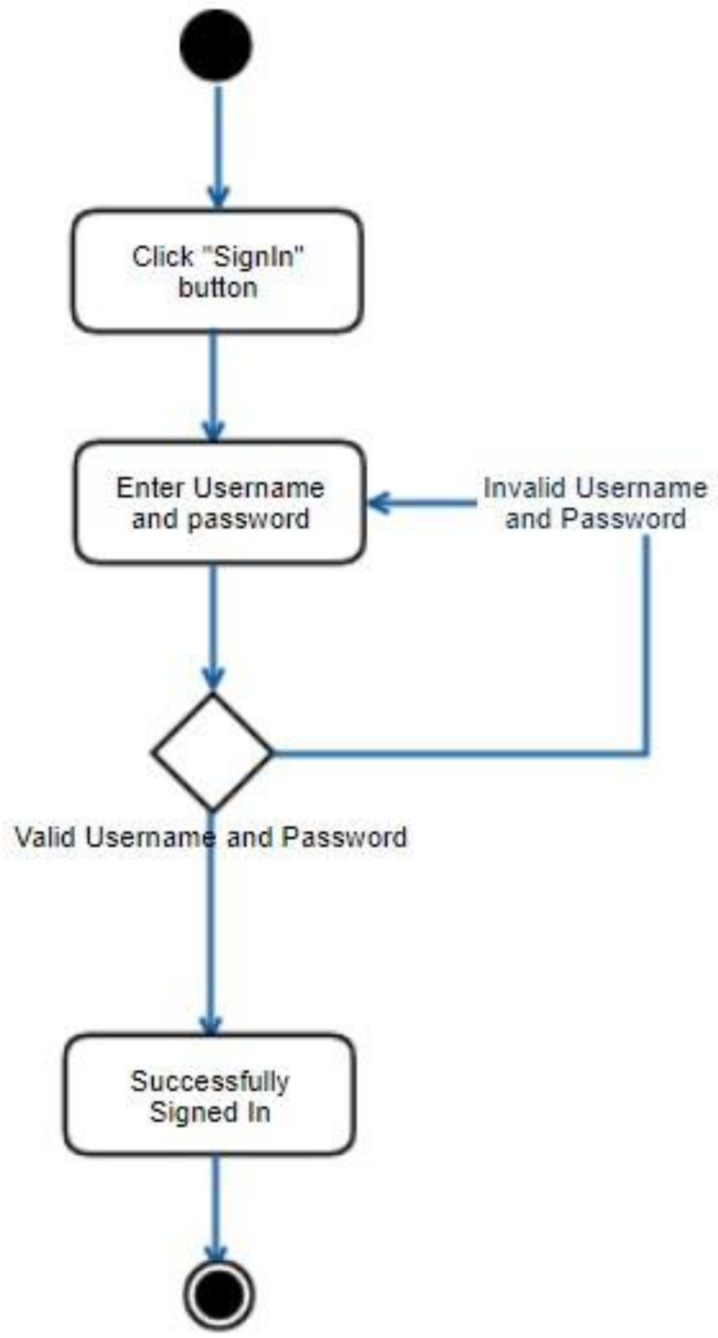
In this section of component design, we will take a closer look at each component of the LFADS in a more systematic way. Each component will have a functional description and closer detail. The components that have been studies below through activity diagram are:

- Sign-up
- Sign-in
- Report Generation
- Download report

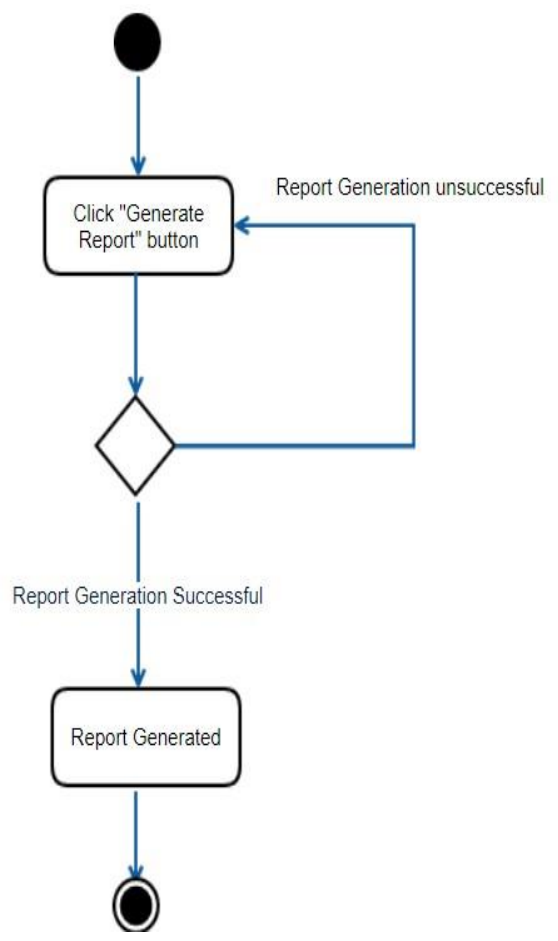
SignUp


Click "SignUp" button
Enter Username and password
Invalid Username and Password
Valid Username and Password
Enter Token
Successfully Signed Up

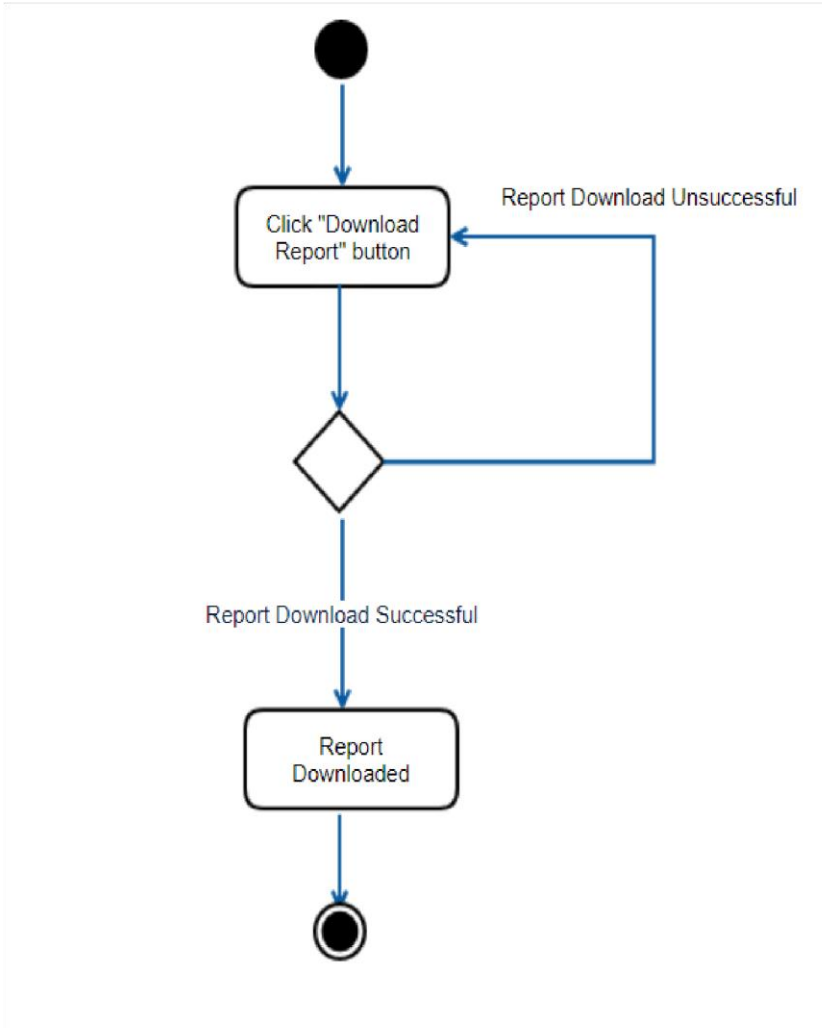
Sign-In


Click "SignIn" button
Enter Username and password
Invalid Username and Password
Valid Username and Password
Successfully Signed In

Report Generation


Click "Generate Report" button
Report Generation unsuccessful
Report Generation Successful
Report Generated

Download Report


Click "Download Report" button
Report Download Unsuccessful
Report Download Successful
Report Downloaded

6 HUMAN INTERFACE DESIGN

6.1 OVERVIEW OF USER INTERFACE

UI is designed according to UI design principles.

The structure principle: UI is organized in such a way that related things are combined together and unrelated things are separated.

The simplicity principle: It is easy to follow the provided interface.

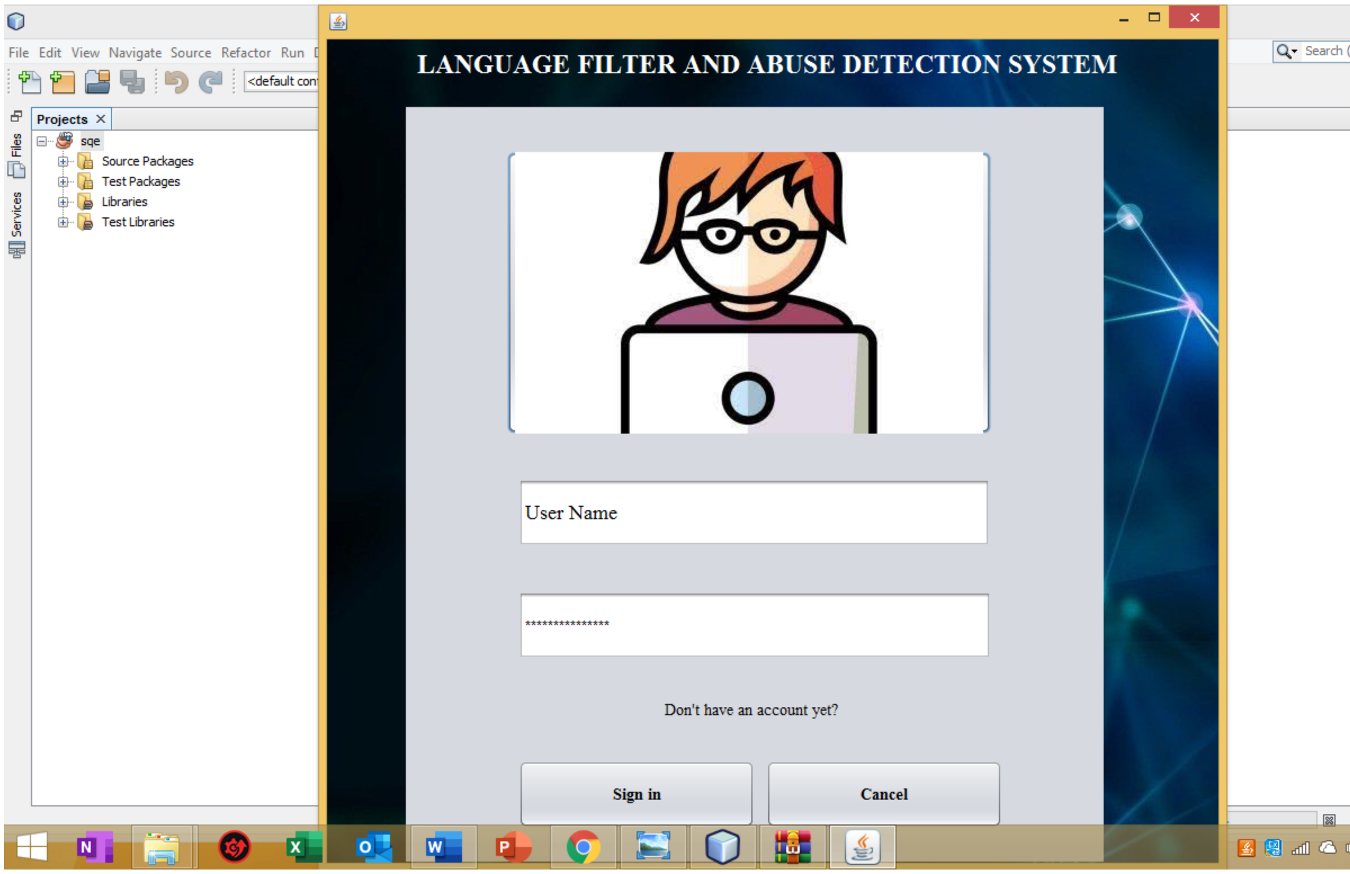
The visibility principle: All system's functions are available through UI. It does not overwhelm users with too many alternatives.

The reuse principle: In design, same names were used to perform the same operations with different objects in order to reduce ambiguity.

There are a total of 4 pages in the user interface. The sign-up page is for the new users who do not yet have an account. These Users are required to enter a token Number. Once registered, the users can detect abusive words by clicking on the generate report button. The users also have an option to view and download the generated report.

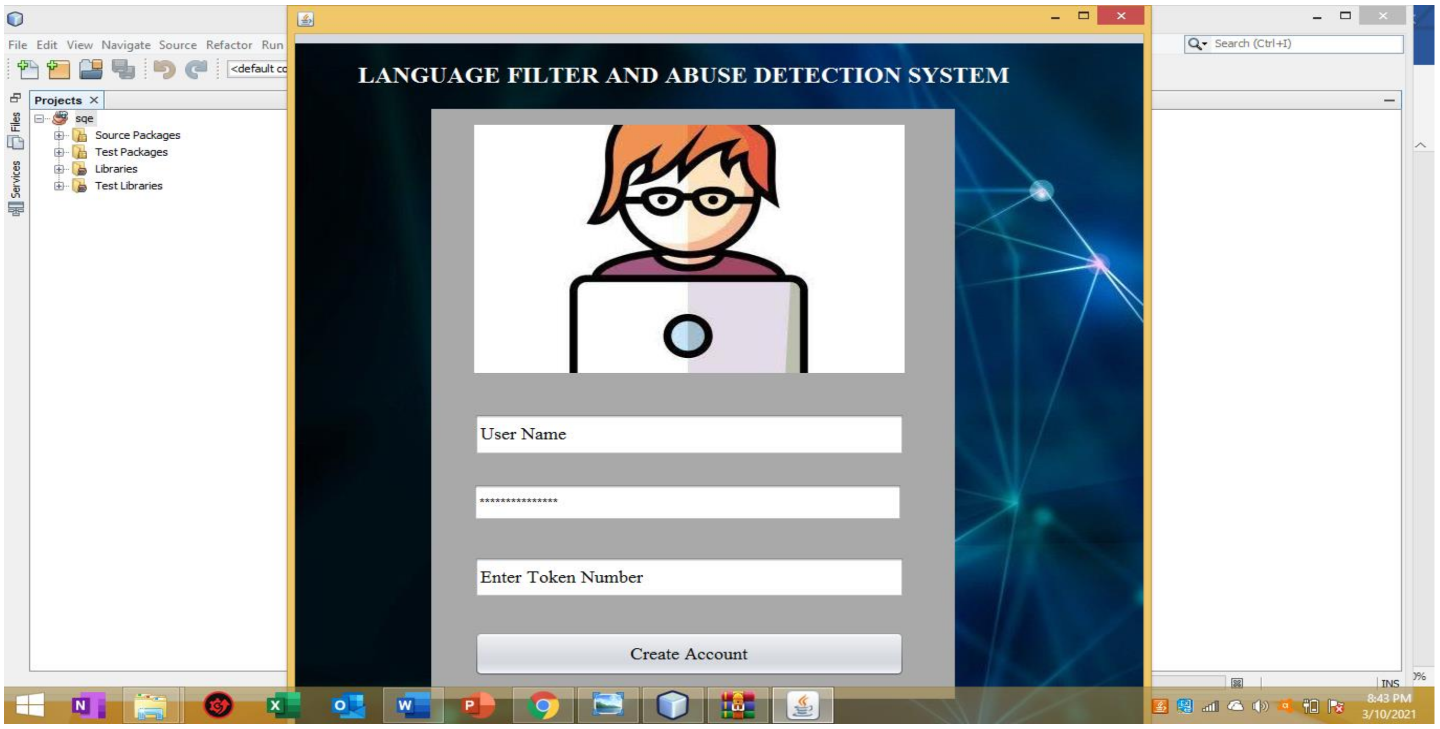
6.2 Screen Images

- **Welcome / Sign in**

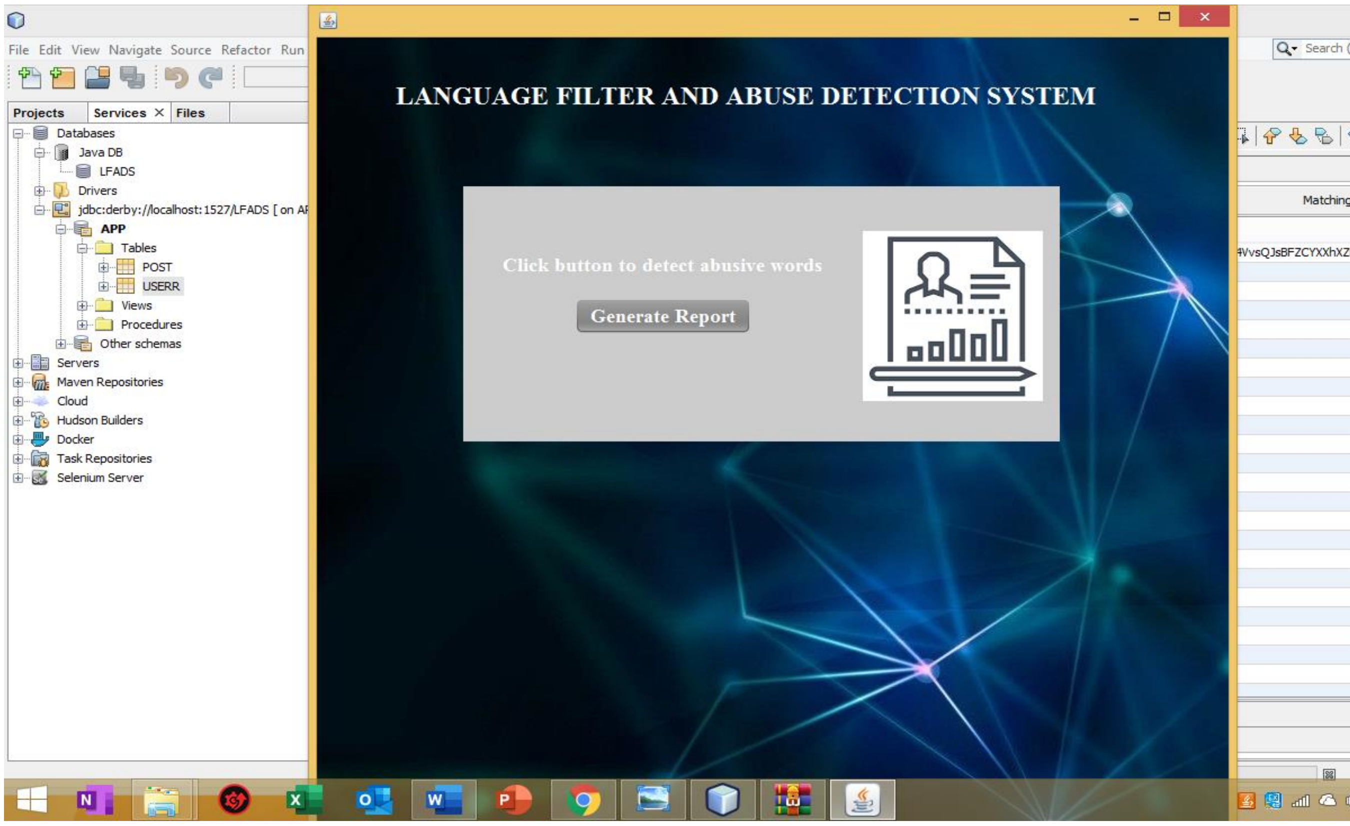

LANGUAGE FILTER AND ABUSE DETECTION SYSTEM
User Name
****
Don't have an account yet?
Sign in
Cancel
File Edit View Navigate Source Refactor Run
Projects
sqe
Source Packages
Test Packages
Libraries
Test Libraries

-

- **Sign up / Create Account**

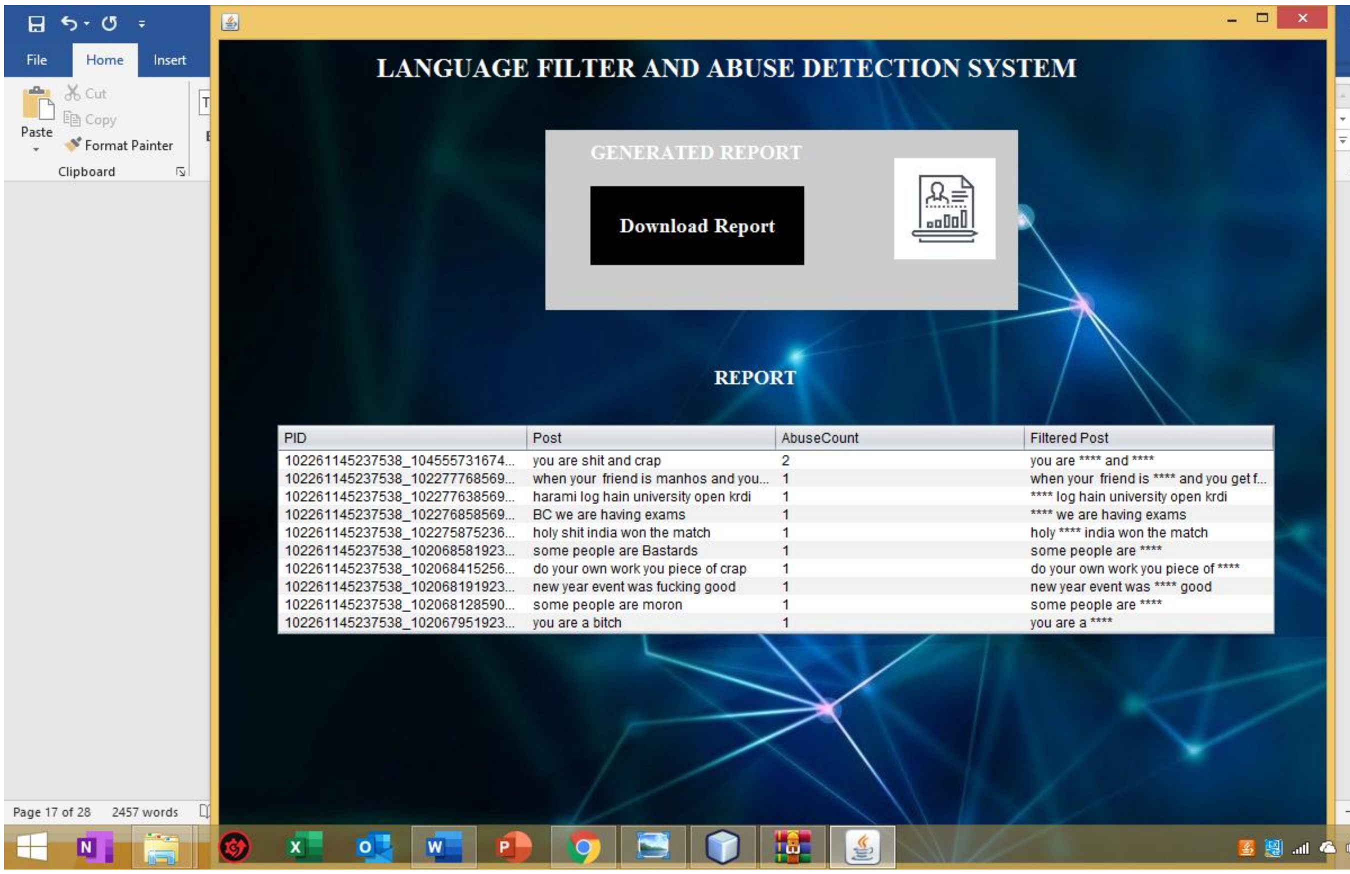


- **Generate Report**


LANGUAGE FILTER AND ABUSE DETECTION SYSTEM
Click button to detect abusive words
Generate Report

-

- **Download / View Report**



PID	Post	AbuseCount	Filtered Post
102261145237538_104555731674...	you are shit and crap	2	you are **** and ****
102261145237538_102277768569...	when your friend is manhos and you...	1	when your friend is **** and you get f...
102261145237538_102277638569...	harami log hain university open krdi	1	**** log hain university open krdi
102261145237538_102276858569...	BC we are having exams	1	**** we are having exams
102261145237538_102275875236...	holy shit india won the match	1	holy **** india won the match
102261145237538_102068581923...	some people are Bastards	1	some people are ****
102261145237538_102068415256...	do your own work you piece of crap	1	do your own work you piece of ****
102261145237538_102068191923...	new year event was fucking good	1	new year event was **** good
102261145237538_102068128590...	some people are moron	1	some people are ****
102261145237538_102067951923...	you are a bitch	1	you are a ****

-

6.3 Screen Objects and Actions

In the **"Welcome**" page, the user provides login username and password in the appropriate text boxes and confirms this operation by clicking the sign in button. If the user doesn't have an account, the user can click on the "**Don't have an account"** link. This opens up the "**create account page"** where the user is required to enter the user name, password and the token in the appropriate text boxes. Once logged in the user can click on the **"Generate Report Button"**. This will result in the filtering of the abusive content from the provided social media account and display the result. Once done, the user can click on the button of download report to download the generated report.

7 Requirements Matrix

Label	**Requirement**	**Component**
RL.01	The user must be able to create an account	(User Interface) Sign up page
RL.02	The user must be able to log into his/her account.	(User Interface) Sign in page
RL.03	The token provided by the user must link the system to the Facebook account.	Controller
RL.04	The posts must be filtered out on the basis of its content	Controller and Database
RL.05	The system must be able to generate the report containing the filtered posts, the abusive word count and the post content.	Controller
RL.06	The user must be able to view the generated report.	User Interface
RL.07	The user must be able to download the generated report.	User Interface
RL.08	The software should be able to run on any platform that supports java	Nil
RL.09	The software should be delivered with proper manual and documentation.	Nil
RL.10	There shall be relevant icons of buttons and function of buttons to facilitate the users to understand.	User Interface
RL.11	All devices with Windows 8+, 7, XP, MAC, Linux will be able to run this software.	Nil
RL.12	The system will use the web browser for fetching data from Facebook.	Nil
RL.13	The system must be able to run with all types of web browsers including safari, chrome, Firefox etc.	Nil

RL.14	The system will be able to support 5 simultaneous users	Nil
RL.15	The mean time to download the report must not take more than 1 sec.	Nil
RL.16	The user will be signed Up/Signed In within 0.5 seconds	Nil
RL.17	Every user will have access to only his/her own information/data	Nil
RL.18	No sensitive information of a user including its account's token will be breached by any other user.	Nil
RL.19	The system must be available at all times.	Nil
RL.20	The software should be delivered with complete source code.	Nil