

AUTHOR'S DECLARATION

All thanks to Almighty Allah, extremely powerful, the most gracious, compassionate and beneficent and His prophet **Hazrat Muhammad (Peace Be upon Him)** who is always being a true source of guidance for whole humanity. Again and Again thanks to **Almighty Allah**.

Thanks to my loving family, especially to my **parents** who taught me about not only the moral and spiritual living but also the style of living in the world. Especially thanks to my **Father** helped me financially and giving a moral support, encouraged me in every despair moment and enabled me to face the challenges of this project. .

My acknowledgment is due to many people for their help in the completion of this project. Especially, whose commitment and guiding method really helped me to get up whenever I fall during the project. I owe a lot to him for giving me time out of his very precious time and busy schedule. I am also thankful to all of teachers of my department who helped me in the completion of this project.

I would like to extend my earnest and whole hearted acknowledgments to **Institute of Southern Punjab Multan** for providing us with a center that radiates excellence and fosters good behavior.

Nouman Rasool
(BSIT-023R15-25)

FINAL YEAR PROJECT UNDERTAKING FORM

(Institute of Southern Punjab)

I hereby affirm that the originality and authenticity of the Final Year Project to be undertaken will be upheld. The report and the system that I submit at the conclusion of the Final Year Project will be the result of my own investigations and effort.

I understand that cheating and plagiarism constitute a serious violation of the university regulations, which will not only result in a failing grade for the Final Year Project but subject me to further disciplinary actions.

Signature of Student

Name: **Nouman Rasool**

Date: 05-15, 2020

Student Name	Nouman Rasool
Registration No.	BSIT-023R15-25
Contact No.	03040708363
Email Address	Nomiimalix197@gmail.com
Discipline	Computer Sciences
Year of Study	2015-2019
Area of Study	Python
Proposed Project Title	Tweeter Sentiment Analysis

Signature of Supervisor:

Name: Sir Kanwar Kaleem

Date: 5th May 2020

COPYRIGHTS ©

All copyrights of software application and report strictly reserved By ISP Multan.

Publisher:

ISP Bosan Road Multan

Developing and composing:

Nouman Rasool,
BSIT-023R15-25,
ISP Multan

Proofreading:

Mr. Kanwal Kaleem,
Mr. Syed Zohair Naqvi,
Ms. Manal Ahmed,
Ms. Mulika Fatima,

DEDICATION

I am dedicating To My Loving Parents, Friends and Respected Teachers, to my Supervisor Sir Kanwar Kaleem, The Institute of Southern Punjab my second magnificent home, all the people in my life who touch my heart.

ACKNOWLEDGMENT

All thanks to Almighty Allah, extremely powerful, the most gracious, compassionate and beneficent and His prophet **Hazrat Muhammad (Peace Be upon Him)** who is always being a true source of guidance for whole humanity. Again and Again thanks to **Almighty Allah**.

Thanks to my loving family, especially to my **parents** who taught me about not only the moral and spiritual living but also the style of living in the world. Especially thanks to my **Father** helped me financially and giving a moral support, encouraged me in every despair moment and enabled me to face the challenges of this project. .

My acknowledgment is due to many people for their help in the completion of this project. Especially, whose commitment and guiding method really helped me to get up whenever I fall during the project. I owe a lot to him for giving me time out of his very precious time and busy schedule. I am also thankful to all of teachers of my department who helped me in the completion of this project.

I would like to extend my earnest and whole hearted acknowledgments to **Institute of Southern Punjab Multan** for providing us with a center that radiates excellence and fosters good behavior.

Nouman Rasool
(BSIT-023R15-25)

ABSTRACT

This project addresses the problem of sentiment analysis in twitter; that is classifying tweets according to the sentiment expressed in them: positive, negative or neutral. Twitter is an online micro-blogging and social-networking platform which allows users to write short status updates of maximum length 140 characters. It is a rapidly expanding service with over 200 million registered users out of which 100 million are active users and half of them log on twitter daily - generating nearly 250 million tweets per day. Due to this large amount of usage we hope to achieve a reflection of public sentiment by analyzing the sentiments expressed in the tweets. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.

PROJECT BRIEF

Project title	Twitter Sentiment Analysis
Objective	The purpose of the project is to build a Desktop Application to check the positive or negative tweets about a person, company or product. And tell them that how people are reacts on them
Submitted by	Nouman Rasool
Submitted to	Mr. Kanwar Kaleem
Source language	Python, SQLite3
Database	SQLite3
Operating system	WINDOW 10
Tools used	Jupyter, VSCODE, Python Shell, Sublime text 3 DB browser Sqlite 3 and MS office.

Certificate

It is certifying that Nouman Rasool BSIT-023R15-25 Session has worked on and completed his software project “Twitter Sentiment Analysis” at department of computer science. Institute of southern Punjab Multan (ISP) in partial fulfilment of the requirement for the degree of BSIT in computer science under my guidance and supervision.

In our opinion, it is satisfactory and up to the mark and therefore fulfil the requirement of BSIT computer science degree.

Internal examiner

Department of computer science

Institute of southern Punjab Multan campus

(Signature)

External examiner

(Signature)

Project coordinator

Institute of southern Punjab

(Signature)

H.O.D

Institute of southern Punjab

(Signature)

Table of Contents

CHAPTER 1:	INTRODUCTION	15
1.1	Introduction	15
1.1.1	Sentiment Meaning?	15
1.1.2	Sentiment Analysis	15
1.1.3	What is Twitter?.....	15
1.1.4	Collect Tweet	16
1.1.5	Analysis the Tweet.....	16
1.1.6	Tweet Stats.....	16
1.2	Aim of project	16
1.3	Scope	17
1.4	SDLC MODEL	17
1.4.1	BASIC STAGES OF SOFTWARE DEVELOPMENT LIFE CYCLE.....	17
1.5	Types of SDLC Model	18
1.5.1	Waterfall Model.....	19
1.5.2	V-Shaped Model	20
1.5.3	Spiral Model (SDM)	21
1.5.4	Agile Model	24
1.6	Summary of Outcomes.....	27
CHAPTER 2:	LITERATUREREVIEW&ANALYSIS	29
2.1	Literature Review	29
2.2	Objectives.....	29
2.3	Problem Statement	30
2.4	Related Work.....	30
2.4.1	Umati Project to Monitor Hate Speech on Social Media:.....	30
2.4.2	Rule Based Approach to Detecting Hate Speech.....	30
2.5	Proposed system.....	31
2.6	Comparison table.....	31
CHAPTER 3:	DESIGN.....	33
3.1	Introduction	33
3.2	Data Flow Diagram (DFD):	33
3.2.1	Physical DFD	33
3.3	Use case diagram.....	34

3.3.1	Use Case Diagram:	34
3.4	Sequence Diagrams	35
3.5	Class Diagram:	36
3.6	Entity Relationship Diagram:	36
3.7	Database	36
3.7.1	Database Schema:	37
2-	Primary Key	38
3-	Foreign key	38
4-	Composite Key	38
3.8	Normalization	39
3.8.1	Types of Normalization	39
CHAPTER 4:	Functional Requirement	41
4.1	Functional requirement:	41
4.2	Non-Functional Requirements:	41
CHAPTER 5	IMPLEMENTATION	44
5.1	User Manual Create New Account	44
5.2	Testing of the system	48
5.3	Test Cases	49
Test case title:	User Login	50
Appendix	51
References:	52

List of Figures

Figure 1.1 V-Shaped Model.....	20
Figure 1.2 Spiral Model	22
Figure 1.3 Iterative and Incremental Model	23
Figure 1.4 Agile Models	24
Figure 1.5 RAD Model	26
Figure 3.1 Physical DFD.....	33
Figure 3.2 Use Case Diagram	34
Figure 3.3 User Case Diagram.....	35
Figure 3.4 Sequence Diagram.....	35
Figure 3.5 Class Diagram	36
Figure 3.6 Entity Relationship Diagram	36
Figure 5.1	44
Figure 5.2.....	44
Figure 5.3	45
Figure 5.4	45
Figure 5.5	46
Figure 5.5	46
Figure 5.6.....	47

List of Tables

Table 2.1 Comparison Table.....	31
Table 6.1 Testing	49

CHAPTER NO 1
INTRODUCTION

CHAPTER 1: INTRODUCTION

1.1 Introduction

1.1.1 Sentiment Meaning?

The meaning of sentiment is thought, attitude or judgement by feeling sentiment analysis, also know opinion mining, study of certain entities of people's sentiment. Sentiment information is get by internet. People are post their information or content by different social media like as online social media, forums and micro-blogging. By researcher's, API (application programming interface) are provided by many social media websites. Twitter has three types of API which are available. Which are Rest API, Search API and Streaming API. Rest API means the developer collect the status data and user information. The search API allow developer to query specific twitter content. Streaming API able to collect Twitter content is real time. The developer can mix these APIs to create own application. .

1.1.2 Sentiment Analysis

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker

1.1.3 What is Twitter?

In 2006, Jack Dorsey, co-founder of Twitter, had an idea -- he would create a SMS-based communications platform in which friends could keep tabs on each other by updating statuses. In the beginning, Twitter was an idea all-too-similar to texting. Twitter is a micro blogging media in real time to express the persuasion of a person or group about a particular topic to appear going on a timeline. The message which is displayed on Twitter is named as Tweet. The users are made by friends and followings, tweets and their timeline are key components of Twitter.

I have designed a desktop application which performed Hate Speech sentiment analysis on Twitter which get tweets that matched particular keywords provided by the user. For example, if a user is interested in performing sentiment analysis on tweets which contain the word "Kashmir" he / she will enter that keyword and the desktop application will perform the appropriate sentiment analysis and display the results in the form of graph like negative, week negative and strong negative. This desktop application can be easily used by user and it provides a layer of abstraction to the developer from the low level operations so it is easier to learn. I implemented our algorithm in python and integrated it with GUI using TKINTER. For acquiring

tweets from Twitter I used the API in this system. If I used the Twitter Streaming API and the user specified a keyword which is not very common in Twitter, my application may have to wait for a long time to acquire enough tweets to display reasonable results. There are three ways of performing sentiment analysis in my application:

1. Collect Tweets
2. Analysis the tweets
3. Tweet Stats

1.1.4 Collect Tweet

This feature calculates the popularity score of the keyword which is a number from 100 to -100. The more positive popularity score suggests that the keyword is highly positively popular on Twitter, while the more negative popularity score suggests that the keyword is highly negatively popular on Twitter.

1.1.5 Analysis the Tweet

This feature compares the popularity score of two or three different keywords and replies with which keyword is currently most popular on Twitter. This can have many interesting applications for example having our web application

1.1.6 Tweet Stats

This feature is for long term sentiment analysis. I input a number of popular keywords on Twitter on which a backend operation runs after every hour, calculates the popularity score for the tweets generated on that keyword within an hour time frame and stores the results against every hour.

1.2 Aim of project

The objective of this project is to implement an algorithm for automatic classification negative tweets. This system is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream like negative. Moreover, the response on twitter is more prompt and also more general (since the number of users who tweet is substantially more than those who write web blogs on a daily basis). This system could be done by analyzing overall public sentiment towards that firm with respect to time.

1.3 Scope

This project will be helpful for the political parties, companies as well as to the common people. It will also be helpful to the political party for reviewing about the program that they are going to do or program that they have performed. Similarly, companies also can get review about their new product on newly released hardware or software. Also the movie maker can take review on currently running movie. By analyzing the tweets analyzer can get result on how negative people about it. Below are the module details:

- Maintain Login Module
- Calculate all tweets scores.
- Compare all tweets
- Compare tweets statistically viewed by users.

1.4 SDLC MODEL

The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. SDLC is a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. The software development life cycle is also known as the software development process.

1.4.1 BASIC STAGES OF SOFTWARE DEVELOPMENT LIFE CYCLE

Stage 1. Planning and requirement analysis

Each software development life cycle model starts with the analysis, in which the stakeholders of the process discuss the requirements for the final product. The goal of this stage is the detailed definition of the system requirements. Besides, it is needed to make sure that all the process participants have clearly understood the tasks and how every requirement is going to be implemented. Often, the discussion involves the QA specialists who can interfere the process with additions even during the development stage if it is necessary.

Stage 2. Designing project architecture

At the second phase of the software development life cycle, the developers are actually designing the architecture. All the different technical questions that may appear on this stage are discussed by all the stakeholders, including the customer. Also, here are defined the technologies used in

the project, team load, limitations, time frames, and budget. The most appropriate project decisions are made according to the defined requirements.

Stage 3. Development and programming

After the requirements approved, the process goes to the next stage – actual development. Programmers start here with the source code writing while keeping in mind previously defined requirements. The system administrators adjust the software environment, front-end programmers develop the user interface of the program and the logics for its interaction with the server.

The programming by itself assumes four stages

- Algorithm development
- Source code writing
- Compilation
- Testing and debugging

Stage 4. Testing

The testing phase includes the debugging process. All the code flaws missed during the development are detected here, documented, and passed back to the developers to fix. The testing process repeats until all the critical issues are removed and software workflow is stable.

Stage 5. Deployment

When the program is finalized and has no critical issues – it is time to launch it for the end users. After the new program version release, the tech support team joins. This department provides user feedback; consult and support users during the time of exploitation. Moreover, the update of selected components is included in this phase, to make sure, that the software is up-to-date and is invulnerable to a security breach.

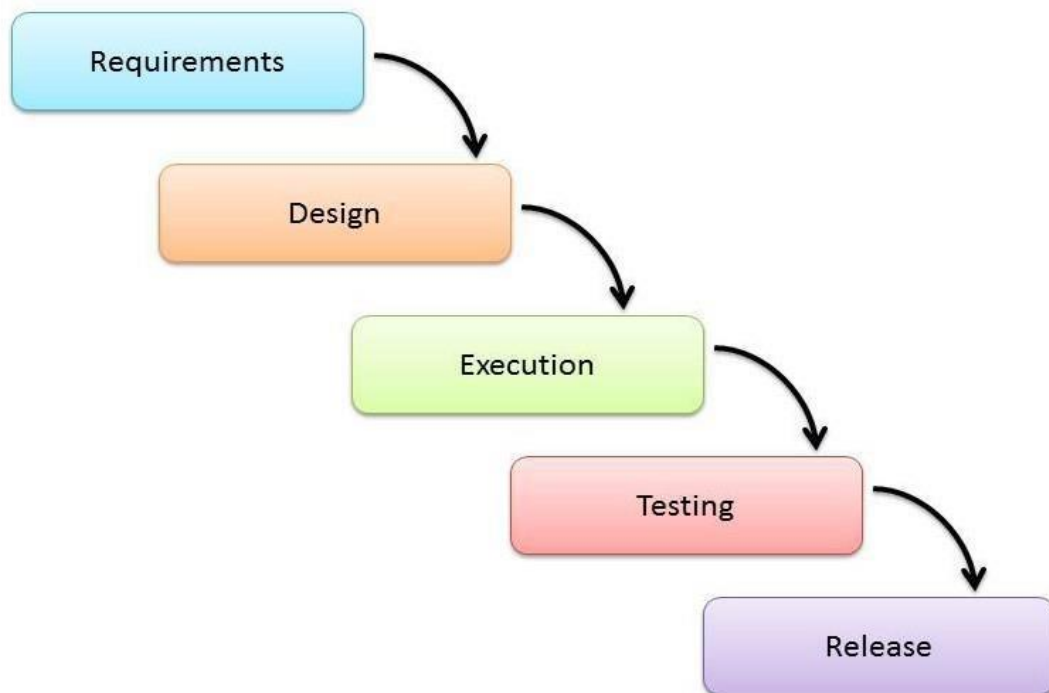
1.5 Types of SDLC Model

- Waterfall mode
- Spiral model
- V-shaped model
- Agile model
- RAD Model

1.5.1 Waterfall Model

Description

The Waterfall Model is a linear sequential flow. In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. The waterfall approach is the earliest approach and most widely known that was used for software development.



The usage

Projects which not focus on changing the requirements, for example, projects initiated from a request for proposals, the customer has a very clear documented requirements

Advantages and Disadvantages

Advantages	Disadvantages
<ul style="list-style-type: none">• Easy to explain to the users.• Structures approach.• Stages and activities are well defined.• Helps to plan and schedule the project.• Verification at each stage ensures early detection of errors/misunderstanding.• Each phase has specific deliverables.	<ul style="list-style-type: none">• Assumes that the requirements of a system can be frozen.• Very difficult to go back to any stage after it finished.• A little flexibility and adjusting scope is difficult and expensive.• Costly and required more time, in addition to the detailed plan.

1.5.2 V-Shaped Model

Description

It is an extension of the waterfall model, instead of moving down in a linear way, the process steps are bent upwards after the implementation and coding phase, to form the typical V shape. The major difference between the V-shaped model and waterfall model is the early test planning in the V-shaped model.

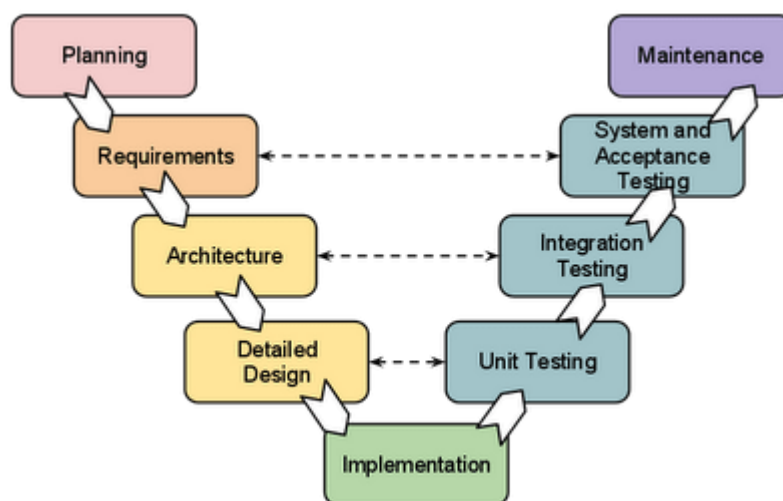


Figure 1.1 V-Shaped Model

The usage

- Software requirements clearly defined and known
- Software development technologies and tools are well-known

Advantages and Disadvantages

Advantages	Disadvantages
<ul style="list-style-type: none">• Simple and easy to use• Each phase has specific deliverables.• Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle.• Works well for where requirements are easily understood.• Verification and validation of the product in the early stages of product development.	<ul style="list-style-type: none">• Very inflexible, like the waterfall model.• Adjusting scope is difficult and expensive.• The software is developed during the implementation phase, so no early prototypes of the software are produced.• The model doesn't provide a clear path for problems found during testing phases.• Costly and required more time, in addition to a detailed plan

1.5.3 Spiral Model (SDM)

Description

It is combining elements of both design and prototyping-in-stages, to combine advantages of top-down and bottom-up concepts. This model of development combines the features of the prototyping model and the waterfall model. The spiral model is favored for large, expensive, and complicated projects. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.

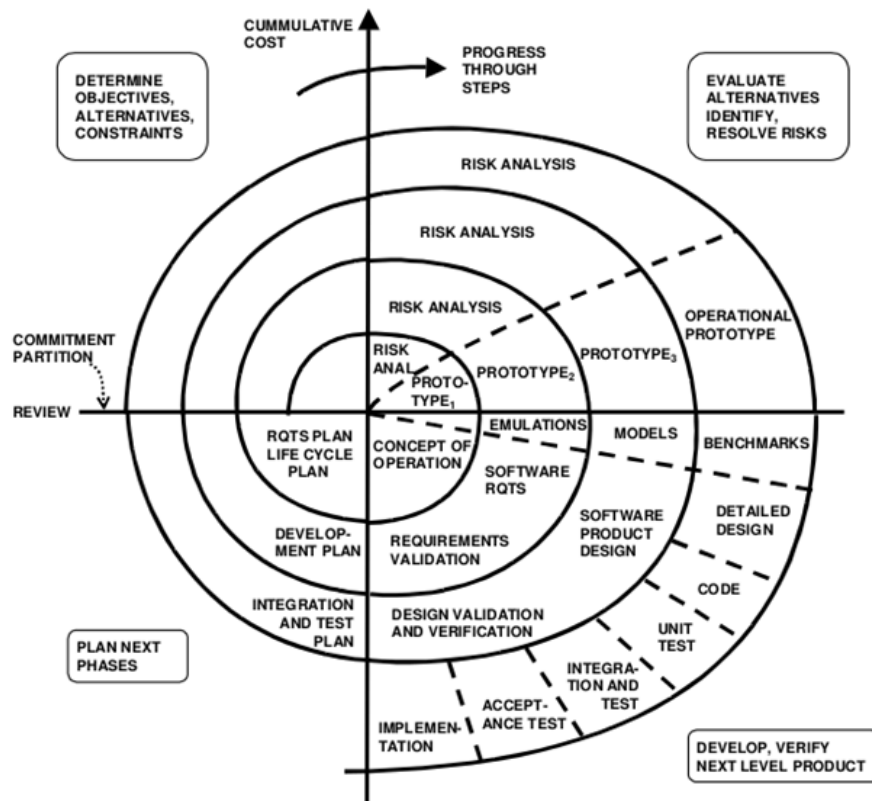


Figure 1.2 Spiral Model

The usage

It is used in the large applications and systems which built-in small phases or segments.

Advantages and Disadvantages:

Advantages	Disadvantages
<ul style="list-style-type: none"> Estimates (i.e. budget, schedule, etc.) become more realistic as work progressed because important issues are discovered earlier. Early involvement of developers. Manages risks and develops the system into phases. 	<ul style="list-style-type: none"> High cost and time to reach the final product. Needs special skills to evaluate the risks and assumptions. Highly customized limiting re-usability

Iterative and Incremental Model

Description

It is developed to overcome the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during the development of earlier parts or versions of the system. It can consist of mini waterfalls or mini V-Shaped model

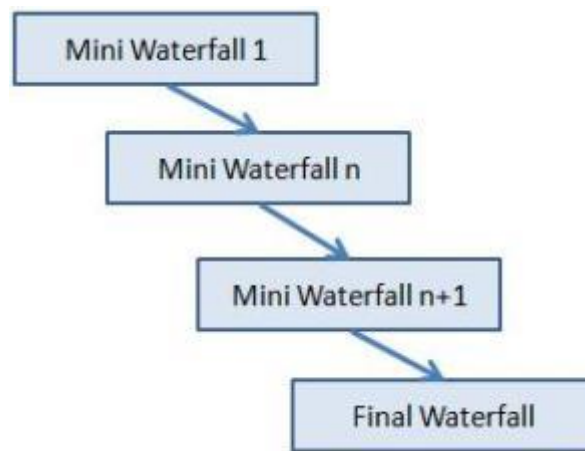
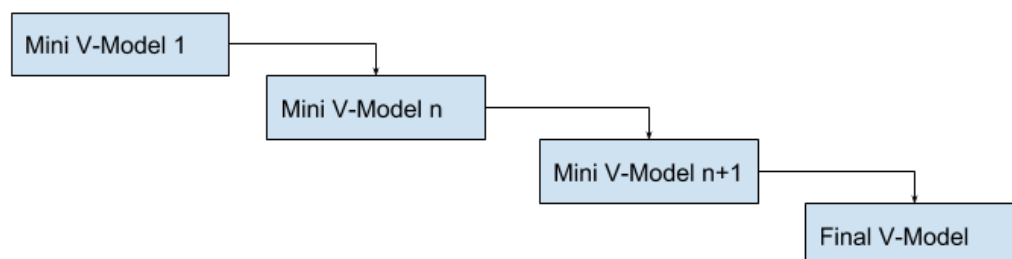


Figure 1.3 Iterative and Incremental Model

The usage

It is used in shrink-wrap application and large system which built-in small phases or segments. Also, can be used in a system has separated components, for example, ERP system. Which we can start with the budget module as a first iteration and then we can start with the inventory module and so forth.



Advantages

- Produces business value early in the development lifecycle.
- Better use of scarce resources through proper increment definition.
- Can accommodate some change requests between increments.
- More focused on customer value than the linear approaches.
- We can detect project issues and changes earlier.

Disadvantages

- Requires heavy documentation.
- Follows a defined set of processes.
- Defines increments based on function and feature dependencies.
- Requires more customer involvement than the linear approaches.
- Partitioning the functions and features might be problematic.
- Integration between the iterations can be an issue if it is not considered during the development and project planning.

1.5.4 Agile Model

Description

It is based on iterative and incremental development, where requirements and solutions evolve through collaboration between cross-functional teams.

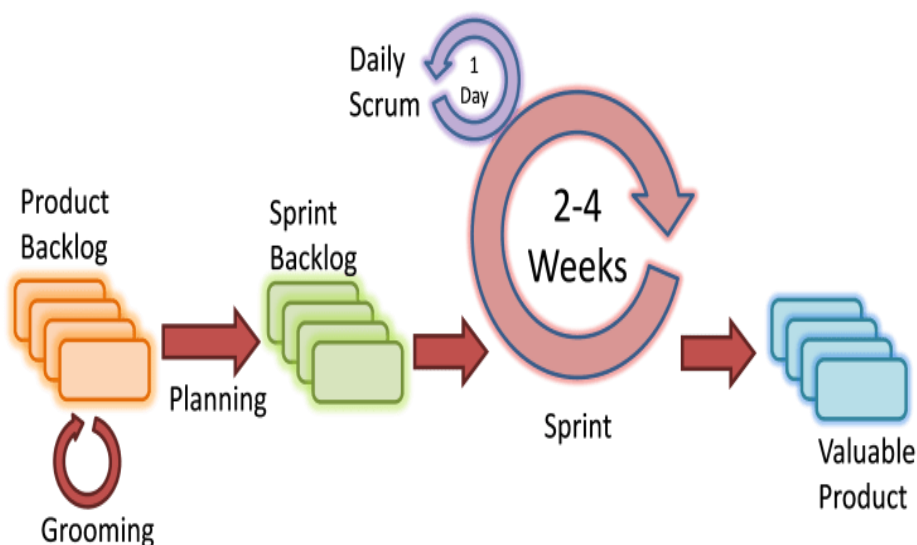


Figure 1.4 Agile Models

Agile Model

The usage

It can be used with any type of the project, but it needs more engagement from the customer and to be interactive. Also, we can use it when the customer needs to have some functional requirement ready in less than three weeks and the requirements are not clear enough. This will enable more valuable and workable piece for software early which also increase the customer satisfaction.

Advantages and Disadvantages

Advantages	Disadvantages
<ul style="list-style-type: none">• Decrease the time required to avail some system features.• Face to face communication and continuous inputs from customer representative leaves no space for guesswork.• The end result is the high-quality software in the least possible time duration and satisfied customer.	<ul style="list-style-type: none">• Scalability.• The ability and collaboration of the customer to express user needs.• Documentation is done at later stages.• Reduce the usability of components.• Needs special skills for the team.

I used Rapid Application development (RAD) model to develop my application.

Which model I have choose and why?

My system is based on **Rapid application development (RAD) model** because I have all information about my project and no need of going backward during Process.

Rapid Application Development Model

The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

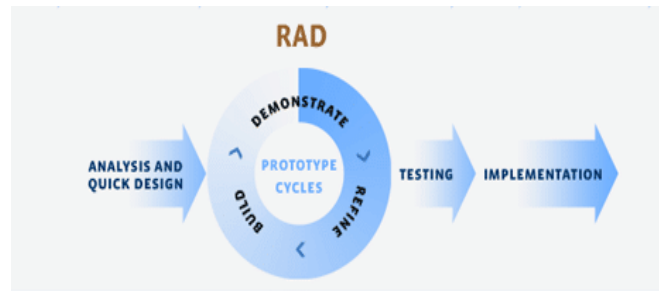


Figure 1.5 RAD Model

Rapid Application Development model has four different phases, which are following.

- 1) Requirement Planning.
- 2) User Description.
- 3) Construction.
- 4) Cut over

User Requirements

It involves the use of various techniques used in requirements elicitation like brainstorming, task analysis, form analysis, user scenarios, FAST (Facilitated Application Development Technique), etc. It also consists of the entire structured plan describing the critical data, methods to obtain it and then processing it to form final refined model.

User Description

This phase consists of taking user feedback and building the prototype using developer tools. In other words, it includes re-examination and validation of the data collected in the first phase. The dataset attributes are also identified and elucidated in this phase.

Construction

In this phase, refinement of the prototype and delivery takes place. It includes the actual use of powerful automated tools to transform process and data models into the final working product. All the required modifications and enhancements are too done in this phase.

Cutover

All the interfaces between the independent modules developed by separate teams have to be tested properly. The use of powerfully automated tools and subparts makes testing easier. This is followed by acceptance testing by the user.

Application of Rapid Development Model?

- Use of reusable components helps to reduce the cycle time of the project.
- Feedback from the customer is available at initial stages.
- Reduced costs as fewer developers are required.
- Use of powerful development tools results in better quality products in comparatively shorter time spans.
- The progress and development of the project can be measured through the various stages.
- It is easier to accommodate changing requirements due to the short iteration time spans.

1.6 Summary of Outcomes

Through this application I can find the results of different tweets sentiment in form of graph, like negative and it also provides a layer of abstraction to the developer from the low level operations so it is easier to learn.

CHAPTER NO 2
LITERATURE REVIEW & ANALYSIS

CHAPTER 2: LITERATUREREVIEW&ANALYSIS

2.1 Literature Review

Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification (Pang and Lee 2008) to learning the polarity of words and phrases (e.g., (Hatzivassiloglou and McKeown 1997; Esuli and Sebastiani 2006)). Given the character limitations on tweets, classifying the sentiment of Twitter messages is most similar to sentencelevel sentiment analysis (e.g., (Yu and Hatzivassiloglou 2003; Kim and Hovy 2004)); however, the informal and specialized language used in tweets, as well as the very nature of the microblogging domain make Twitter sentiment analysis a very different task. It's an open question how well the features and techniques used on more well-formed data will transfer to the microblogging domain.

Just in the past year there have been a number of papers looking at Twitter sentiment and buzz (Jansen et al. 2009; Pak and Paroubek 2010; O'Connor et al. 2010; Tumasjan et al. 2010; Bifet and Frank 2010; Barbosa and Feng 2010 ; Davidov, Tsur, and Rappoport 2010). Other researchers have begun to explore the use of part-of-speech features but results remain mixed. Features common to microblogging (e.g., emoticons) are also common, but there has been little investigation into the usefulness of existing sentiment resources developed on non-microblogging data.

Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data (Pak and Paroubek 2010; Bifet and Frank 2010). (Barbosa and Feng 2010) exploit existing Twitter sentiment sites for collecting training data. (Davidov, Tsur, and Rappoport 2010) also use hashtags for creating training data, but they limit their experiments to sentiment/non-sentiment classification, rather than 3-way polarity classification, as we do.

2.2 Objectives

The main objectives of my system is given below.

- To Compare tweets in efficient and compact manner
- To fast response of every activity
- To facilitate easy data entry
- To provide data security

2.3 Problem Statement

Sentiment analysis of in the domain of micro-blogging is a relatively new topic so there is still a lot of room for further system in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

2.4 Related Work

The process of understanding the system includes becoming involved, sometimes deeply involved in the system, as it currently exists. During this process the system designer learns a great deal about the subject matter. The system designer must know what it is going on and what he should do.

2.4.1 Umati Project to Monitor Hate Speech on Social Media:

Umati is a hate speech monitoring project that analyses incidents multilingual hate speech in the Kenyan online space such as blogs, forums, online newspapers, Facebook and Twitter (iHub Research, 2013). The first phase of Umati involves the use of human monitors to collect and analyze hate speech from the various online platforms. The human monitors scour the platforms for incidents of hate speech. Once a human monitor encounters a statement that is considered to be hate speech, they enter it into an online form, whilst providing additional information about the statement in the form. Finally the statement is sorted into one of three categories: offensive speech, moderately dangerous speech and extremely dangerous speech (iHub Research, 2013). This hate speech detection process is illustrated in Figure 2.4 (iHub Research, 2013). The current human processing of hate speech text by Umati phase I is time consuming, involves a lot of effort and human input. A more automated way of detecting hate speech in text would be preferable. The next phase, Umati II will involve the application of machine learning and natural language processing techniques to automatically identify instances of hate speech. This phase is however on going and not much has been achieved from it.

2.4.2 Rule Based Approach to Detecting Hate Speech

Regular expressions are an algebraic notation for specifying search strings (Daniel &

James, 2000). Regular expressions can be used in classification rule builders to match a wide variety of patterns and consequent use of the matches to set classification labels. In his work, Maloba (2013) proposes the use of regular expressions to detect hate speech in multi lingual (English, Swahili and Sheng') text. In the work he notes that hate speech in Kenya is mostly dependent on ethnic grouping and uses this as the basis for formulating rules based on well-known ethnic stereotypes to match and identify hate speech.

2.5 Proposed system

The system will calculate the statistics of sentiments in post of tweets. It's easy to measure the thing which has some physical existence. But measuring the sentiments

2.6 Comparison table

Features	Accuracy	Positive	Negative	Both	Objective
Word token	61.7	61.1	73.1	14.6	37.7
Word + prior	63.0	61.2	75.5	14.6	40.7
10 features	65.7	65.1	77.1	16.1	46.2

Table 2.1 Comparison Table

CHAPTER NO 3

DESIGN

CHAPTER 3: DESIGN

3.1 Introduction

The most important phase of any project is the designing of the logical data and database which involve the designing of UML diagram and different normalized tables and then the relationship between those normalized tables. For input designing, table designing is done. There are several steps involved in developing an effective database design. The steps are such as, what data element must be stored, who will access them, and how.

3.2 Data Flow Diagram (DFD):

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

3.2.1 Physical DFD

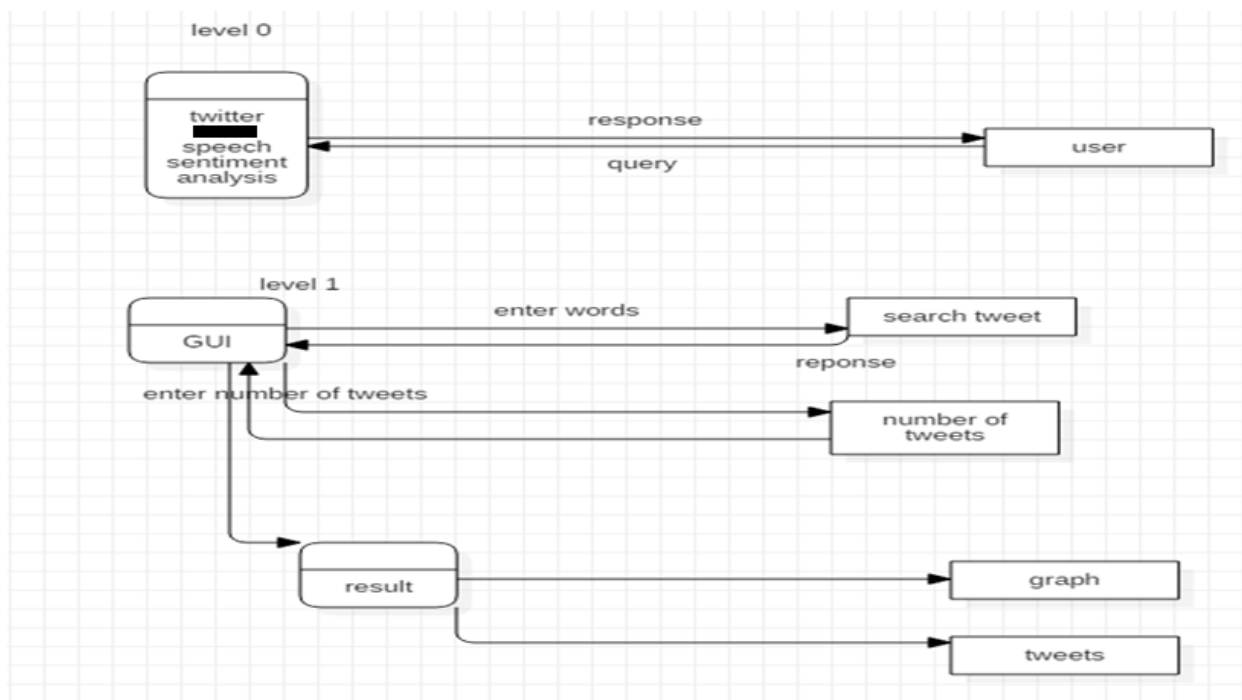


Figure 3.1 Physical DFD

3.3 Use case diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

It is often used in early stages design process to collect internal requirements of a project. The diagram shows the overall functionality of the system. The use cases diagram of this systems are,

3.3.1 Use Case Diagram:

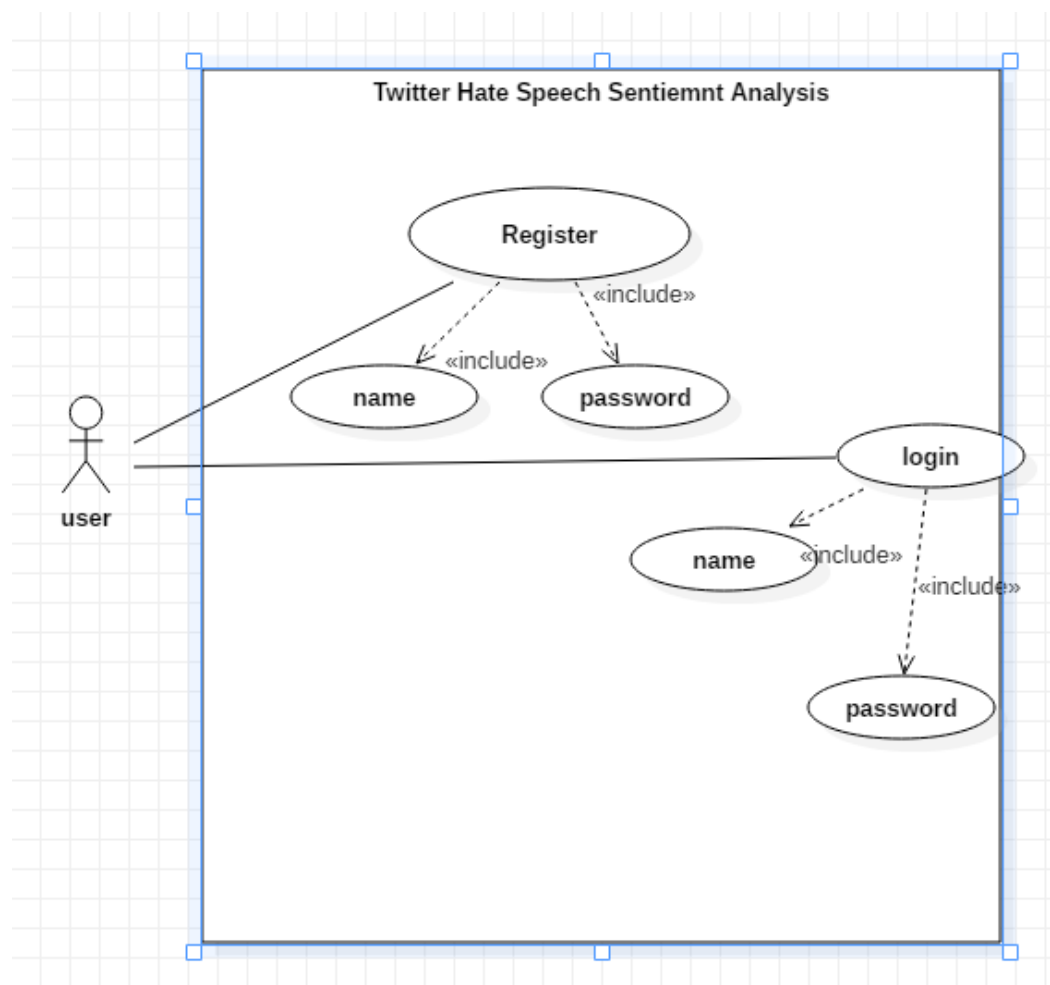


Figure 3.2 Use Case Diagram

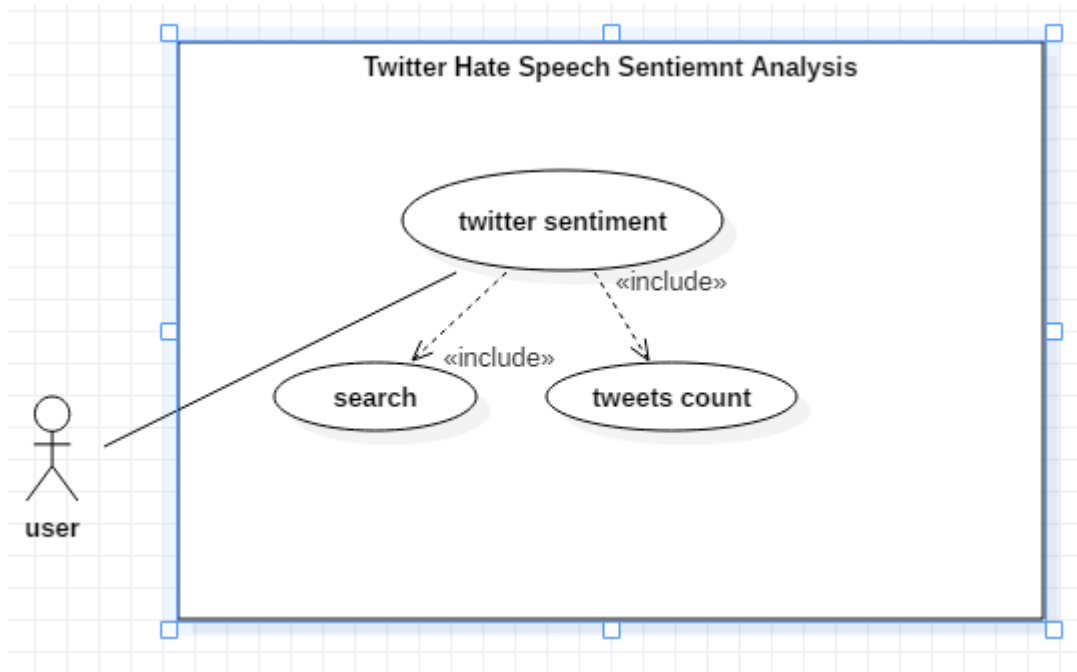


Figure 3.3 User Case Diagram

3.4 Sequence Diagrams

A sequence diagram, in the context of UML, represents object collaboration and is used to define event sequences between objects for a certain outcome.

Sequence Diagram for scanning files:

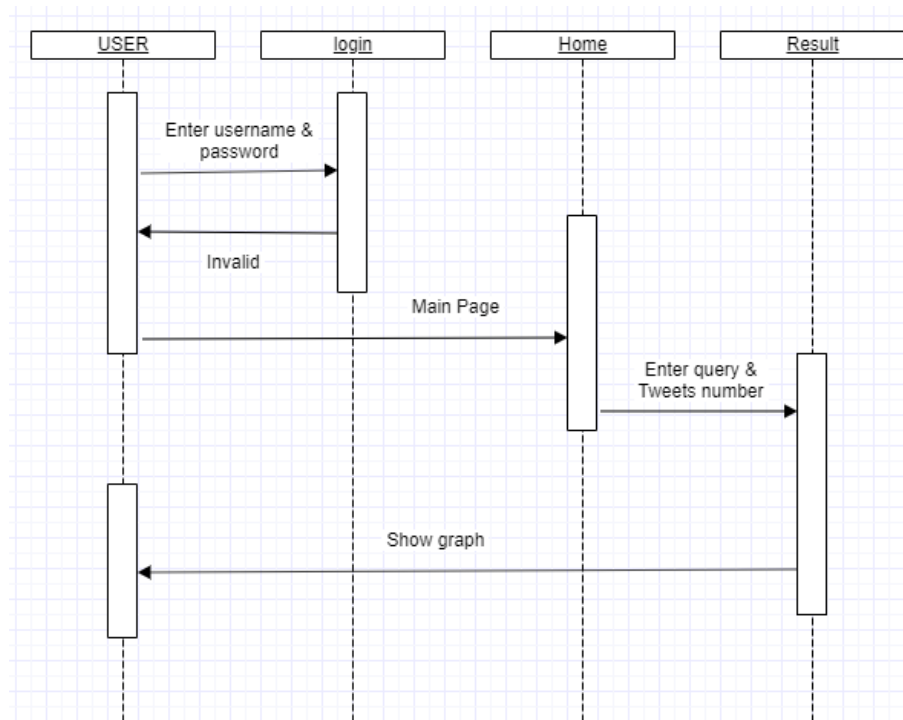


Figure 3.4 Sequence Diagram

3.5 Class Diagram:

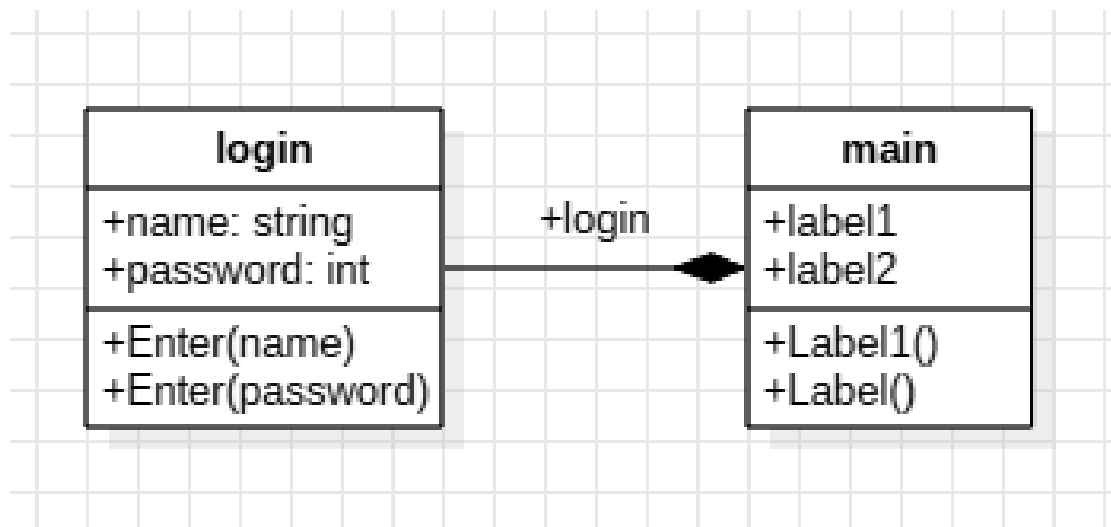


Figure 3.5 Class Diagram

3.6 Entity Relationship Diagram:

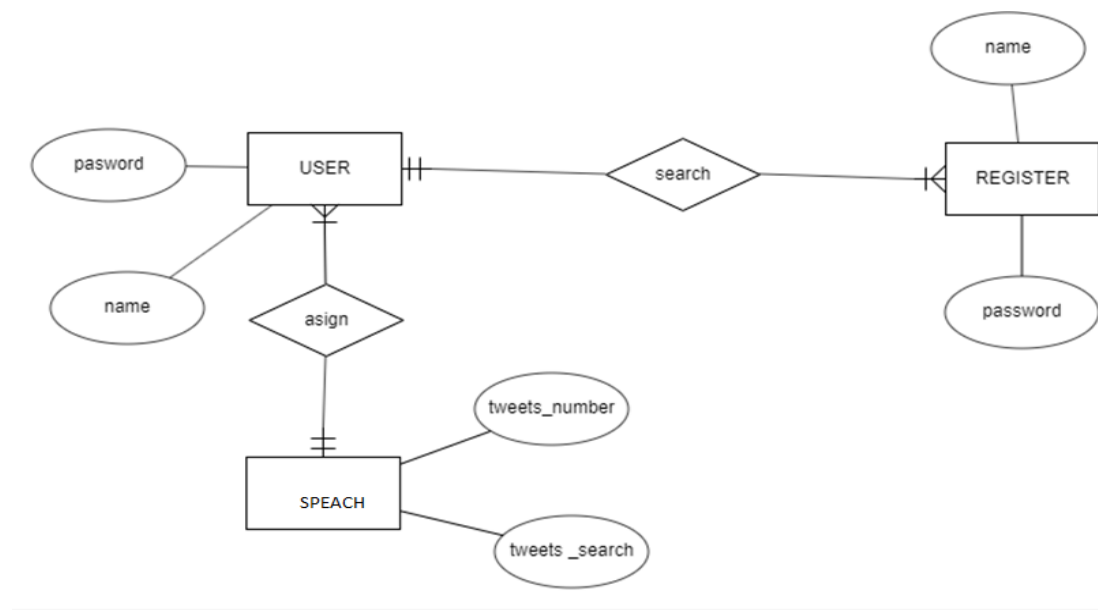


Figure 3.6 Entity Relationship Diagram

3.7 Database

A database is collection of data, which is organized in such a way that each piece of data is available to those who need it and with minimum duplication of data.

Another definition in terms of database may be as “A database is a collection of data organized so as to minimize redundancy and maximize access.”

Advantages of Database

The most common advantages of database may be as follows:

- Sharing of data
- Ease of application development
- Uniform security, privacy and integrity controls
- Reduced program maintenance
- Minimal data redundancy
- Consistency of data
- Data integrity

Actors:

There are only one actors that performing their specific role in the accomplishment of this system.

- ✓ User

3.7.1 Database Schema:

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

A database schema can be divided broadly into two categories –

Physical Database Schema:

This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

Logical Database Schema:

This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Constraints:

These are the conditions that obey database.

Entity

Any object or concept identified by an enterprise that exists independently and about which necessary to store data. It may be anything like a person, a place, an event, or concept or an object.

Attributes

These are the characteristics or properties of an entity that is of interest to the organization. For example, USER is an entity in my project, its attributes mostly these are given: Username, Password etc. the entities and also their attributes also given below in sequence.

Keys

A key is a distinct for each individual entity in an entity set. Key attributes are the attributes whose values are uniquely identified and do not exist again.

1-Super Key

A super key is an attribute or a set of attributes that uniquely identified an entity. For example, Login name is a super key because it can be used to identify each Admin uniquely.

2- Primary Key

The primary key is one or more column in a table used to uniquely identify each row to the table. Primary key values can't be null and must be unique across the column. A multicolumn primary key is called Composite Primary Key.

3- Foreign key

Foreign key represents relationships between the tables or relations. A foreign key is a column (or a group of columns) whose values are derived from the primary/unique key of the same or some other tables

4- Composite Key

If the key consists of more than one attribute for unique identification, then it will be called as composite key.

Before description of tables, it is important to understand normalization.

3.8 Normalization

Normalization means putting a relation in to a higher normal. The purpose of normalization is to produce a stable set of relations that is faithful model of operations of the enterprise.

3.8.1 Types of Normalization

Following are major types of Normalization

- First Normal Form
- Second Normal Form
- Third Normal Form

First Normal Form

A relation is in First Normal Form if and only every attribute is single valued for each tuple. First Normal Forms does not allow any repeating fields. An alternative way of describing first normal form is to say that the domains of the attributes of a relation are atomic.

Second Normal Form

A relation is in Second Normal Form if and only if it is 1NF (first normal form) and all the non-key attributes are fully functionally dependent on the key. In my project dost flour and general mill database table of product all attributes of this table dependent on key attributes.

Third Normal Form

A relation is in third normal form if and only if it is in second normal form and no non-key attribute is transitively dependent on the primary key.

Third Normal Form can be defined as “A relation is in third normal form, if it is in second normal form and remove all transitive dependencies

CHAPTER NO 4

DEVELOPMENT

CHAPTER 4: Functional Requirement

4.1 Functional requirement:

These requirements are organized by the features discussed in the vision document. Features from vision documents are then refined into use case diagrams and to sequence diagram to best capture the functional requirements of the system.

- System will maintain Login Module
- System will calculate all tweets scores.
- System will compare all tweets
- System will compare tweets statistically viewed by users

4.2 Non-Functional Requirements:

The Non-functional requirements are part of software system specifications which covers the non-functional requirements that the software system need to address.

Performance Requirements:

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

Safety Requirements:

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.

Security Requirements:

The any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

Software Quality Attributes:

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

Software Requirement:

- Windows
- Python Platform
- Modern Web Browser
- Twitter API

CHAPTER NO 5

IMPLEMENTATION

CHAPTER 5 IMPLEMENTATION

5.1 User Manual Create New Account

Subscribe To Our Youtube Channel.
[/www.youtube.com/channel/UCUGAq4ALoWW4PDU6CmlriSg?view_as=subscriber](https://www.youtube.com/channel/UCUGAq4ALoWW4PDU6CmlriSg?view_as=subscriber)

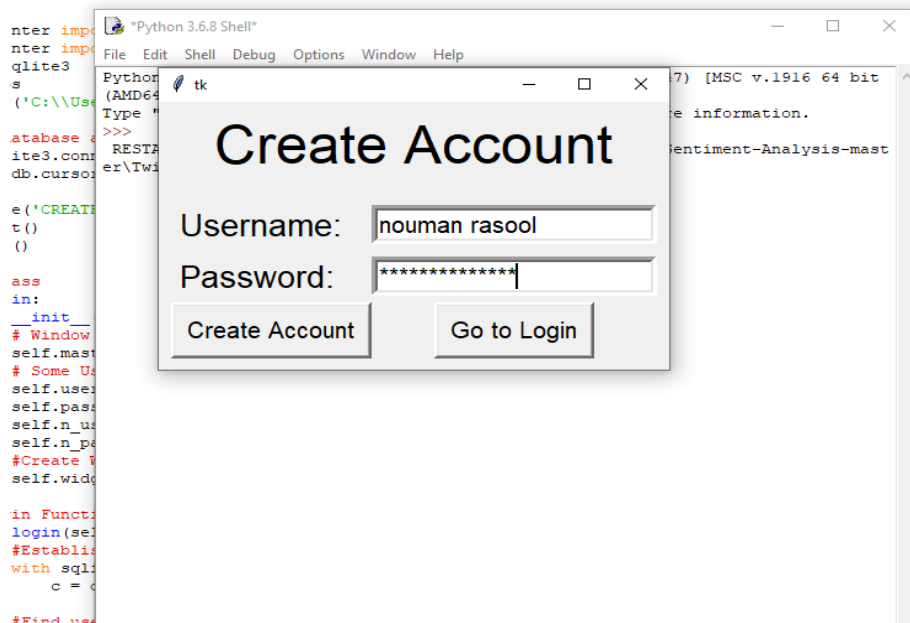


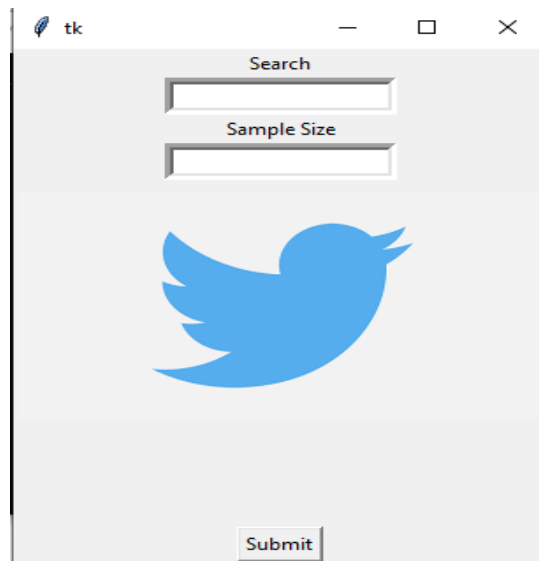
Figure 5.1

After Create Account Login the Account



Figure 5.2

Put Query and Number of tweets you want to search



The image shows a Tkinter window titled 'tk'. Inside the window, there are two text input fields. The first field is labeled 'Search' and the second is labeled 'Sample Size'. Both fields are currently empty. Below the input fields is a large blue Twitter bird logo. At the bottom of the window is a 'Submit' button.

Figure 5.3

Like as Pakistan is query and 80 is number for tweet search



The image shows the same Tkinter window as in Figure 5.3, but with the 'Search' field containing the text 'pakistan' and the 'Sample Size' field containing the text '80'. The Twitter bird logo and the 'Submit' button remain the same.

Figure 5.4

Result for your account have been created

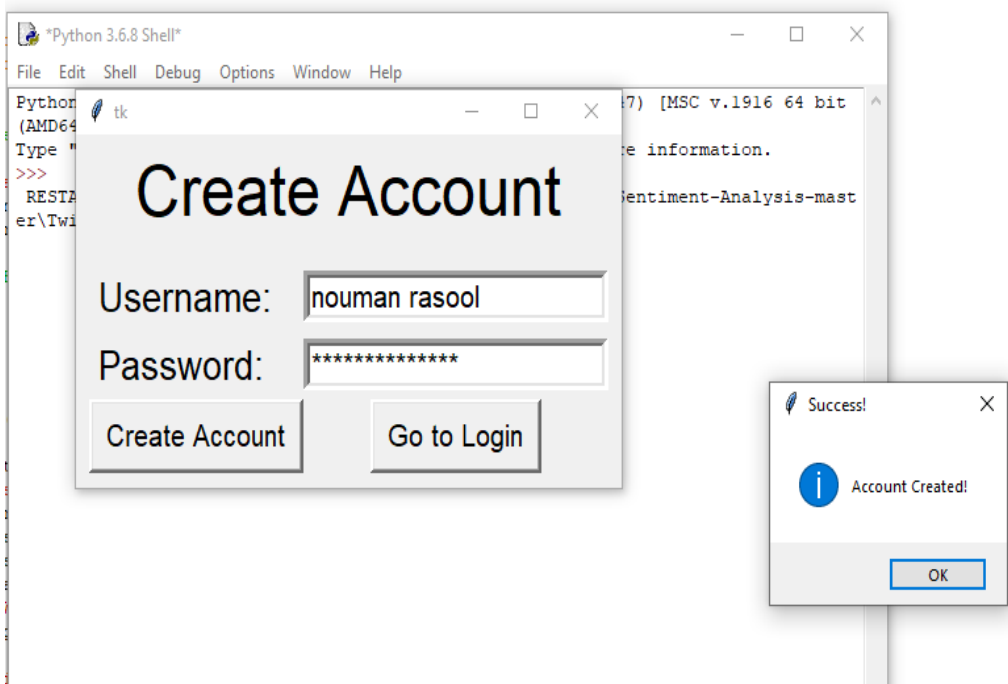


Figure 5.5

Result of Database

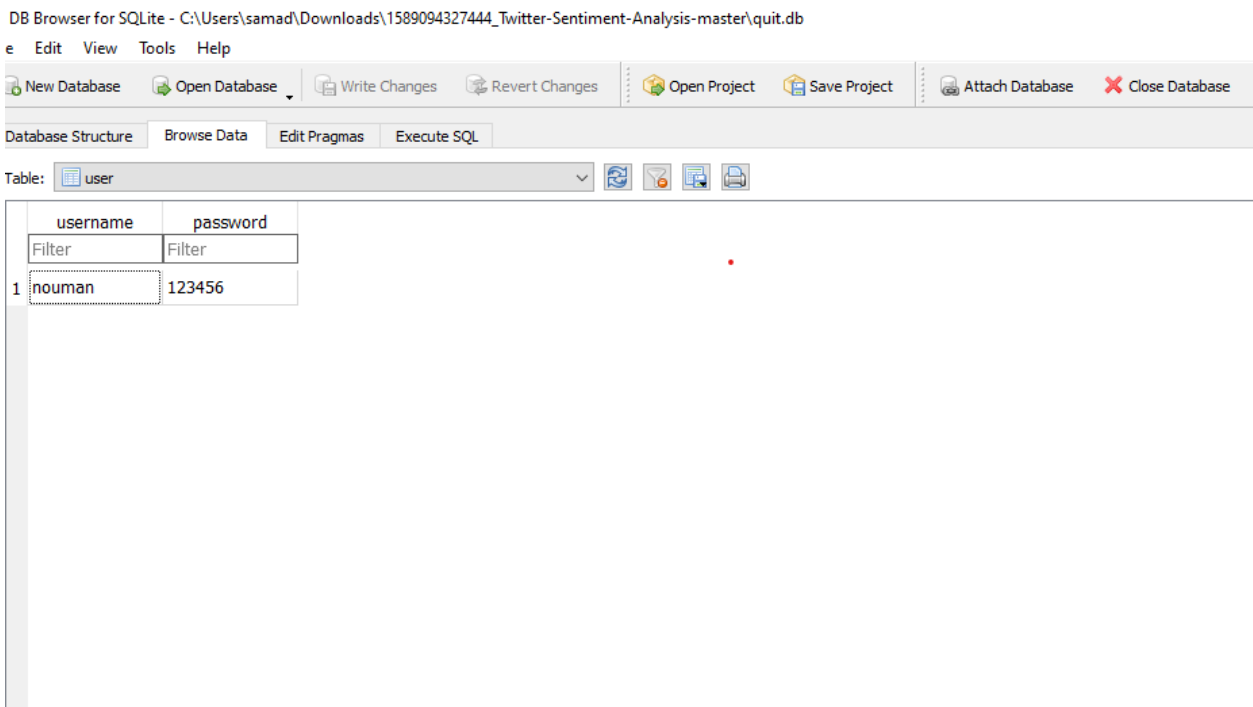


Figure 5.5

Graphical representation of result of search query

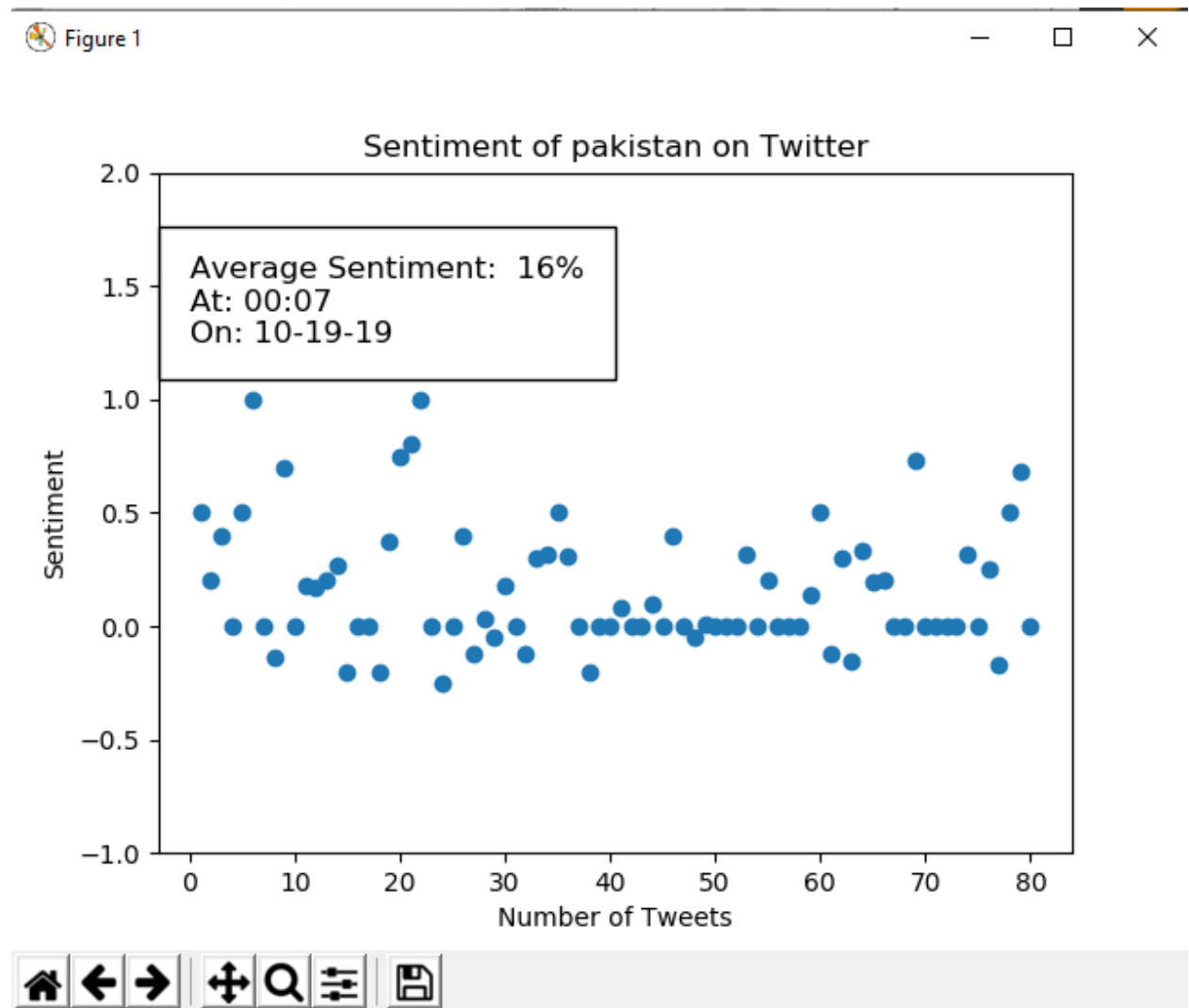


Figure 5.6

5.2 Testing of the system

Testing of a system is a stage of any application/software testing whereas the purpose of this test is to evaluate the system's compliance with the specified requirements: The process of testing an integrated system to verify that it meets specified

Purpose of testing:

The basic purpose of testing that the system meets the different requirements including, functional, performance, reliability, security, usability and so on. While finding bugs is one of the purposes of software testing

Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated.

Black Box Testing

It is also called as Behavioral/Specification-Based/Input-Output Testing Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing. Testers create test scenarios/cases based on software requirements and specifications. So it is AKA Specification Based Testing. Tester performs testing only on the functional part of an application to make sure the behavior of the software is as expected. So it is AKA Behavioral Based Testing.

White Box Testing

It is also called as Glass Box, Clear Box, and Structural Testing. White Box Testing is based on applications internal code structure. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually done at the unit level.

Test	Options Tested	Expected Result	Result
1.	Click on Add	Add information	As expected
2.	Click on Close	Close Form	As expected
3.	Click on Update	Update information	As expected
4.	Click on Save	Save information	As expected
5.	Click on Delete	Delete information	As Expected

Table 6.1 Testing

Integration Testing

It occurs after unit testing and before validation testing. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group.

Acceptance Testing

The main purpose of this testing technique is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

System Development

The systems development life cycle (SDLC), also referred to as the application development Life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system

5.3 Test Cases

Test cases or software system is working correctly or not. The mechanism for determining whether a software or application has passed or failed such a test is known as a test oracle usually collected into test suits. After deployment there is an important phase to test some of the important use cases these are known as test case these all are given on next pages.

Test case title: User Login

Test case1: User Login		Priority(H,L): High
Test objective: Verify User Login		
Test descriptions: “Admin enter the required field, press login button”, client program contact to the server, server contact with database, database update the result and send the results to the user.		
Requirements verified: Yes		
Test environment: Its must be in running state, Database should contain appropriate table and link must be established between server and client program.		
Test setup/ pre condition: IIS must be in running state. The entire mandatory field must be entered.		
Location	Expected result	
The user will login to access application	“User Login successfully”. Display main menu.	
Pass: Yes	Condition: Pass	Fail: No
Problem /issues: NIL		
status: successfully executed		

Appendix

A Sample Appendix

This is a sample Appendix.

.

References:

Books:

- Modern database Management 12th edition by Jeffery A. Hoffer
- Vijay Srinivasan Agneeswaran, Ph.D. “Big Data Analytics beyond Hadoop” book published by Pearson Education, Inc.

Site Address:

- <https://developer.twitter.com/en/apps>
- <https://pypi.org/project/matplotlib/>
- <https://pythonprogramming.net/gui-development-tutorials/>
- <https://pythonprogramming.net/tkinter-depth-tutorial-making-actual-program/>
- <https://www.sqlite.org/download.html>
- https://www.cse.ust.hk/~rossiter/independent_studies_projects/twitter_emotion_analysis/twitter_emotion_analysis.pdf
- <https://pdfs.semanticscholar.org/abbc/1e4150e8fc40300d3b5d1ac05153583f4f95.pdf>