

An Project report on

Semi Supervised Machine Learning Approach For Detecting DDoS Attack

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

C.LAVANYA

164G1A0547

B.CHANDRIKA

164G1A0518

K.N.V.KISHORE KUMAR

164G1A0538

T.KUMUDA

164G1A0542

Under the Guidance of

Mr.B.Sreedhar, M.Tech

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY ANANTHAPURAMU

(Accredited by NAAC with 'A' Grade & Accredited by NBA, Affiliated to JNTUA, Approved by AICTE, New Delhi)

2019-2020

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPUR

(Accredited by NAAC with 'A' Grade, Affiliated to JNUTA, Approved by AICTE, New Delhi)

Rotarypuram Village, B K Samudram Mandal, ananthapuramu-515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(B.Tech program accredited by NBA)



Certificate

This is to certify that a seminar report entitled **SEMI SUPERVISED MACHINE LEARNING APPROACH FOR DETECTING DDOS ATTACK** is the bonafide work carried out by **C.LAVANYA** bearing Roll Number **164G1A0547**, **B.CHANDRIKA** bearing Roll Number **164G1A0518**, **K.N.V.KISHORE KUMAR** bearing Roll Number **164G1A0538**, **T.KUMUDA** bearing Roll Number **164G1A0542** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2019-2020.

Signature of the Guide

Mr.B.Sreedhar, M.Tech.,
Assistant Professor

Head of the Department

Dr.G.K.V.Narashima Reddy, Ph.D,
Professor

Date:

Place: Rotarypuram

EXTERNAL EXAMINAR

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Mr.B.Sreedhar**, M.Tech., **Assistant Professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Mr. R. Sandeep Kumar**, M.Tech., **Assistant Professor**, project coordinator valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to **Dr. G. K. V. NARASIMHA REDDY**, Ph.D, **Professor and Head of Department, Computer Science & Engineering**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr. T. Hitendra Sarma**, Ph.D, **Principal, Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

DECLARATION

We, Ms. C.Lavanya having reg no: 164g1a0547, Ms. B.Chandrika having reg no: 164g1a0518, Mr. K.N.V. Kishore Kumar having reg no: 164g1a0538 and T.Kumuda having reg no:164g1a0542 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that dissertation entitled “SEMI SUPERVISED MACHINE LEARNING APPROACH FOR DETECTING DDoS ATTACK ” embodies the report of our project work carried out by us during IV year Bachelor of Technology in Mr. B.Sreedhar, M.Tech., Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, ANANTAPUR and this work has been submitted for the partial fulfilment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

C.LAVANYA

Reg no:164G1A0547

B.CHANDRIKA

Reg no:164G1A0518

K.N.V.KISHORE KUMAR

Reg no:164G1A0538

T.KUMUDA

Reg no:164G1A0542

CONTENTS

Abstract	v
List of Figures	vi
List of Screens	vii
1. INTRODUCTION	
1.1 Motivation	1
1.2 Problem Definition	2
2. LITERATURE SURVEY	
2.1 Introduction	4
2.2 Existing System	5
2.3 Disadvantages of Existing System	5
2.4 Proposed System	5
3. FEASIBILITY STUDY	
3.1 Economic Feasibility	6
3.2 Technical Feasibility	6
3.3 Social Feasibility	7
4. REQUIREMENTS	
4.1 Software Requirement Specification	8
4.1.1 User Requirements	9
4.1.2 Software Requirements	9
4.1.3 Hardware Requirements	10
5. ANALYSIS	
5.1 Introduction to Django	11
5.1.1 Advantages of Django framework	14
5.2 Installation of Python	16
5.3 Installing WAMP Server	18
5.4 Installation of PyCharm	21
5.5 HTML	25
5.5.1 Importance of HTML	25
5.5.2 Functionality of HTML in the project	26
6. DESIGN	
6.1 UML Introduction	27
6.2 Why Use UML in Projects?	27
6.3 Diagrams	32
6.3.1 UML Diagrams	32
6.3.1.1 Activity Diagram	33
6.3.1.2 Use Case Diagram	34
6.3.1.3 Sequence diagram	35
7. MODEL	
7.1 What is SDLC?	37
8. IMPLEMENTATION	39

9. TESTING	
9.1 Types of tests	41
9.1.1 Unit testing	41
9.1.2 Integration testing	41
9.1.3 Functional test	42
9.1.4 System test	42
9.1.5 White Box Testing	42
9.1.6 Black Box Testing	42
9.1.7 Test strategy and approach	43
10. EXECUTION AND RESULTS	
10.1 Introduction	44
10.2 Output screens	44
CONCLUSION	51
REFERENCES	52

ABSTRACT

Despite the important evolution of the information security technologies in recent years, the DDoS attack remains a major threat of Internet. The attack aims mainly to deprive legitimate users from Internet resources. The impact of the attack relies on the speed and the amount of the network traffic sent to the victim.

Even though advanced Machine Learning (ML) techniques have been adopted for DDoS detection, the attack remains a major threat of the Internet. Most of the existing ML-based DDoS detection approaches are under two categories: supervised and unsupervised. Supervised ML approaches for DDoS detection rely on availability of labelled network traffic datasets. Whereas, unsupervised ML approaches detect attacks by analyzing the incoming network traffic. Both approaches are challenged by large amount of network traffic data, low detection accuracy and high false positive rates.

We present an online sequential semi-supervised ML approach for DDoS detection. Semi supervised machine learning approach is the combination of supervised and unsupervised machine learning approaches. The unsupervised part of the approach allows to reduce the irrelevant normal traffic data for DDoS detection which allows to reduce false positive rates and increase accuracy. Whereas, the supervised part allows to reduce the false positive rates of the unsupervised part and to accurately classify the DDoS traffic.

LIST OF FIGURES

Fig.No	Description	Pg.No
1	DDoS Architecture	1
2	Architecture of detection of DDoS attack	3
3	Django architecture	11
4	Python Welcome Window	16
5	Download python	16
6	Installation window	17
7	Installation Completed Window	18
8	Select WampServer version	19
9	Download WampServer	19
10	Choosing language	19
11	Setup WampServer	19
12	Accept the license agreement to continue installation	20
13	Setup location to store files	20
14	Installing WampServer	21
15	Downloading PyCharm	22
16	Setup PyCharm	22
17	Select location	23
18	Installation options	23
19	Choose Start menu folder	24
20	Installing PyCharm	24
21	Leave the Run PyCharm Community Edition	25
22	Class Structure	28
23	Actors and Use cases	29
24	Objects and Timestamps	29
25	Activities	30
26	Collaborations	30
27	Deployment	31
28	States and Relationships	31
29	Components	31
30	Activity Diagram	33
31	Use Case Diagram	35
32	Sequence Diagram	36

LIST OF SCREENS

S.No	Screens	Pg.No
1	Login Page	44
2	DDoS Dataset analysis	45
3	Labeled Data	45
4	Unlabeled Data	46
5	Manual adding Data	46
6	DDoS analysis	47
7	Graphical analysis--Column chart	47
8	Graphical analysis--Bar Chart	48
9	Graphical analysis--Spline Chart	48
10	Stopping DDoS attacks	49
11	Sending E-mails	49
12	E-mail	50

CHAPTER 1

INTRODUCTION

1.1 Motivation

In recent years, widespread adoption of the internet has resulted in to rapid advancement in information technologies. The internet is used by the general population for the purposes such as financial transactions, educational endeavours, and countless other activities. The use of the internet for accomplishing important tasks, such as transferring a balance from a bank account, always comes with a security risk. Today's web sites strive to keep their users' data confidential and after years of doing secure business online, these companies have become experts in information security. The database systems behind these secure websites store non-critical data along with sensitive information, in a way that allows the information owners quick access while blocking break-in attempts from unauthorized users.

A Distributed Denial of Service (DDoS) attack is a non-intrusive internet attack made to take down the targeted website or slow it down by flooding the network, server or application with fake traffic. When against a vulnerable resource-intensive endpoint, even a tiny amount of traffic is enough for the attack to succeed.

Distributed Denial of Service (DDoS) attacks are threats that website owners must familiarize themselves with as they are a critical piece of the security landscape. Navigating the various types of DDoS attacks can be challenging and time consuming. To help you understand what a DDoS attack is and how to prevent it, we have written the following guide.

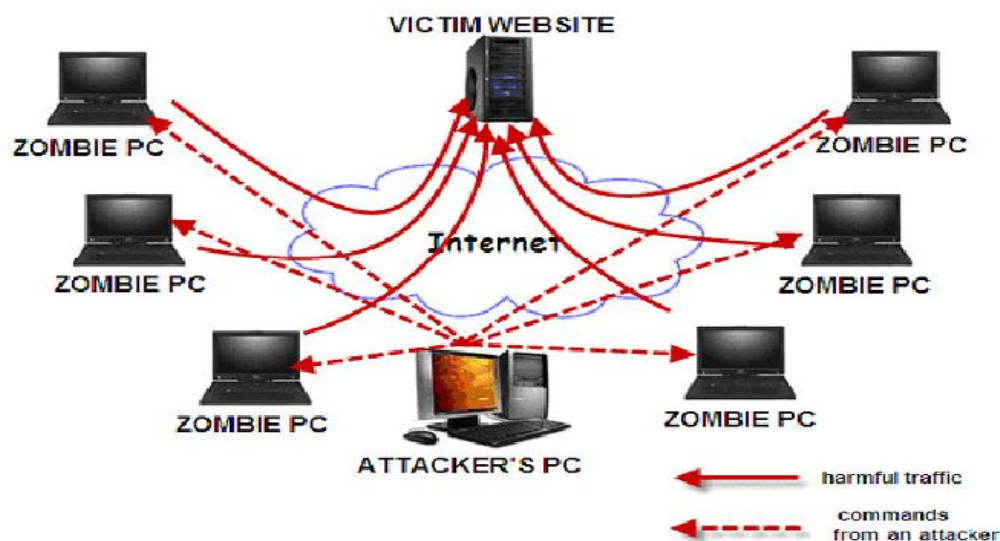


Figure1:DDoS Architecture

1.2 Problem Definition

There are two categories of the DDoS attack namely Direct DDoS attack and Reflection-based DDoS. In the Direct DDoS attack the attacker uses the zombie hosts to flood directly the victim host with a large number of network packets. Whereas, in the Reflection based DDoS attack the attacker uses the zombie hosts to take control over a set of compromised hosts called Reflectors. The latter are used to forward a massive amount of attack traffic to the victim host. Recently, destructive DDoS attacks have brought down more than 70 vital services of Internet including Github, Twitter, Amazon, Paypal, etc.

Most of the existing ML-based DDoS detection approaches are under two categories: supervised and unsupervised. Supervised ML approaches for DDoS detection rely on availability of labeled network traffic datasets. Whereas, unsupervised ML approaches detect attacks by analyzing the incoming network traffic. Both approaches are challenged by large amount of network traffic data, low detection accuracy and high false positive rates.

The appearance of malicious apps is a serious threat to the Android platform. Most types of network interfaces based on the integrated functions, steal users' personal information and start the attack operations. In this paper, we propose an effective and automatic malware detection method using the text semantics of network traffic. In particular, we consider each HTTP flow generated by mobile apps as a text document, which can be processed by natural language processing to extract text-level features. Later, the use of network traffic is used to create a useful malware detection model. We examine the traffic flow header using N-gram method from the natural language processing (NLP). Then, we propose an automatic feature selection algorithm based on chi-square test to identify meaningful features. It is used to determine whether there is a significant association between the two variables. We propose a novel solution to perform malware detection using NLP methods by treating mobile traffic as documents. We apply an automatic feature selection algorithm based on N-gram sequence to obtain meaningful features from the semantics of traffic flows. Our methods reveal some malware that can prevent detection of antiviral scanners. In addition, we design a detection system to drive traffic to your own-institutional enterprise network, home network, and 3G / 4G mobile network. Integrating the system connected to the computer to find suspicious network behaviors.

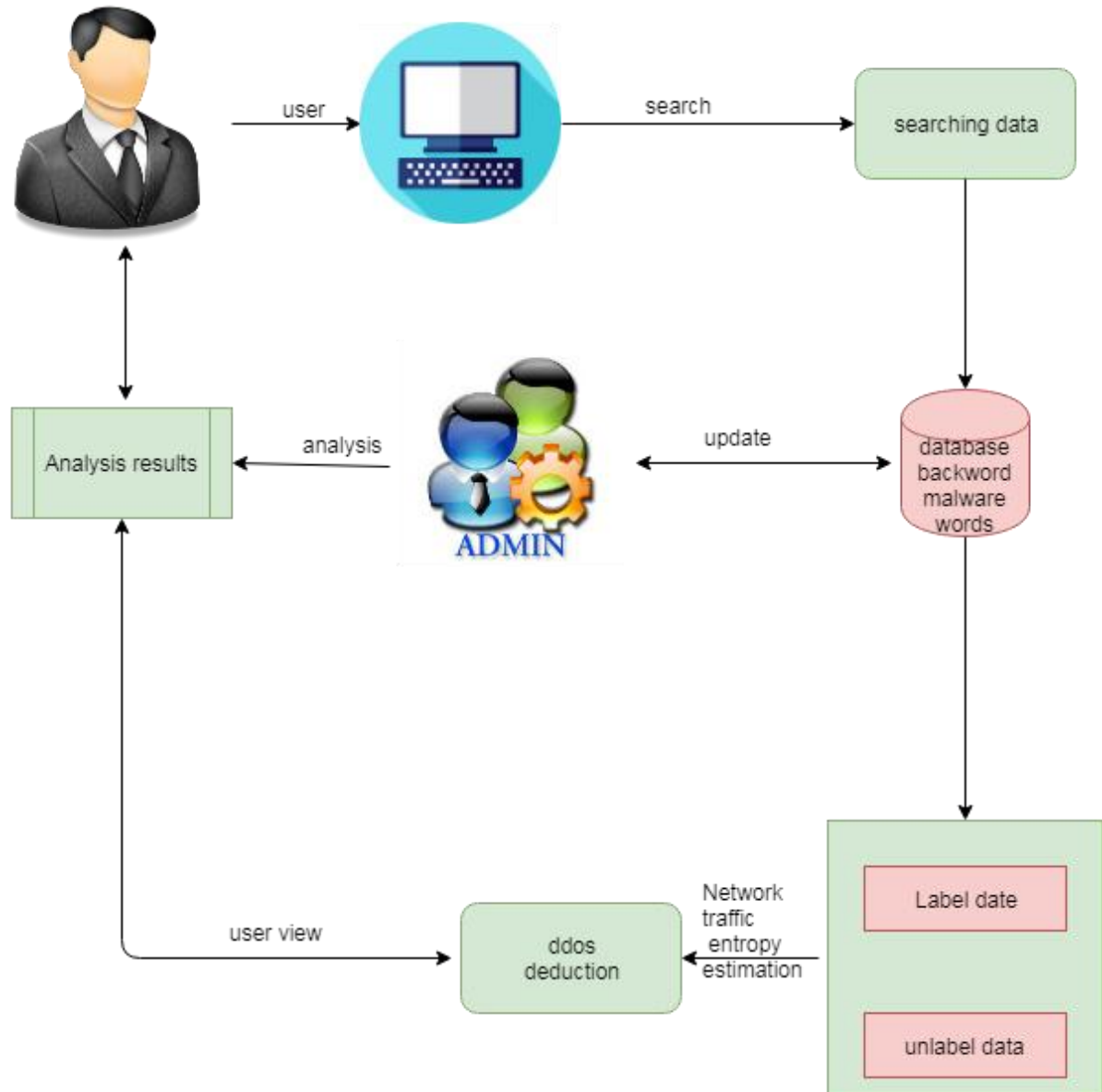
ARCHITECTURE:

Figure 2:Architecture of detection of DDoS attack

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction

1.Idhammad M, Afdel K, Belouch M (2017) Dos detection method based on artificial neural networks. Int J Adv Comput Sci Appl(ijacsa) 8(4):465–471.

DoS attack is a major Internet security problem-DoS is that lots of clients simultaneously send service requests to certain server on the internet such that this server is too busy to provide normal services for others. Attackers using legitimate packets and often changing package information, so that traditional detection methods based on feature descriptions is difficult to detect it.

This paper present an artificial intelligence DoS attack detection method based on neural networks. In this method, analysis of server resources and network traffic, To training the ability of detection normal or abnormal, it have better results for detect DoS attack.

2.Papalexakis EE, Beutel A, Steenkiste P (2014) Network anomaly detection using co-clustering. In: Encyclopedia of social network analysis and mining. Springer, Berlin, pp 1054–1068.

Early Internet architecture design goals did not put security as a high priority. However, today Internet security is a quickly growing concern. The prevalence of Internet attacks has increased significantly, but still the challenge of detecting such attacks generally falls on the end hosts and service providers, requiring system administrators to detect and block attacks on their own. In particular, as social networks have become central hubs of information and communication, they are increasingly the target of attention and attacks. This creates a challenge of carefully distinguishing malicious connections from normal ones. Previous work has shown that for a variety of Internet attacks, there is a small subset of connection measurements that are good indicators of whether a connection is part of an attack or not.

In this paper we look at the effectiveness of using two different co-clustering algorithms to both cluster connections as well as mark which connection measurements are strong indicators of what makes any given cluster anomalous relative to the total data set. We run experiments with these co-clustering algorithms on the KDD 1999 Cup data set. In our experiments we find that soft co-clustering, running on samples of data, finds consistent parameters that are strong indicators of anomalous detections and creates clusters, that are

highly pure. When running hard co-clustering on the full data set (over 100 runs), we on average have one cluster with 92.44% attack connections and the other with 75.84% normal connections.

These results are on par with the KDD 1999 Cup winning entry, showing that co-clustering is a strong, unsupervised method for separating normal connections from anomalous ones. Finally, we believe that the ideas presented in this work may inspire research for anomaly detection in social networks, such as identifying spammers and fraudsters

2.2 Existing Systems

The existing Machine Learning based DDoS detection approaches can be divided into three categories. Supervised ML approaches that use generated labeled network traffic datasets to build the detection model. Two major issues are facing the supervised approaches. First, the generation of labeled network traffic datasets is costly in terms of computation and time.

Without a continuous update of their detection models, the supervised machine learning approaches are unable to predict the new legitimate and attack behaviors. Second, the the presence of large amount of irrelevant normal data in the incoming network traffic is noisy and reduces the performances of supervised ML classifiers.

2.3 Disadvantage of Existing system

- 1) The main drawback of the unsupervised approaches is the high false positive rates.
- 2) The supervised machine learning approaches are unable to predict the new legitimate and attack behaviors

2.4 Proposed System

It is online sequential semi supervised ML approach for DDoS detection is implemented, A time based sliding window algorithm is used to estimate the entropy of the network header features of the incoming network traffic. Combining both previous algorithms in a sophisticated semi-supervised approach for DDoS detection. This allows to achieve good DDoS detection performance compared to the state-of-the-art DDoS detection methods. The unsupervised part of the approach allows to reduce the irrelevant normal traffic data for DDoS detection which allows to reduce false positive rates and increase accuracy. Whereas, the supervised part allows to reduce the false positive rates of the unsupervised part and to accurately classify the DDoS traffic.

CHAPTER-3

FEASIBILITY STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.

For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

3.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also

able to make some constructive criticism, which is welcomed, as he is the final user of the system. The initial steps in carrying out a feasibility analysis for a Web project, based on information obtained interactively from a prospective client, are modelled as a rule-based expert system that draws on empirical formulas to provide a quantitative valuation of the clients replies. The model is centred in the pre-design phase, concerned in particular with ascertaining whether coherent aims, message, audience and services have been identified for the Web project. The model is constructed empirically by means of a knowledge acquisition process.

The various data collected from the user are described, as well as the way in which these data are translated into quantitative values that subsequently play a role in the final diagnosis for the Web project. The system has been evaluated in two ways: system users were asked to provide feedback, and the experts who took part in the knowledge acquisition phase were asked to review system performance.

CHAPTER-4

REQUIREMENTS

4.1 Software Requirement Specification

Software Requirement Specification (SRS) is the starting point of the software development activity. It is a complete description of the behaviour of a system which is to be developed. The SRS document enlists all necessary requirements for project development. To derive the requirements we need to have clear and thorough understanding of the product which is to be developed. This is prepared after detailed communication with project team and the customer.

A SRS is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations.

Characteristics of SRS:

Correct –

An SRS is correct if, and only if, every requirement stated therein is one that the software shall meet. Traceability makes this procedure easier and less prone to error.

Unambiguous –

An SRS is unambiguous if, and only if, every requirement stated there in has only one interpretation. As a minimum, this requires that each characteristic of the final product be described using a single unique term.

Verifiable –

It is verifiable if there exists some finite cost-effective process with which a person or machine check whether software product meets requirements.

Consistent –

Consistency refers to internal consistency. If an SRS does not agree with some higher-level document, such as a system requirements specification, then it is not correct. An SRS is internally consistent if, and only if, no subset of individual requirements described in it conflict.

Modifiable –

SRS is said to be modifiable if its structure and style are such that any changes to the requirements can be made easily, completely and consistently while retaining the structure and style.

Traceable –

SRS is said to be traceable if the origin of each of its requirements is clear and it facilitates the referencing of each requirement in future enhancement.

Ranked for importance or stability –

SRS is ranked for importance or stability if each requirement in it has an identifier to indicate either the importance or stability of that particular requirement.

4.1.1 User Requirements

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system engineering and refined by establishing a complex information description, detailed functional and behavioural description, and indication of performance requirements and design constraints, appropriate validation criteria and other data pertinent to requirements.

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

4.1.2 Software Requirements

- Operating system: Windows 7 or above.
- Coding Language: Python.
- Front-End: Python.
- Designing: Html, CSS, JavaScript.
- Data Base: MySQL.
- Tools: PyCharm

4.1.3 Hardware Requirements

For developing the application the following are the Hardware Requirements:

- CPU type : Intel Pentium V

- Hard Disk : 16 GB available hard disk space (32-bit) or 20GB (64-bit)
- Ram : 4 GB.
- Hard Disk : 40 GB
- Floppy Drive : 1.44 Mb

CHAPTER-5

ANALYSIS

5.1 Introduction to Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django 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 is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

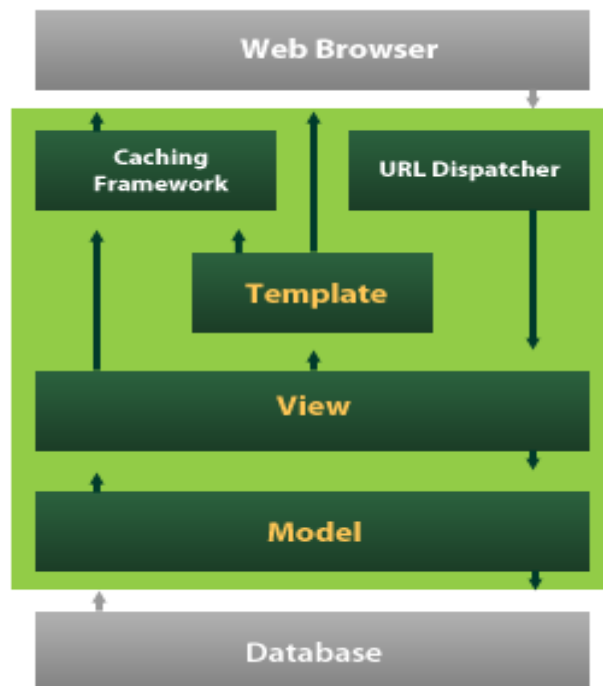


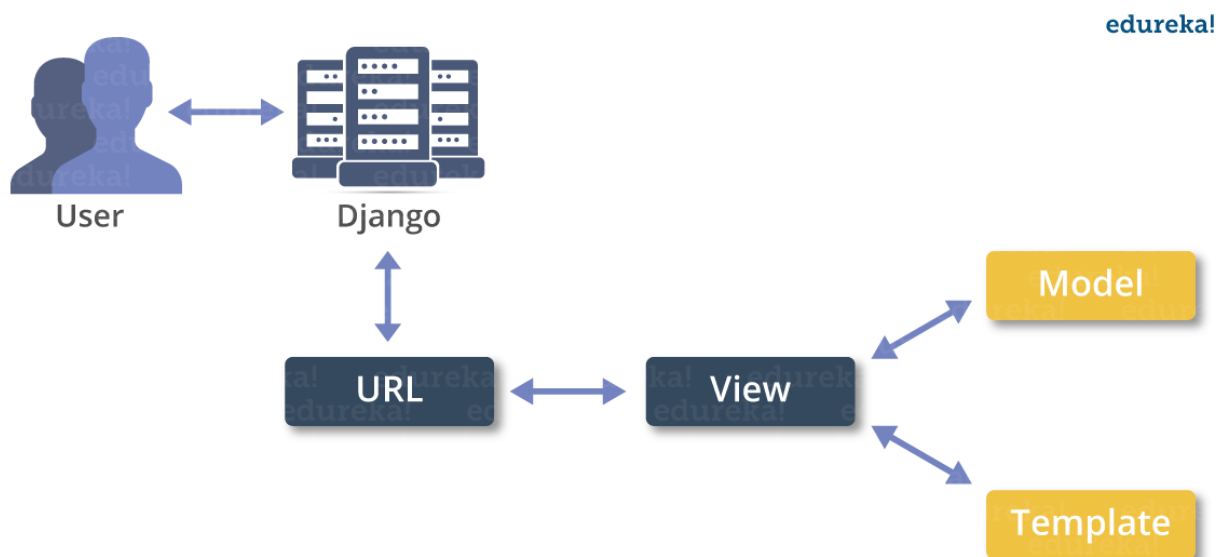
Figure 3. Django architecture

- 1.The URL dispatcher(urls.py)maps the requested URL to a view function and calls it. If caching is enabled, the view function can check to see if a cached version of the page exists and bypass all further steps, returning the cached version, instead. Note that this page-level caching is only one available caching option in Django. You can cache more granularly, as well.
- 2.The view function(usually in views.py)performs the requested action, which typically involves reading or writing to the database. It may include other tasks, as well.

3.The model(usually in models.py)defines the data in Python and interacts with it. Although typically contained in a relational database (MySQL, PostgreSQL, SQLite, etc.),other data storage mechanisms are possible as well(XML, text files, LDAP ,etc.).

4.After performing any requested tasks, the view returns an HTTP response object(usually after passing the data through a template)to the web browser. Optionally, the view can save a version of the HTTP response object in the caching system for a specified length of time.

5.Templateds typically return HTML pages. The Django template language offers HTML authors a simple-to-learn syntax while providing all the power needed for presentation logic. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models



Features of Django

- Rapid Development
- Secure
- Scalable
- Fully loaded
- Versatile
- Open Source
- Vast and Supported Community

Rapid Development:

Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

Secure:

Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Scalable:

Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

Fully loaded:

Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

Versatile:

Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

Open Source:

Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.

Vast and Supported Community:

Django is an one of the most popular web framework. It has widely supportive community and channels to share and connect

5.1.1 Advantages of Django framework:

Accelerates custom web application development

Django is one of the most mature web frameworks for Python. Its design rules focus extensively on reducing web application development time. The features provided by Django enable developers to build custom web applications rapidly according to varying business requirements. A large percentage of Python programmers even opt for Django when they have to meet both goals and deadlines.

Written in Python

Django is one of the web frameworks which are written in Python programming language. Hence, it becomes easier for programmers to build web applications with clean, readable, and maintainable code by taking advantage of syntax rules of Python. Also, the developers can easily curtail the development time by building custom web applications without writing additional code.

Designed as a batteries-included web framework

Django is one of the web frameworks that adopt the batteries-included approach. While developing a custom web application, Django provides the resources required by developers out of the box. It provides code for common operations like database manipulation, HTML templating, URL routing, session management, and security. The batteries included approach help developers to curtail web application development time significantly.

Supports MVC programming paradigm

Django, like other modern web frameworks, supports model-view-controller (MVC) design rule. The MVC programming paradigm allows programmers to keep a web application's user interface (UI) and business logic layers separated. The approach further helps programmers to simplify and speed up development of large web applications by separating their user interface and business logic layers. Django further allows programmers to reuse the same business logic across multiple projects.

Compatible with major operating systems and databases

Nowadays, users access web applications on various devices and platforms. Django enhances the accessibility of web applications by supporting major operating systems like Windows, Linux and MacOS. At the same time, the ORM system provided by Django makes it easier for programmers to work with several widely used databases. They can even use the ORM system to perform common database operations and migrate from one database to another without writing additional code.

Provides robust security features

The built-in security features provided by Django help developers to protect the web applications from a variety of targeted security attacks – cross-site scripting, SQL injection and cross-site request forgery. At the same time, the web framework enhances the security of web applications by preventing common security mistakes related to Python coding.

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. Python, 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

5.2 Installation of Python

Go to www.python.org

Click “Downloads” Link at the top of the page.

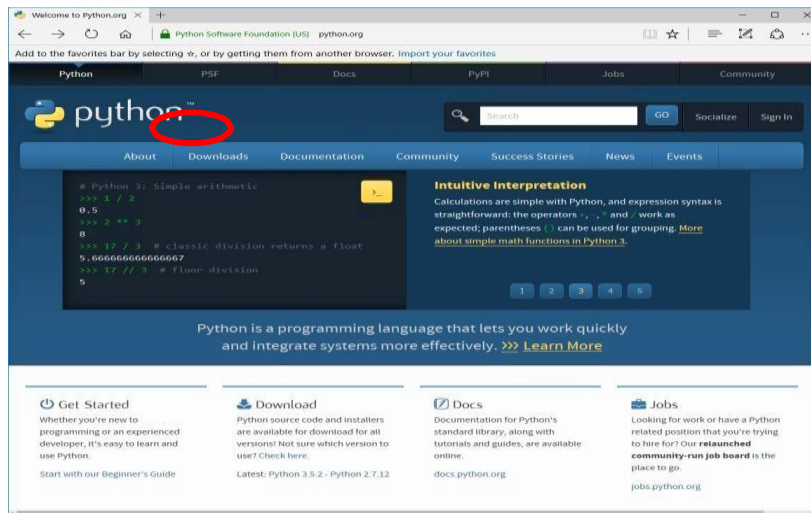


Figure 4. Python Welcome Window

Click “Download Python 3.6.2” or whatever the 3.X version currently is:

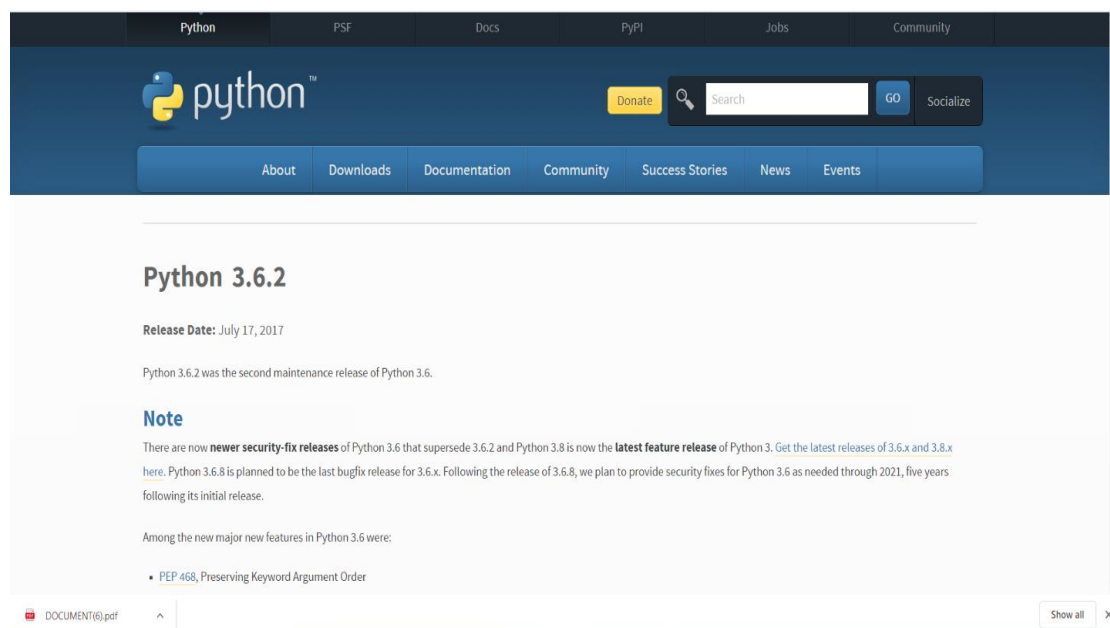


Figure 5. Download python

- When the installation window comes up, click “Install Now”
- You can choose to check the “Install launcher for all users (recommended)” or not either way should be OK

- You can choose to “Add Python 3.6 to PATH” or not – either way should be OK Note: Depending on how Windows is set up, you might need to provide an administrator password to install on your system at this point.
- You can choose to “Customize Installation” if you want, especially if you want to install to a location other than the default one shown. Generally, I recommend installing to the default location unless you have a problem doing so.

In any case, you might want to note the location of the installation in case you have difficulty later. If you are specifying the location yourself, put it in a location you are likely to easily find/remember.

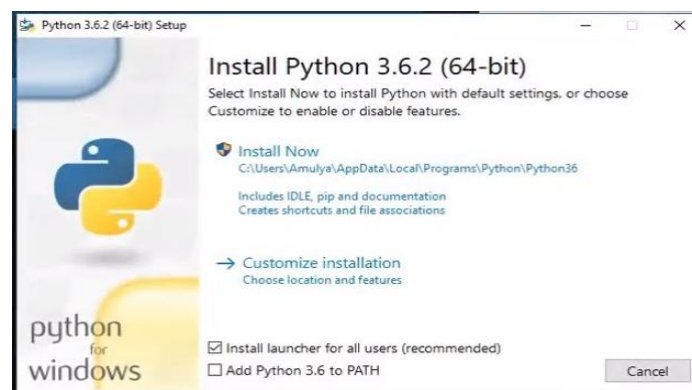


Figure 6. Installation window

- You should see Python installing at this point. When it finishes, you should see a screen that says the installation was successful. You can click “Close”

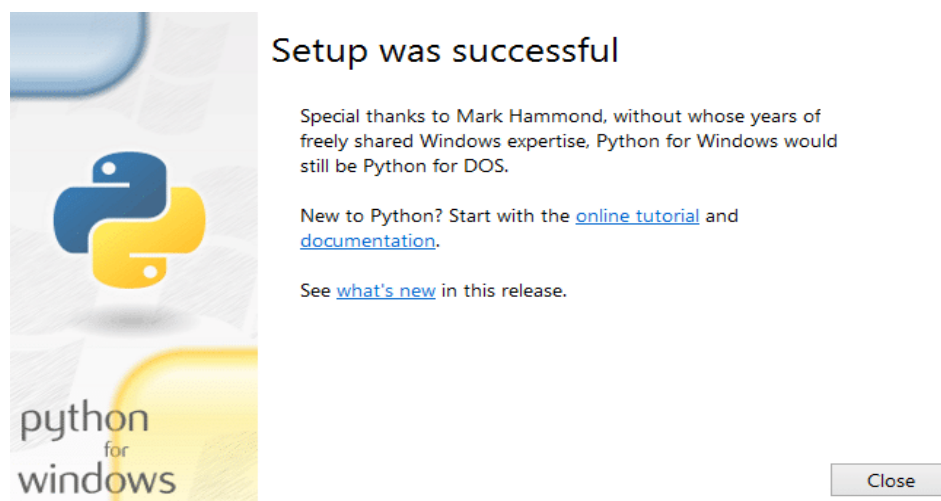


Figure 7. Installation Completed Window

5.3 Installing WAMP Server

1: Download the WAMP Server

Go to the official website www.wampserver.com/en/ and download the WampServer setup. There are two versions of WampServer are available i.e. 64-bits (x64) and 32-bits (x86), choose according to your computer's configuration.



Figure 8. Select WampServer version

As soon as you will click on the download option, a pop up will appears showing some warnings. Just don't worry about these warning rather simply click on the link "download directly" like shown below and move ahead with the download process.

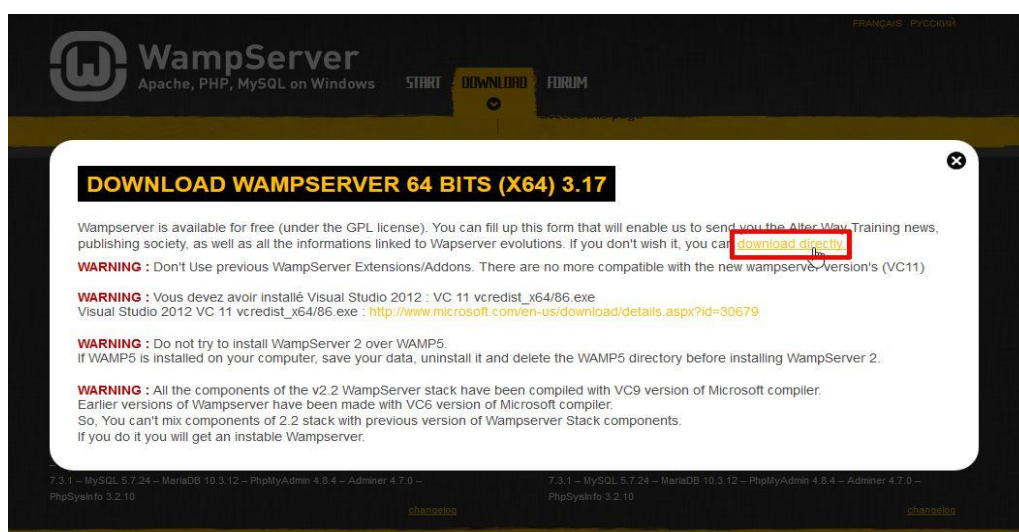


Figure 9. Download WampServer

2: WAMP server Installation

Run the setup and select the language in which you want to install the Wamp server or Windows 10.



Figure 10. Choosing language

Click next to continue

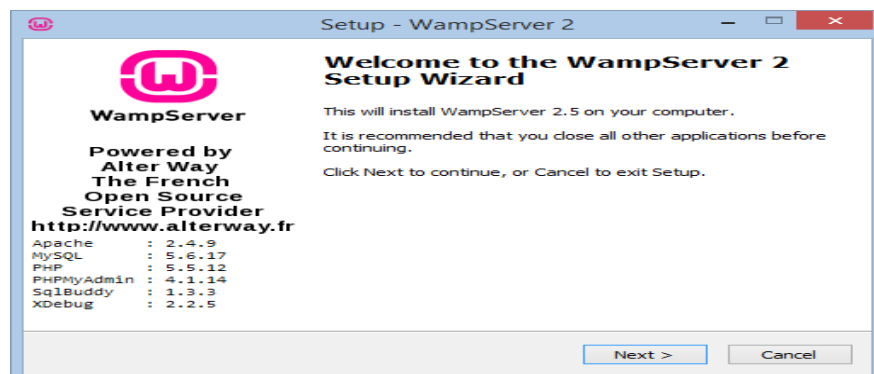


Figure 11. Setup WampServer

Select the "I accept the agreement"

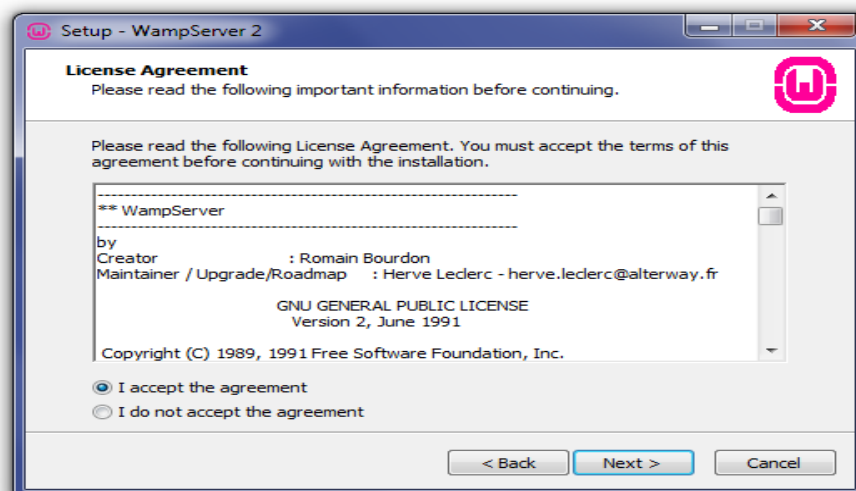


Figure 12. Accept the license agreement to continue installation

Basically here we can configure installation location in local PC hard drive. So, type in the exact location and click next to move with further steps.

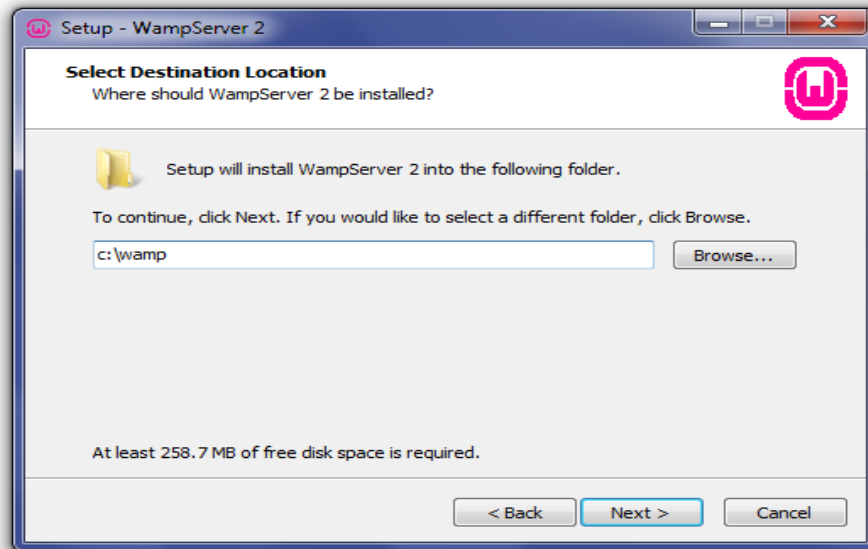
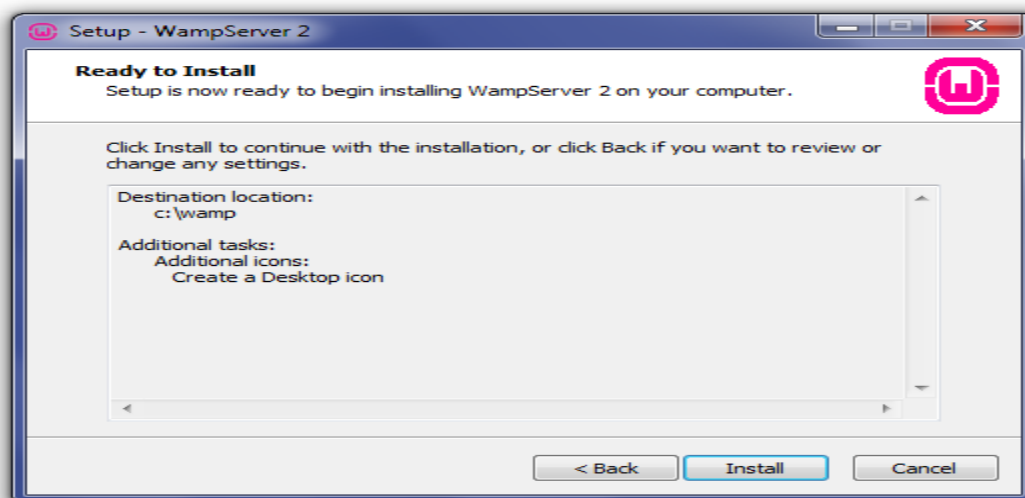


Figure 13. Setup location to store files

Next screen is a screen where we can set the icons both in Windows Quick Launch & Desktop. For example if we want to show the desktop icon we'll tick the square box in front of that option.



However by clicking next we're finally on the installation. Here the screen also show the options selected for installation to review. If everything goes right click Install and it'll begin to extract the files to install.

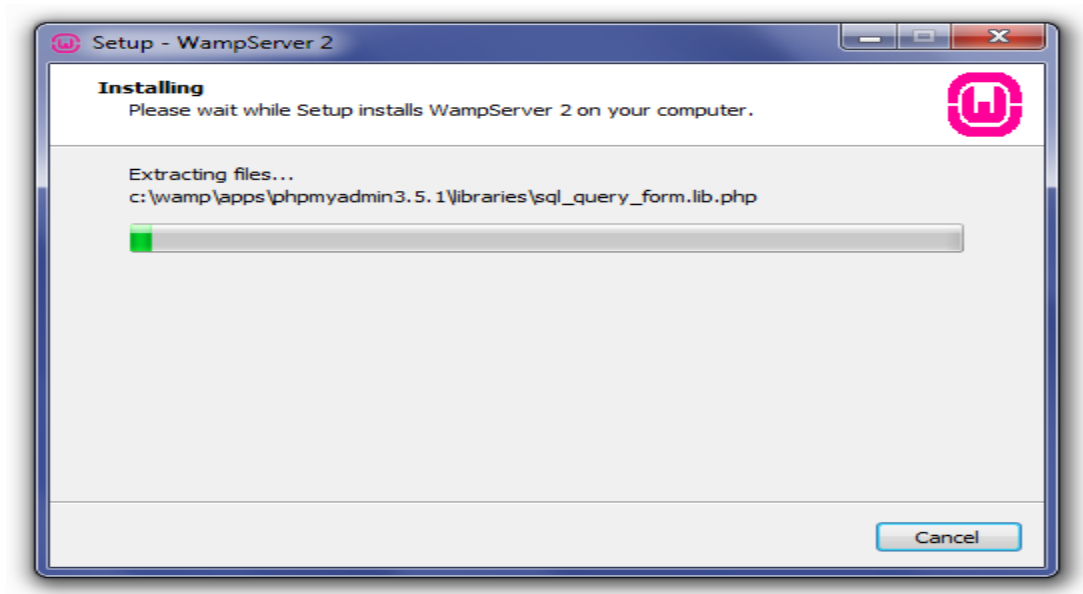


Figure 14. Installing WampServer

5.4 Installation of PyCharm

PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactorings and rich navigation capabilities.

PyCharm's smart code editor provides first-class support for Python, JavaScript, CoffeeScript, TypeScript, CSS, popular template languages and more. Take advantage of language-aware code completion, error detection, and on-the-fly code fixes!

Use smart search to jump to any class, file or symbol, or even any IDE action or tool window. It only takes one click to switch to the declaration, super method, test, usages, implementation, and more.

PyCharm's huge collection of tools out of the box includes an integrated debugger and test runner; Python profiler; a built-in terminal; integration with major VCS and built-in database tools; remote development capabilities with remote interpreters; an integrated ssh terminal; and integration with Docker and Vagrant.

Installing PyCharm

Step1) To download PyCharm visit the website: <https://www.jetbrains.com/pycharm/download/> Click the "DOWNLOAD" link under the Community Section.

Download PyCharm

Windows

macOS

Linux

Professional

Full-featured IDE
for Python & Web
development

DOWNLOAD

Free trial

Community

Lightweight IDE
for Python & Scientific
development

DOWNLOAD

Free, open-source

Figure 15. Downloading PyCharm

Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click "Next".

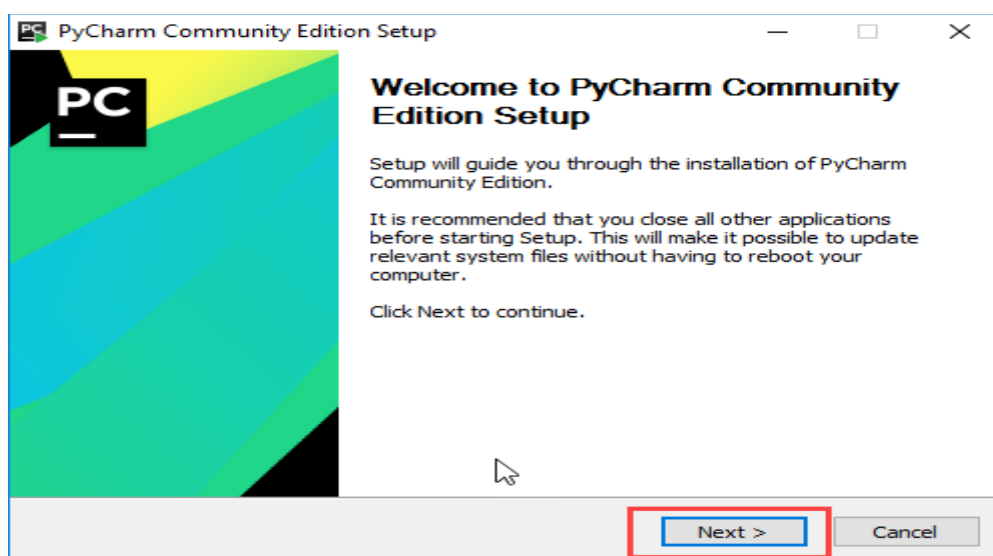


Figure 16. Setup PyCharm

Step 3) On the next screen, Change the installation path if required. Click “Next”.

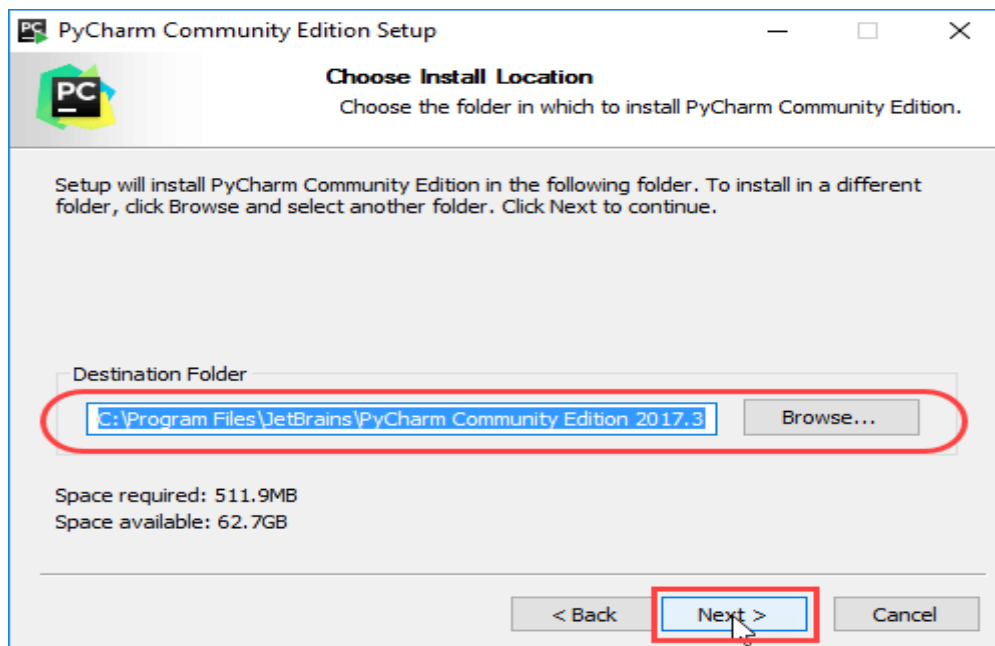


Figure 17. Select location

Step 4) On the next screen, you can create a desktop shortcut if you want and click on “Next”.

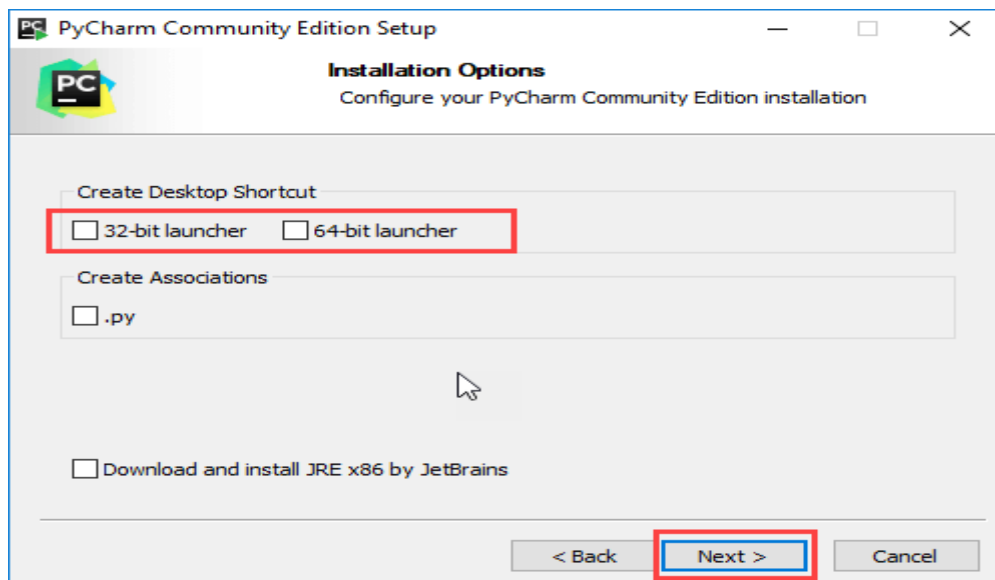


Figure 18. Installation options

Step 5) Choose the start menu folder. Keep selected JetBrains and click on “Install”.

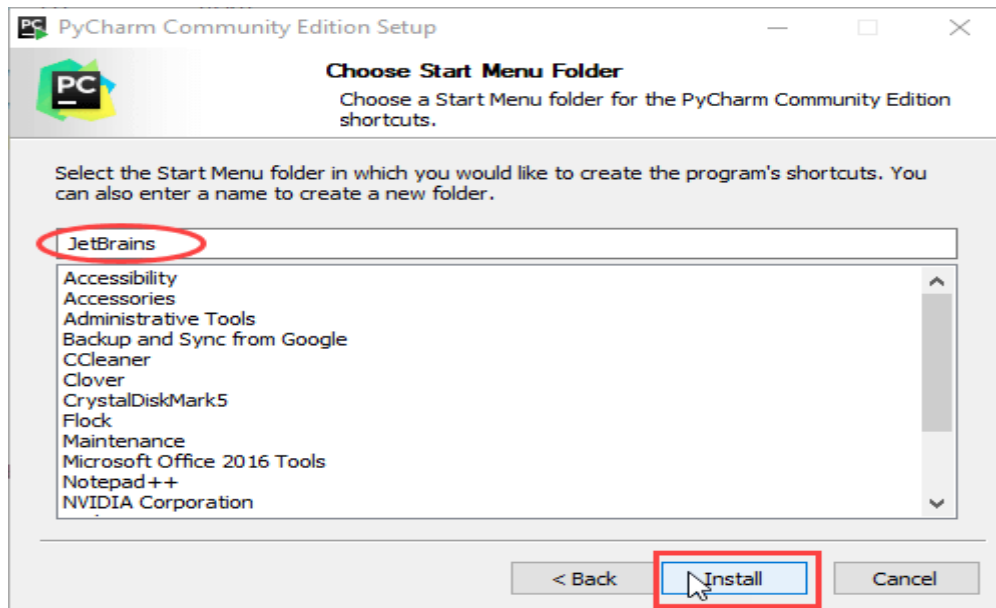


Figure 19. Choose Start menu folder

Step 6) Wait for the installation to finish.

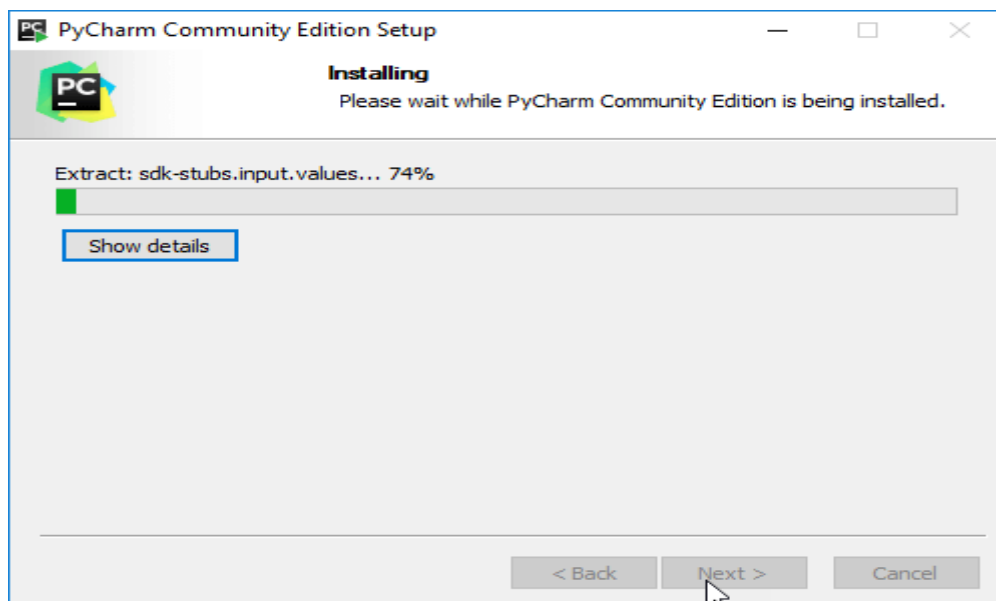


Figure 20. Installing PyCharm

Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.

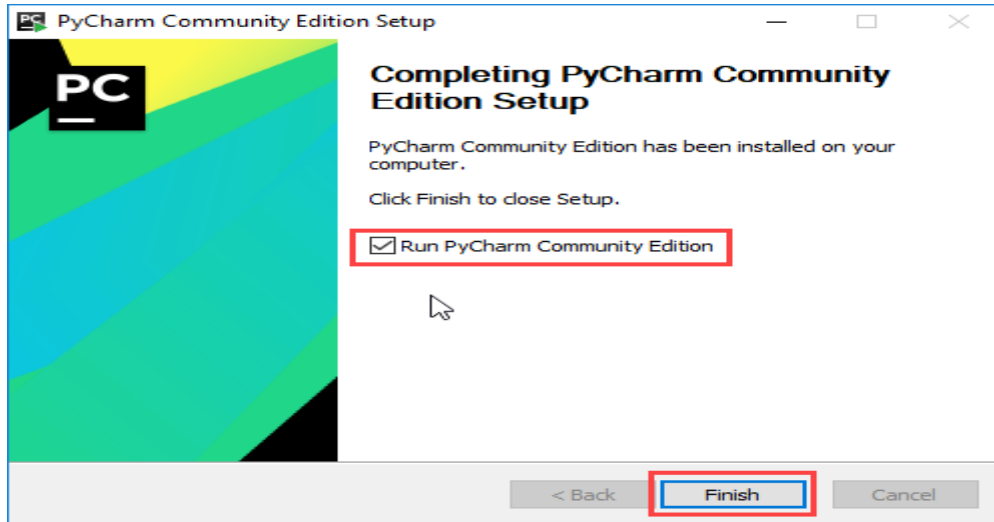


Figure 21. Leave the Run PyCharm Community Edition check box checked to run this software.

5.5 HTML

Hyper Text Mark up Language is a structural mark-up language used to create and format web document. A mark-up language such as HTML is simply a collection of codes, called Elements that are used to indicate the structure and format of a document. A user agent, usually a web browser that renders the document, interprets the meaning of these codes to figure how to structure or display a document. HTML is not invention but it is an improved version of Standard Generalized Mark-up Language (SGML).

HTML in the following four stages:

- Level-0 included only the basic structural elements and assured that all browsers supported all features.
- Level-1 advanced features included highlighted text and graphics that were supported depending on the browser capability.
- Level –2 introduced the World Wide Web as an interactive medium and the feature of fill out forms on the Internet.
- Level-3 introduced frames, inline, video, sound, etc.

5.5.1 Importance of HTML

- HTML can be used to display any type of document on the host computer, which can be geographical at a different location.
- It is a versatile language and can be used on any platform or desktop.
- The appearance of a Web page is important, and HTML provides tags to make the document look attractive. Using graphics, fonts, different sizes, color, etc. can enhance the presentation of the document.
- HTML is easy language to program. This helps programmers to write the code for their websites without any error and without any mistakes.
- Dynamic HTML, also referred as "DHTML" is basically a collection of various technologies. We can create only static web-pages using HTML, But DHTML can be used for creating dynamic web pages. DHTML is a collection of various technologies which helps us to create a responsive and interactive websites with animations and effects.

5.5.2 Functionality of HTML in the project

As we know this is purely web-based project. This helps to embed PHP Pages with the page using some simple tags.

- Used to design the forms.
- Admin can communicate easily with server.

CHAPTER-6

DESIGN

System engineering and analysis encompasses requirements gathering at the system level with a small amount of top level analysis design. Information engineering encompasses requirements at the strategic business level and at the business area level.

6.1 UML Introduction

The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

- UML is specifically constructed through two different domains they are:
- UML Analysis modelling, this focuses on the user model and structural model views of the system.
- UML design modelling, which focuses on the behavioural modelling, implementation
- modelling and environmental model views.

6.2 Why Use UML in Projects?

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance.

Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modelling

Language(UML) was designed to respond to these needs. Simply, Systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be does easily through UML diagrams.

In this project two basic UML diagrams have been explained

1. Class Diagrams

2. Use Case Diagrams
3. Sequence Diagrams
4. Activity Diagrams
5. Collaboration Diagrams
6. Deployment Diagrams
7. State Chart Diagrams
8. Component Diagrams

Class Diagrams

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. This is one of the most important of the diagrams in development. The diagram breaks the class into three layers. One has the name, the second describes its attributes and the third its methods.

A padlock to left of the name represents the private attributes. The relationships are drawn between the classes. Developers use the Class Diagram to develop the classes. Analysts use it to show the details of the system.

Architects look at class diagrams to see if any class has too many functions and see if they are required to be split.

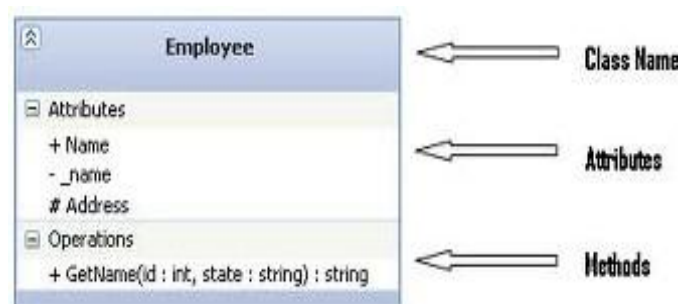


Figure 22. Class Structure

Use Case Diagrams

In software engineering, a use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between

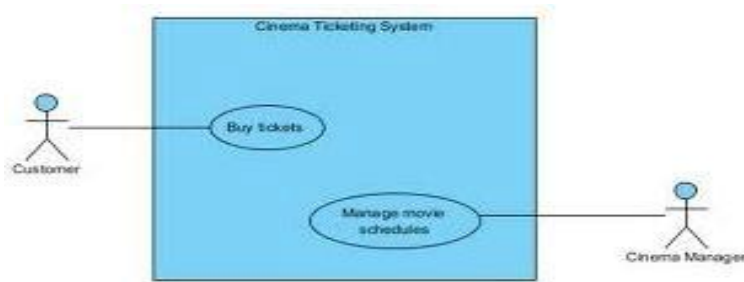


Figure 23. Actors and Use cases

Sequence Diagrams

—



Figure 24. Objects and Timestamps

Activity Diagrams

0 0 0 0 0 0

and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

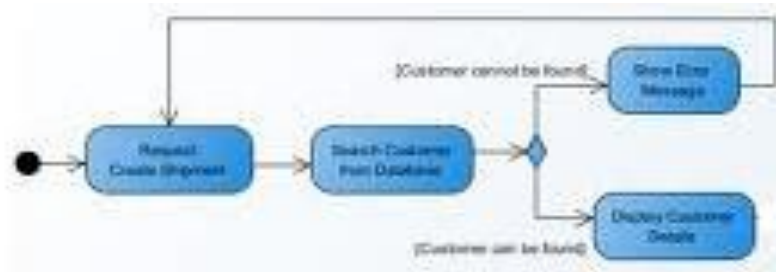


Figure 25. Activities

Collaboration Diagrams

A Communication diagram models the interactions between objects or parts in terms of sequenced messages. Communication diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.

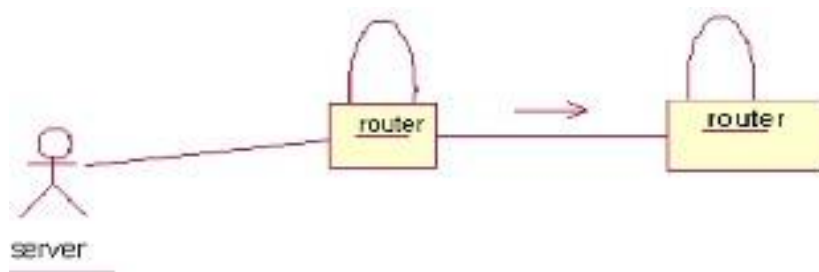


Figure 26. Collaborations

Deployment Diagrams

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

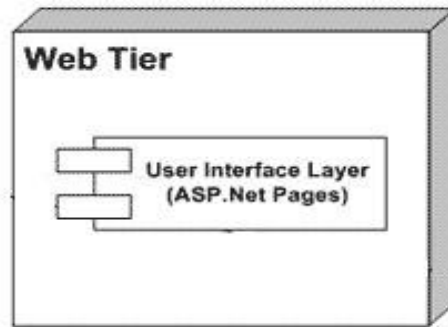


Figure 27. Deployment

State Chart Diagrams

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

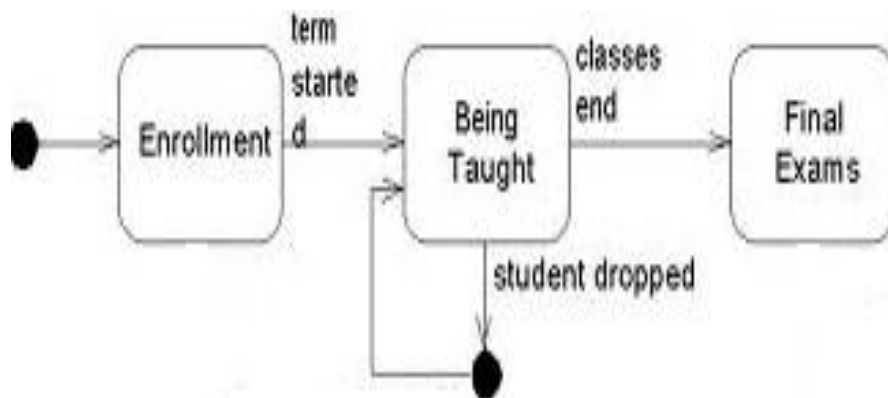


Figure 28. States and Relationships

Component Diagrams

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.



Figure 29. Components

These are different diagrams are available in a UML. Each diagram will be used for specific purpose. All diagrams programmers mainly used class diagram in order to generate code this kind of technique is called as forward engineering. That means here we convert model into code. If you convert code into model that kind of process is called as backward engineering.

6.3 Diagrams

6.3.1 UML Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components, a Meta-model and a Notation. In the future, some form of method or process may also be added to or associated with, UML.

The Unified Modelling Language is a standard language for Specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS

The primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

6.3.1.1 Activity Diagram

An Activity Diagram is essentially a flow chart showing flow of control from activity to activity. They are used to model the dynamic aspects of a system. They can also be used to model the flow of an object as it moves from state to state at different points in the flow of control.

An activity is an ongoing non-atomic execution within a state machine. Activities ultimately result in some action, which is made up of executable atomic computations that result in a change of state. This distinguishes a use case diagram from all other kinds of diagrams by its particular content.

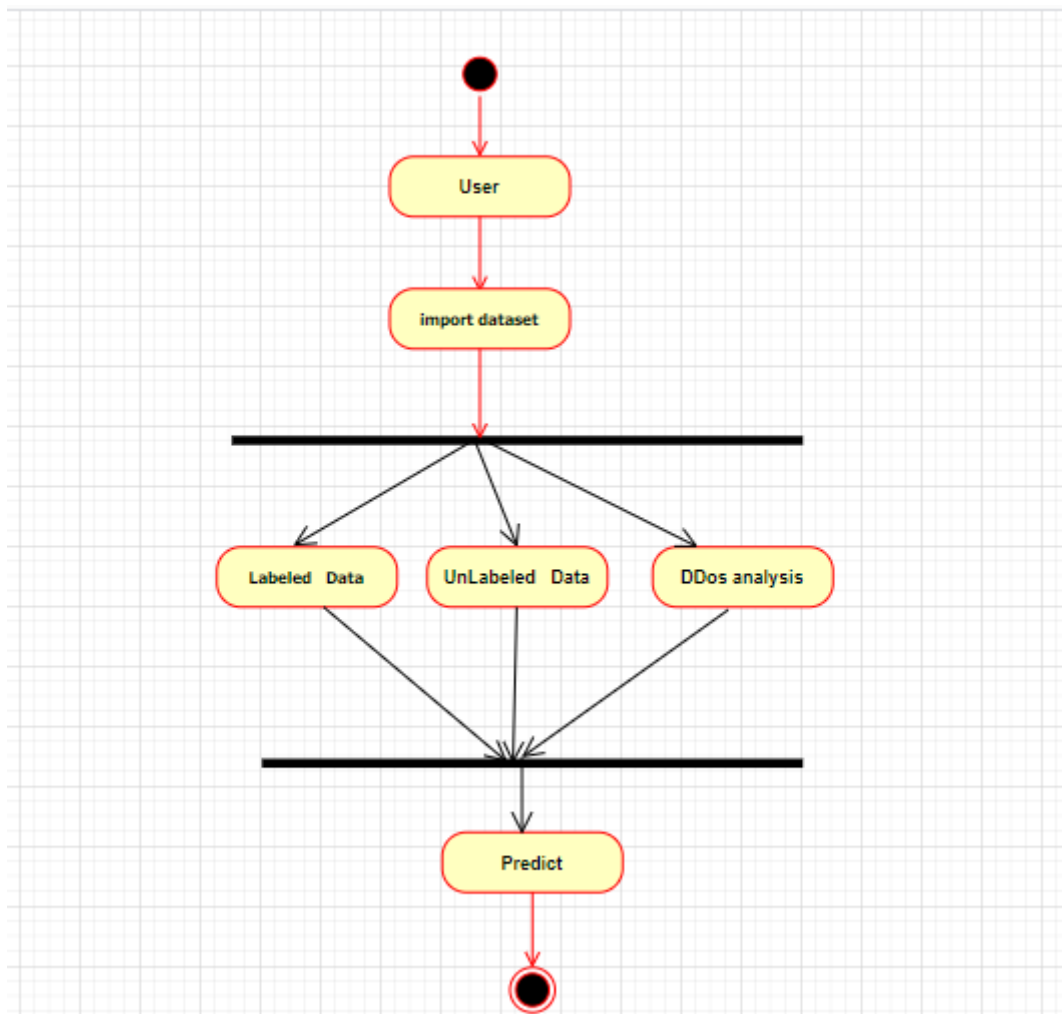


Figure 30. Activity Diagram

6.3.1.2. Use Case Diagram

Use Case Diagram represents the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behaviour of the system from an external point of view.

Use Case contains the following:

- The name of the use case is unique across the system so that developers (and project participants) can unambiguously refer to the use case.
- Participating actors are actors interacting with the use case.
- Entry Conditions describe the conditions that need to be stratified before the use case is initiated.
- The Flow of events describes the sequence of interactions of the use case, which are to be numbered for reference. The common case (i.e., cases that are expected by the user) and the exceptional cases (i.e., cases unexpected by the user, such as errors, and unusual conditions) are described separately in different use cases for clarity. We organize the steps in the flow of events in two columns, the left column representing steps accomplished by the actor, the right column representing steps accomplished by the system. Each pair of actor system steps represents an interaction.
- Exit conditions describe the conditions that are satisfied after the completion of the use case.
- Quality requirements are requirements that are not related to the functionality of the system, its implementation, and hardware platforms it runs on, and so on.

A UML use case diagram is the primary form of system/software requirements for a new software program under development. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

A use case diagram is usually simple. It does not show the detail of the use cases:

- It only summarizes **some of the relationships** between use cases, actors, and systems.
- It does **not show the order** in which steps are performed to achieve the goals of each use case.

As said, a use case diagram should be simple and contains only a few shapes. If yours contain more than 20 use cases, you are probably misusing use case diagram.

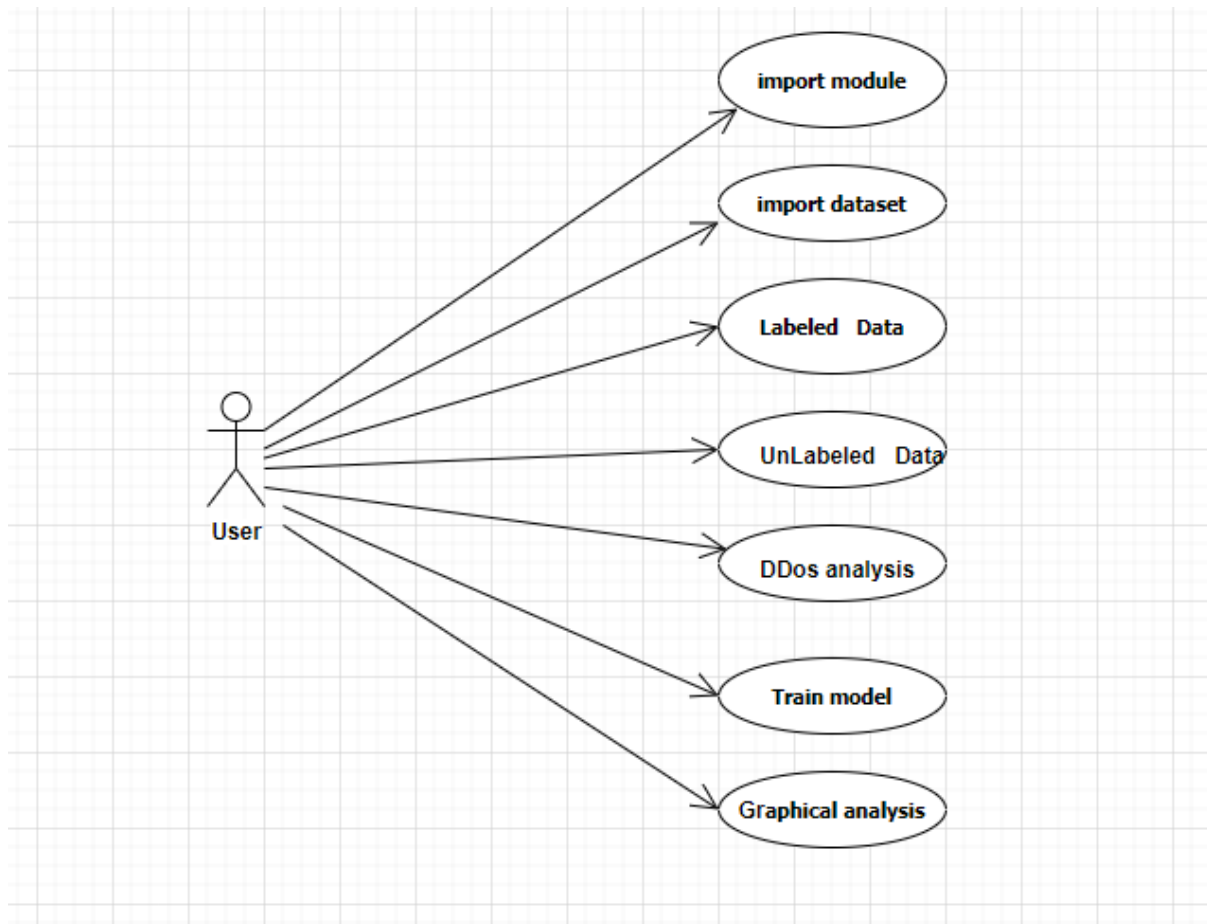


Figure 31. Use Case Diagram

6.3.1.3 Sequence diagram:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure.

These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen

and software developers to document and understand requirements for new and existing systems.

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

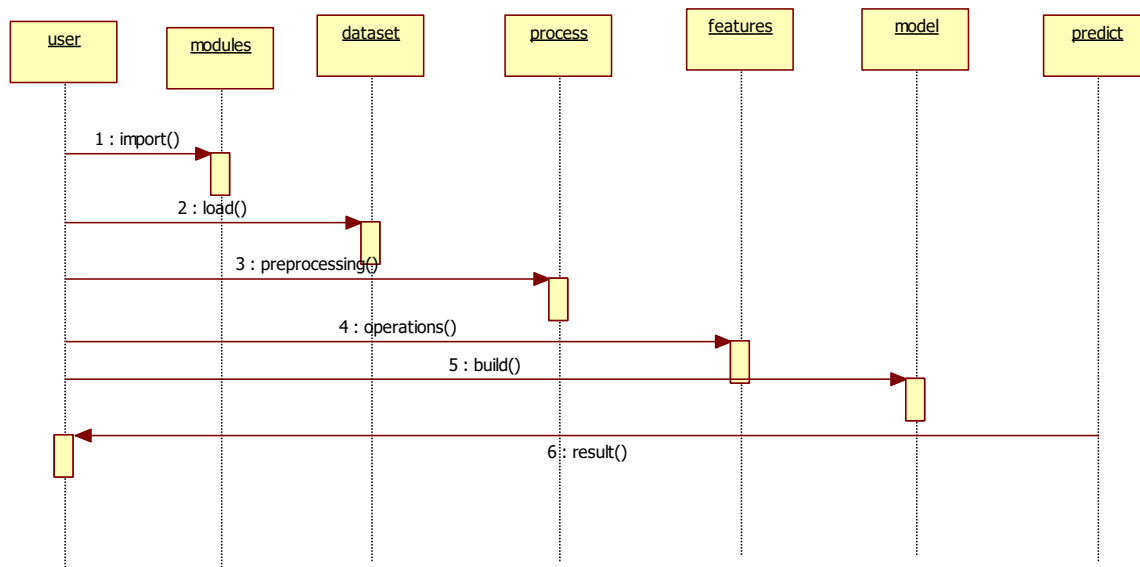


Figure 32.Sequence Diagram

CHAPTER 7

MODEL

7.1 What is SDLC?

SDLC stands for Software Development Life Cycle. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software. SDLC is the process consisting of a series of planned activities to develop or alter the software products.

Benefits of the SDLC Process

The intent of a SDLC process is to help produce a product that is cost-efficient, effective, and of high quality. Once an application is created, the SDLC maps the proper deployment and decommissioning of the software once it becomes a legacy. The SDLC methodology usually contains the following stages: Analysis (requirements and design), construction, testing, release, and maintenance (response). Veracode makes it possible to integrate automated security testing into the SDLC process through use of its cloud based platform.

1.Requirements Gathering:

In this phase we gather all the requirements from the client, i.e. what are the client expected input, output.....

2. Analysis:

In this phase based upon the client requirements we prepare one documentation is called “High Level Design Document”. It contains Abstract, Functional Requirements, Non Functional Requirements, Existing System, Proposed System, SRS,.....

3. Design:

It is difficult to understand the High Level Design Document for all the members, so to understand easily we use “Low Level Design Document”. To design this document we use UML (Unified Modelling Language). In this we have Use case, Sequence, Collaboration.....

4. Coding:

In this phase we develop the coding module by module. After developing all the modules we integrate them.

5. Testing:

After developing we have to check whether client requirements are satisfied or not. If not we are again going to develop.

6. Implementation:

In testing phase if client requirements are satisfied, we go for implementation. i.e. we need to deploy the application in some server.

7. Maintenance:

After deployment, if at all any problems come from the client side; we are providing maintenance for that application.

CHAPTER-8

IMPLEMENTATION

There are three modules can be divided here for this project they are listed as below

- User Apps
- DDOS Attack Deduction
- Classifications of DDOS attack
- Graphical analysis

From the above four modules, project is implemented. Bag of discriminative words are achieved.

1. User Apps

User handling for some various times of smart phones ,desktops laptops and tablets .If any kind of devices attacks for some unauthorized Malwaresoftwares.In this Malwareon threats for user personal dates includes for personal contact, bank account numbers and any kind of personal documents are hacking in possible.

2. DDOS Attack Deduction

User search the any link Notably, not all network traffic data generated by malicious apps correspond to malicious traffic. Many malwaretake the form of repackaged benign apps; thus, Malwarecan also contain the basic functions of a benign app.Subsequently, the network traffic they generate can be characterized by mixed benign and malicious network traffic.We examine the traffic flow header using Co-clustering algorithm from the natural language processing (NLP).

3.Classifications of DDOS Attack:

Here, we compare the classification performance of Co-clustering algorithm with other popular machine learning algorithms. We have selected several popular classification algorithms. For all algorithms, we attempt to use multiple sets of parameters to maximize the performance of each algorithm. Using Co-clustering algorithm algorithms classification for malware bag-of-words weight age.

4. Graphical analysis

The graph analysis is done by the values taken from the result analysis part and it can be analyzed by the graphical representations. Such as pie chart, pyramid chart and funnel chart here in this project.

CHAPTER-9

TESTING

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

9.1 TYPES OF TESTS

9.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.1.2 Integration testing

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

9.1.3 Functional test

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

Functional testing is centred on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

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

9.1.4 System Test

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

9.1.5 White Box Testing

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

9.1.6 Black Box Testing

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

9.1.7 Test strategy and approach

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

Test objectives

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

Features to be tested

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

Integration Testing

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

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

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER-10

EXECUTION AND RESULTS

10.1 Introduction

Executing consists of the processes used to complete the work defined in the project plan to accomplish the project's requirements. Execution process involves coordinating people and resources, as well as integrating and performing the activities of the project in accordance with the project plan.

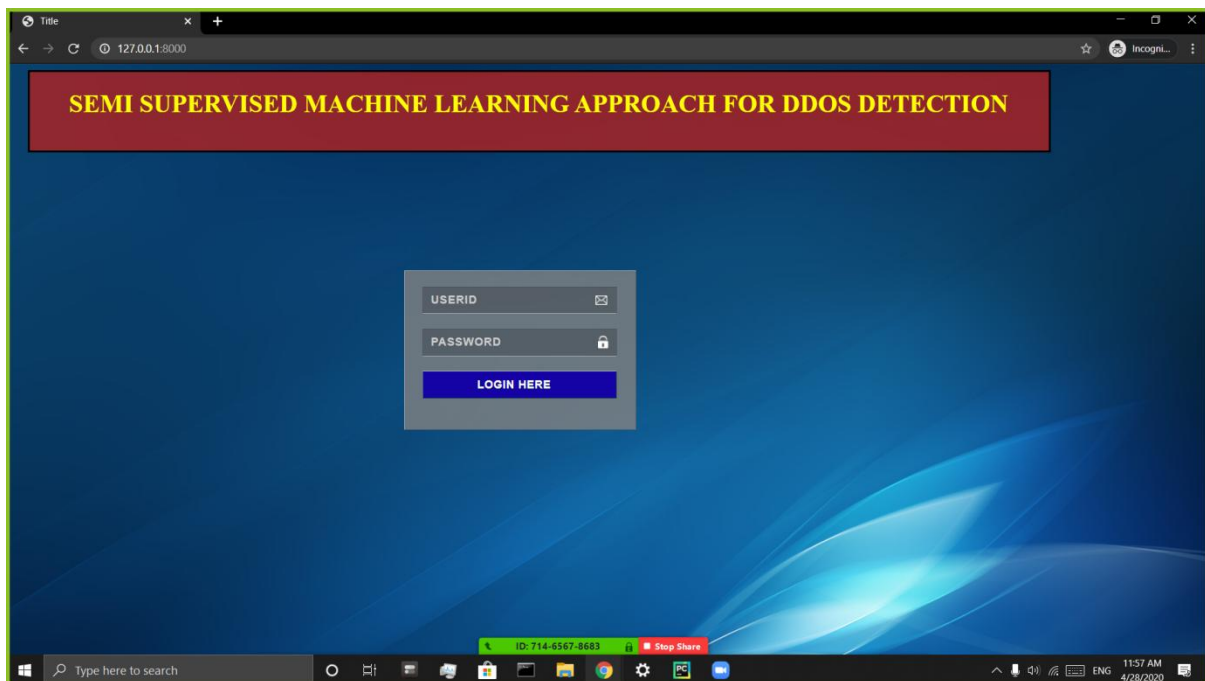
Command to run the project is:

“Python manage.py runserver”

10.2 output screens

Following are the screenshots that will display after execution of our project Source Code:

Screen1:Login Page



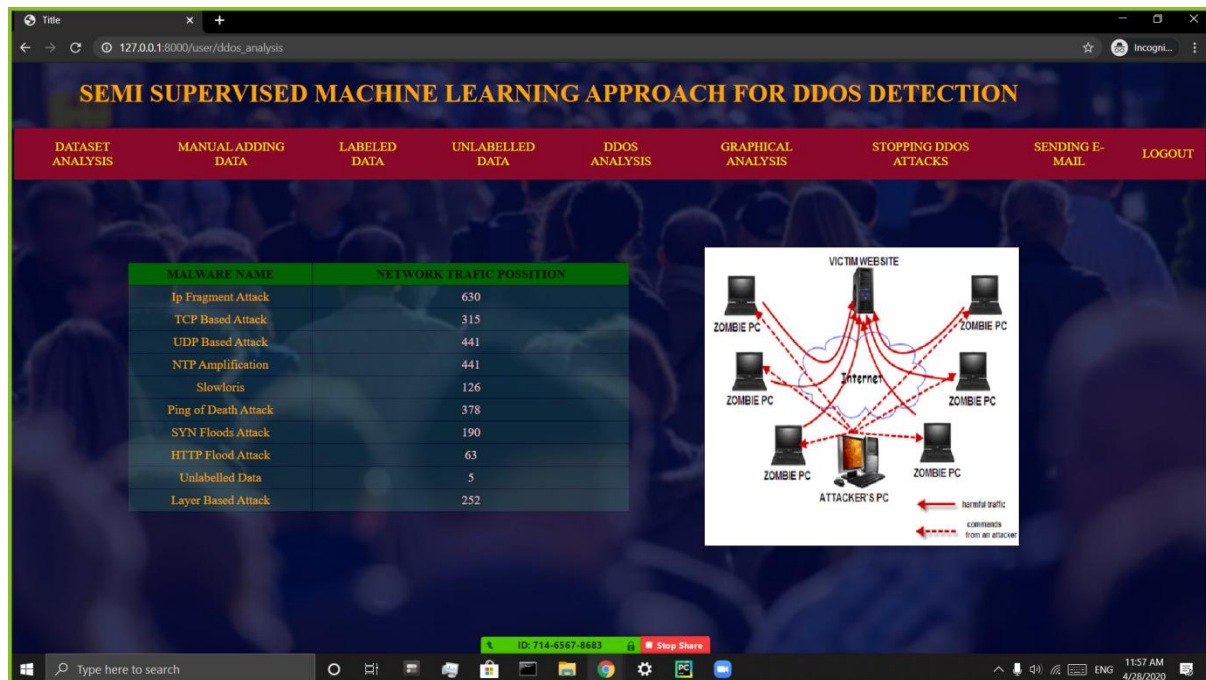
Screen 4: Unlabeled Data



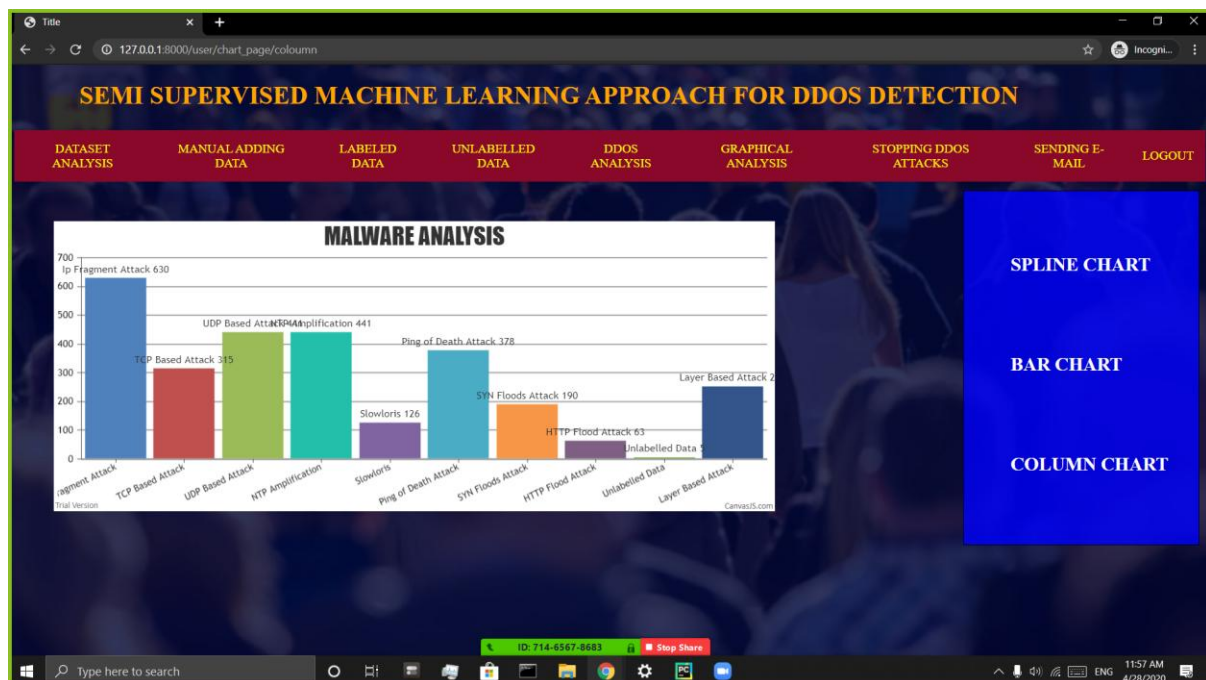
Screen 5: Manual adding Data



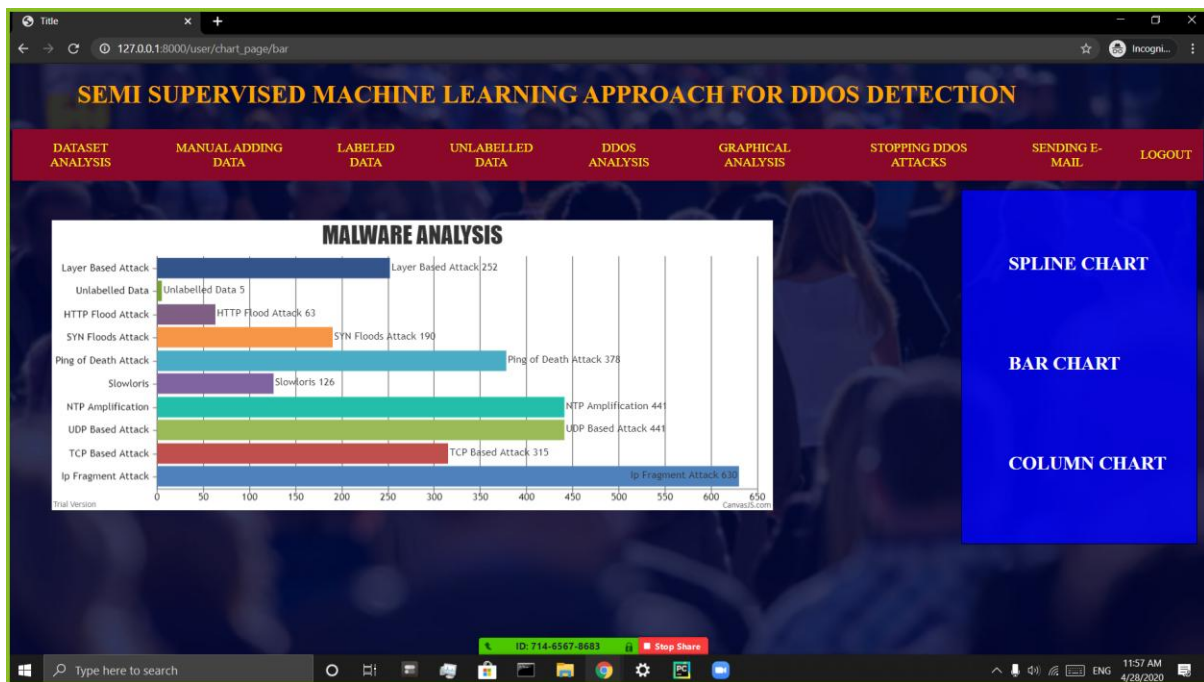
Screen 6:DDoS analysis



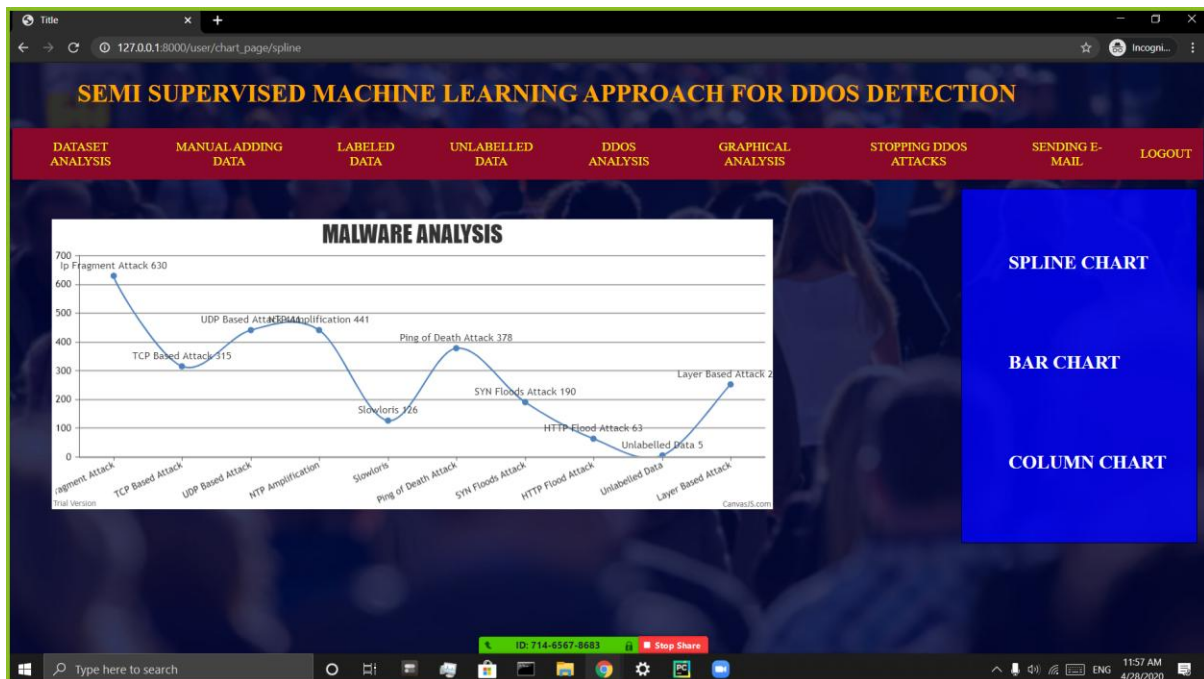
Screen 7:Graphical analysis--Column chart



Screen 8: Graphical analysis--Bar Chart



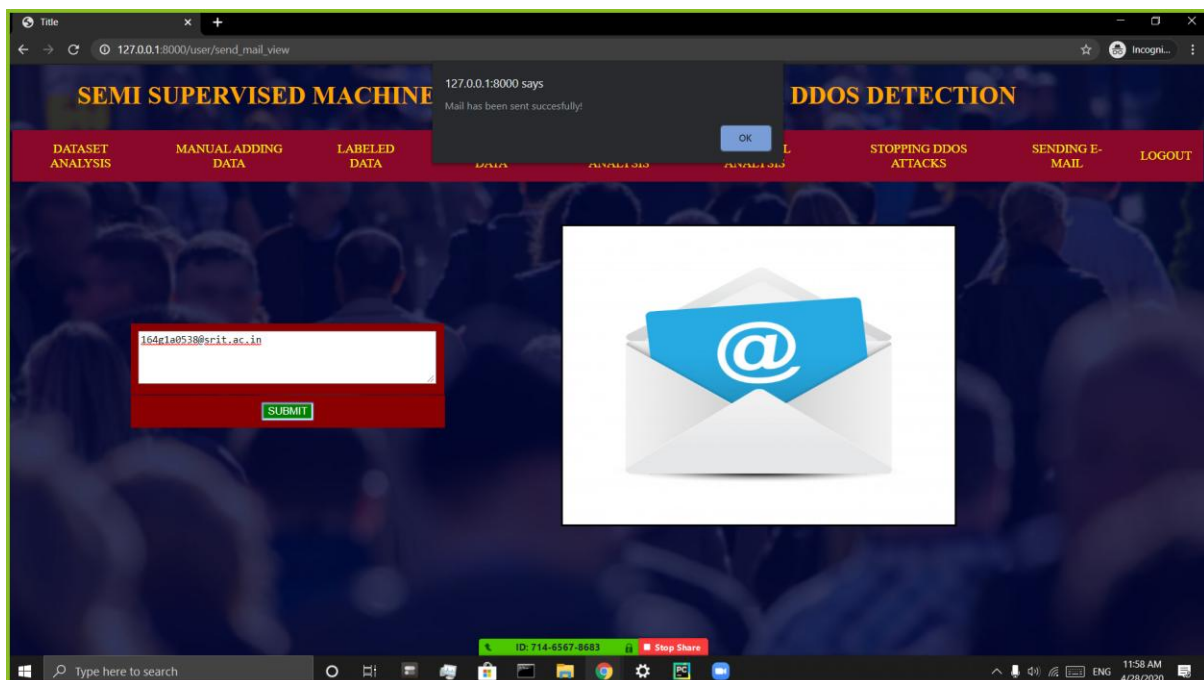
Screen 9: Graphical analysis--Spline Chart



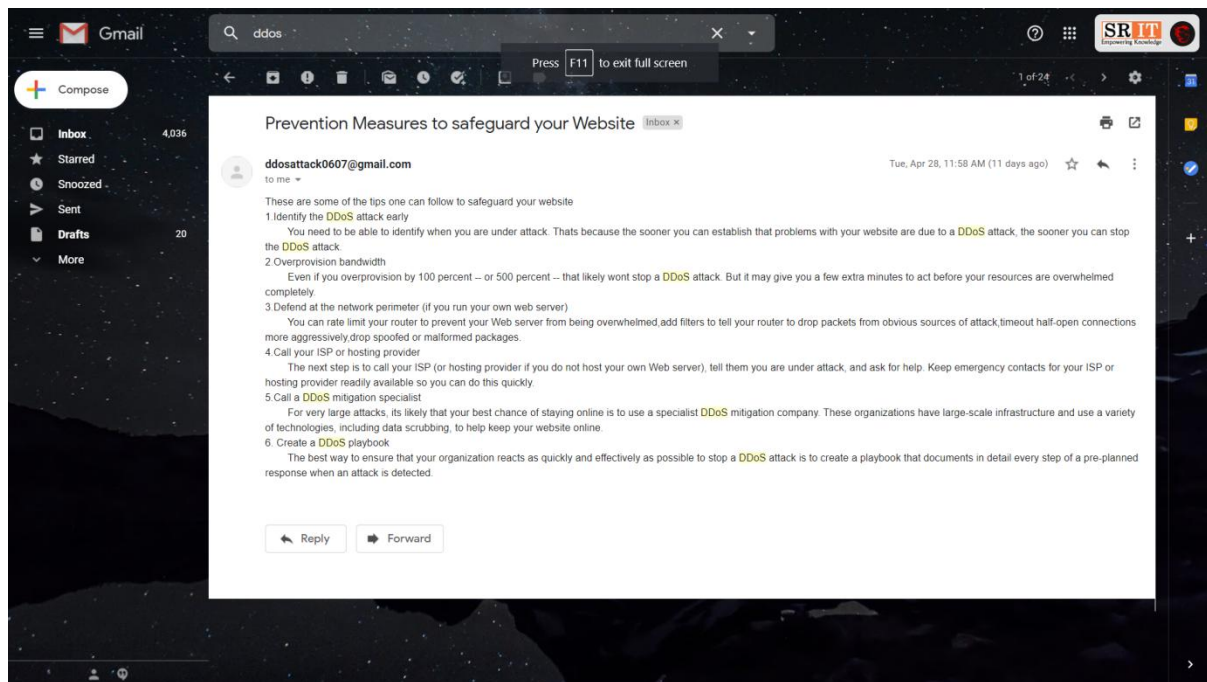
Screen 10: Stopping DDoS attacks



Screen 11: Sending E-Mails



Screen 12:E-Mail



CONCLUSION

However, today Internet security is a quickly growing concern. The prevalence of Internet attacks has increased significantly, but still the challenge of detecting such attacks generally falls on the end hosts and service providers, requiring system administrators to detect and block attacks on their own.

Distributed Denial of Service(DDoS) attack is a very serious problem of web applications. Finding the efficient solution of this problem is essential. Researchers have developed many techniques to detect and prevent this vulnerability.

REFERENCES

1. <https://www.cs.ucr.edu/~epapalex/papers/ASONAM2012.pdf>
2. <https://www.semanticscholar.org/paper/Semi-supervised-machine-learning-approach-for-DDoS-Idhammad-Afdel/cb2952189fbd6c9f003ae23bec7d7a674b598d1a>
3. <https://link.springer.com/article/10.1007/s10489-018-1141-2>
4. <https://www.semanticscholar.org/paper/Network-Anomaly-Detection-Using-Co-clustering-Papalexakis-Beutel/f0d1ab2e72e74ffe21e3a81c695d56c7b888882d>
5. <https://www.semanticscholar.org/paper/Network-Anomaly-Detection-Using-Co-clustering-Papalexakis-Beutel/dd9cd568de5af206b05b6d58c752b4240120d1b7>
6. https://www.researchgate.net/publication/251976778_DDoS_attack_detection_based_on_neural_network
7. <https://ieeexplore.ieee.org/document/5670479>