

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JNANA SANGAMA”, BELAGAVI, KARNATAKA– 560018,



Project Report

On

**“A DATA ANALYTICS APPROACH TO THE CYBERCRIME
UNDERGROUND ECONOMY”**

Submitted in partial fulfillment of the requirement of
Project Work [15CSP85]

In

COMPUTER SCIENCE & ENGINEERING

By

MAHADEVU M V [4MU16CS030]

SHIVA PRASAD N [4MU16CS058]

YASHAS P [4MU16CS081]

PRAJWAL H S [4MU16CS088]

Under the guidance of

SOUMYA B

Asst. Professor & Head,
Department of CSE



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MYSURU ROYAL INSTITUTE OF TECHNOLOGY

Lakshmipura Road, Palahally Post, SR Patna, Mandya – 571606

Academic Year: 2019 – 2020

MYSURU ROYAL INSTITUTE OF TECHNOLOGY

Lakshmipura Road, Palahally Post, SR Patna, Mandya – 571606

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that the project work entitled “**A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY**” is a bonafide work carried out by **Mahadevu M V [4MU16CS030]** , **Shiva Prasad N [4MU16CS058]** , **Yashas P [4MU16CS081]** and **Prajwal H S [4MU16CS088]** , students of VIII Semester B.E at, **Mysuru Royal Institute of Technology, Mandya**, in partial fulfillment of the requirement of project work of the **Visvesvaraya Technological University, Belagavi** during the academic year **2019–2020**. It is certified that all corrections/suggestions indicated during Internal Assessment have been incorporated in the report submitted to the department. The report has been approved as it satisfies the academic requirements in respect of the prescribed by the university.

.....
Signature of the Guide
SOUMYA B,
Asst. Professor & Head,
Department of
Computer Science & Engineering,
Mysuru Royal Institute of
Technology, Mandya

.....
Signature of the HOD
SOUMYA B,
HOD,
Department of
Computer Science & Engineering,
Mysuru Royal Institute of
Technology, Mandya

.....
Signature of the Principal
Dr. H.S SURESH CHANDRA,
Principal,
Mysuru Royal Institute of
Technology, Mandya

External Viva-Voce

Name of the Examiner

Signature with Date

- 1.
- 2.

MYSURU ROYAL INSTITUTE OF TECHNOLOGY, MANDYA
Department of Computer Science and Engineering



DECLARATION

We Mahadevu M V (4MU16CS030), Shiva Prasad N (4MU16CS058), Yashas P (4MU16CS081) & Prajwal H S (4MU16CS088) studying in the 8TH semester BE, Computer Science and Engineering, MRIT, Mandya, hereby declare that the project work entitled “A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY” has been carried out by me independently under the guidance of **Prof. Soumya B**, Assistant Professor & Head, Department of Computer Science and Engineering, MRIT, Mandya. This project work is submitted to the **Visvesvaraya Technological University, Belagavi** in the partial fulfillment of required for the award of degree in **BE of Computer Science and Engineering** during the academic year **2019-2020**.

This dissertation has not been submitted previously for the award of any other degree or diploma to any other institution or university.

Date: / 08 / 2020

Place: Mandya

.....(sign).....

Mahadevu M V (4MU16CS030)

Shiva Prasad N (4MU16CS058)

Yashas P (4MU16CS081)

Prajwal H S (4MU16CS088)

ACKNOWLEDGEMENT

Happiness cannot be expressed by words and help taken cannot be left without thanking. I would like to thank all of them who were a part of my life and my work.

We are thankful to our principal, **Dr. Suresh Chandra H S** for all the facilities provided to us in the college.

We would like to convey my sincere thanks to **Prof. Soumya B**, Head of the department, Computer Science and Engineering Department, MRIT.

We especially thankful to **Prof. Chethan Raj C, Associate Professor & Project Coordinator, Prof. Harish Boriah, Associate Professor & Project Coordinator, Dept. of Computer Science and Engineering**, MRIT, for support, guidance, motivation, useful tips, whole hearted encouragement and individual guidance and timely suggestions for carrying out and the successful completion of this project.

We express my deep profound gratitude to **Prof. Soumya B**, Assistant professor & Head, Dept. of CS&E, MRIT who has been my guide and guiding light in my endeavor & motivation & encouragement for the successful completion of this project successfully.

We would specially thank our all faculty members and non-teaching staffs for their valuable suggestions and encouragement.

Thanking You

MAHADEVU M V (4MU16CS030)

SHIVA PRASAD N (4MU16CS058)

YASHAS P (4MU16CS081)

PRAJWAL H S (4MU16CS088)

DECLARATION

We **Mahadevu M V, Shiva Prasad N, Yashas P & Prajwal H S**, hereby declare that this dissertation work titled **“A Data Analytics Approach to the Cyber Crime Underground Economy”** has been carried out independently by us under the guidance of **Soumya B**, Assistant Professor & Head, Department of Computer Science and Engineering, MRIT, Mandya in partial fulfillment of the requirement of the degree BACHELOR OF ENGINEERING in Computer Science and Engineering under VTU Belagavi.

We further declare that we have not submitted this dissertation either in part or full to any other university for the award of any degree.

Place: Mandya

Mahadevu M V (4MU16CS030)

Shiva Prasad N (4MU16CS058)

Yashas P (4MU16CS081)

Prajwal H S (4MU16CS088)

ABSTRACT

Despite the rapid escalation of cyber threats, there has still been little research into the foundations of the subject or methodologies that could serve to guide Information Systems researchers and practitioners who deal with cyber security. In addition, little is known about Crime-as-a-Service (CaaS), a criminal business model that underpins the cybercrime underground. This research gap and the practical cybercrime problems we face have motivated us to investigate the cybercrime underground economy by taking a data analytics approach from a design science perspective.

To achieve this goal, we propose (1) a data analysis framework for analyzing the cybercrime underground, (2) CaaS and crime ware definitions, and (3) an associated classification model. In addition, we (4) develop an example application to demonstrate how the proposed framework and classification model could be implemented in practice. We then use this application to investigate the cybercrime underground economy by analyzing a large dataset obtained from the online hacking community. By taking a design science research approach, this study contributes to the design artifacts, foundations, and methodologies in this area.

Moreover, it provides useful practical insights to practitioners by suggesting guidelines as to how governments and organizations in all industries can prepare for attacks by the cybercrime underground market where illegal services are provided to help underground buyers conduct cybercrimes, such as attacks, infections, and money laundering in an automated manner”. Thus, CaaS is referred to as a do-it-for-me service, unlike crimeware which is a do-it-yourself product. Because CaaS is designed for novices, its customers do not need to run a hacking server or have high-level hacking skills.

Table of Contents

CONTENTS	PAGE NO.
ACKNOWLEDGEMENT	i
DECLARATION	ii
ABSTRACT	iii
LIST OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF TABLES	vii
LIST OF SNAPSHOTS	viii

TABLE OF CONTENTS

Chapter 1 Introduction	01
1.1 Domain Overview	01
1.2 Project Overview	01
1.3 Existing System	02
1.4 Drawback of Existing System.....	02
1.5 Problem Statement	03
1.6 Project Scope & Motivation.....	04
1.7 Proposed System.....	04
1.8 Advantages of Proposed System.....	05
1.9 Objective of the Project	06
1.8 Organization of Report	06
Chapter 2 Literature Survey.....	07
2.1 Literature Review.....	07
2.2 Conclusion of Review	09
Chapter 3 Software Requirement Specification.....	10
3.1 Introduction.....	10
3.2 Functional Requirements	11
3.3 Non-Functional Requirements	11

3.4 System Requirements.....	13
3.4.1 Hardware Requirements	13
3.4.2 Software Requirements	13
Chapter 4 System Design	15
4.1 Introduction to System Design	15
4.2 High Level Design	15
4.2.1 System Architecture	16
4.2.2 Data Flow Diagram	17
4.3 Low Level Design	18
4.3.1 Flow Chart	18
4.3.2 Sequence Diagram	19
4.3.3 Use Case Diagram	20
4.3.4 Activity Diagram	21
Chapter 5 System Development.....	22
5.1 Introduction to System Development	22
5.2 Module & Methodology	23
5.2.1 Sub Module1	23
5.2.2 Sub Module2.....	24
Chapter 6 Implementation.....	25
6.1 Introduction to System Implementation.....	25
6.2 Language Used for Implementation.....	25
6.3 Algorithm.....	26
6.4 Snapshot of Code Snippet.....	27
Chapter 7 Testing.....	29
7.1 Introduction to Testing	29
7.1 Testing Types.....	29
7.2 Test Cases	30
Chapter 8 Result Analysis.....	32
8.1 Snapshots with Description.....	32
Conclusion	36
Future Work	37
References.....	38

LIST OF FIGURES

Figure No	Description	Page No
Fig 4.2.1	System Architecture of Proposed model	16
Fig 4.2.1(i)	Data Flow Diagram for admin model	17
Fig 4.2.1(ii)	Data Flow Diagram for user	17
Fig 4.3.1	Flow Chart	18
Fig 4.3.2(i)	Sequence Diagram for User	19
Fig 4.3.3(ii)	Sequence Diagram for Admin	19
Fig 4.3.3(i)	Use Case Diagram for Entity/Admin	20
Fig 4.3.3(ii)	Use Case Diagram for Entity/User	20
Fig 4.3.4	Activity Diagram	21
Fig 5.1	Implementing System Development	22
Fig 5.2.1	System Design Modules	23
Fig 5.2.2	Modules for Implementation	24
Fig 6.2.2	Django Web Framework	26

LIST OF TABLES

Table No	Description	Page No
Table 2.1	Literature Review	08-09
Table 3.4.2	Software Requirement Specification(SRS) table	14
Table 7.2	Test Cases	31

LIST OF SNAPSHOT

Snapshot No	Description	Page No
Snapshot 1.5	Problem faced by Companies due to Cubercrime	03
Snapshot 6.4.1	urls.py(code snippet)	27
Snapshot 6.4.2	models.py(code snippet)	28
Snapshot 6.4.3	views.py(code snippet)	28
Snapshot 8.1.1	Login Page	32
Snapshot 8.1.2	Registration Page	32
Snapshot 8.1.3	Upload File Page	33
Snapshot 8.1.4	View File Uploaded Page	33
Snapshot 8.1.5	Download Page	33
Snapshot 8.1.6	Admin view upload Page	34
Snapshot 8.1.7	Admin view request Page	34
Snapshot 8.1.8	Feedback Page	34
Snapshot 8.1.9	Positive Conversation Page	35
Snapshot 8.1.10	Negative Conversation Page	35
Snapshot 8.1.11	CSV File in Exel Conversation	35
Snapshot 8.1.12	Data Analysis Graph	36

CHAPTER 1

INTRODUCTION

1.1 Domain Overview

This project collected data from the largest hacking community and did not consider other hacking communities. Future studies will therefore need to generalize our findings by investigating a wider range of hacking communities. Second, this study has focused on the CaaS and crimeware available in the cybercrime underground, but much in-depth analysis remains to be done on the configurations of cybercrime networks. In this project we use Machine Learning as domain as A Data Analytics approach to analyze the Cybercrime and its Community. We use Data-Analytics Domain with the use of Machine Learning Naïve bayes algorithm approach for predicting a large set of data. We use Python as Coding language with a Django web frameworks for Domain Overview.

1.2 Project Overview

As the threat posed by massive cyberattacks (e.g., ransomware and distributed denial of service attacks (DDoS)) and cybercrimes has grown, individuals, organizations, and governments have struggled to find ways to defend against them. In 2017, ransomware known as WannaCry was responsible for nearly 45,000 attacks in almost 100 countries. The explosive impact of cybercrime has put governments under pressure to increase their cybersecurity budgets. United States President Barack Obama proposed spending over \$19 billion on cybersecurity as part of his fiscal year 2017 budget, an increase of more than 35% since 2016. The cybercrime underground has thus emerged as a new type of organization that both operates black markets and enables cybercrime conspiracies to flourish. In contrast, cybercrime networks are lateral, diffuse, fluid, and evolving the threat posed by the rise of highly professional network-based cybercrime business models, such as Crimeware-as-a-Service (CaaS), remains mostly invisible to governments, organizations, and individuals. Despite the rapid escalation of cyber threats, there has still been little research into the foundations of the subject or methodologies that could serve to guide Information Systems researchers and practitioners who deal with cyber security. In addition, little is known about Crime-as-a-Service (CaaS), a criminal business model that underpins the cybercrime underground.

1.3 Existing System

Cybercrime has undergone a revolutionary change, going from being product-oriented to service-oriented because the fact it operates in the virtual world, with different spatial and temporal constraints, differentiates it from other crime taking place in the physical world. As part of this change, the cybercrime underground has emerged as a secret cybercrime marketplace because emerging technological changes have provided organized cybercriminal groups with unprecedented opportunities for exploitation. The cybercrime underground has a highly professional business model that supports its own underground economy. This business model, known as CaaS, is “a business model used in the underground market where illegal services are provided to help underground buyers conduct cybercrimes, such as attacks, infections, and money laundering in an automated manner,”. Thus, CaaS is referred to as a do-it-for-me service, unlike crimeware which is a do-it-yourself product. Because CaaS is designed for novices, its customers do not need to run a hacking server or have high-level hacking skills. Consequently, the CaaS business model can involve the following roles: writing a hacking program, performing an attack, commissioning an attack, providing an attack server (infrastructure), and laundering the proceeds.

Sood and Enbody have suggested that crimeware marketplaces have three key elements, namely actors (e.g., coders, operators, or buyers), value chains, and modes of operation (e.g., CaaS, pay-per-install, crimeware toolkits, brokerage, or supplying data). Periodic monitoring and analysis of the content of cybercrime marketplaces could help predict future cyber threats.

1.4 Drawbacks of Existing System

- A previous study proposed a data mining framework for crime, dividing crimes harmful to the general public into eight categories:
 1. Traffic violations.
 2. Sex Crime.
 3. Theft.
 4. Fraud.
 5. Gang/Drug Offenses.
 6. Arson.
 7. Violent crime.
 8. Cyber-crime.
- Although this previous study explained how data mining techniques could be applied to crime analysis, it did not consider the specific features of cyber-crime.

- The “absence of capable guardians against crime” is due to organizations failing to take preventive measures against cybercrime.
- Periodic monitoring and analysis of the content of cybercrime marketplaces could help predict future cyber threats.
- Crimeware marketplaces have three key elements, namely actors (e.g., coders, operators, or buyers), value chains, and modes of operation (e.g., CaaS, pay-per-install, crimeware toolkits, brokerage, or supplying data).
- Consequently, the CaaS business model can involve the following roles:
 - i. writing a hacking program
 - ii. performing an attack
 - iii. commissioning an attack
 - iv. providing an attack server (infrastructure) and laundering the proceeds.

1.5 Problem Statement

These proposed sets of definitions for different types of CaaS the most common account hacking methods are phishing and brute force attacks. With an emphasis on selling this as a service, we define an account hacking service as a service that offers to gain unauthorized access to a target’s account by obtaining account information (e.g., username and password) or extra security information (e.g., security questions and answers).

Industry	Company
Technology (e.g., software, automobiles)	3M, Adobe, BMW, CISCO, EA, Exino Inc., GE, GlobalScape, HP, IDC Research Inc., Intel Corp., KDDI Japan, LG, Microsoft, Oracle, Panasonic, Panda Security, Philips, Samsung, Scania, Simba, Softbank Korea, Sony, Sybase, Sycore Business Solutions Corp., SynLan Technologies, Western Digital, Yamaha
Content (e.g., social network services, Internet, news)	ABC, AOL, Baidu, Bang Bros, CBS, Craigslist, Facebook, Google, IMDB, Instagram, Justin.tv, Last.FM, LinkedIn, LiveJournal, MSN, NBC, SoundCloud, Twitter, Warner Bros, Yahoo, YouTube, Zynga
Finance (e.g., banking, investing, and payments)	AlertPay, American Express, AMP, Personal Banking, Bank of America, Blackrock, Canadian Bank, Clickbank, Digital River, Goldman Sachs, iBank, Indian Bank, IP Capital, Kidd, Liberty Reserve, Moneybookers, PayPal, PlaySpan, Polish Bank, State Bank of India, Tradestation, Western Union
E-commerce (e.g., products and services)	Amazon, Best Buy, Dope, eBay, GameStop, GoDaddy, Groupon, Netflix, Nike, Staples, Uber, Walmart
Tele Comm. (e.g., smartphones and service providers)	Apple, AT&T, HTC, KT Freetel, MetroPCS, Nokia, Sprint, Swisscom, T-Mobile, Verizon Wireless
Others	Airsoft Gun, Ajanta Pharmaceuticals, ARMA International, FedEx, Green Leaf Technology, UPS, USG Corp.

Figure 1.5 Problem Faced by Companies due to Cyber crime

1.6 Project Scope & Motivation

To achieve this goal, we propose (1) a data analysis framework for analyzing the cybercrime underground, (2) CaaS and crime ware definitions, and (3) an associated classification model. In addition, we (4) develop an example application to demonstrate how the proposed framework and classification model could be implemented in practice.

We then use this application to investigate the cybercrime underground economy by analyzing a large dataset obtained from the online hacking community. By taking a design science research approach, this study contributes to the design artifacts, foundations, and methodologies in this area. Moreover, it provides useful practical insights to practitioners by suggesting guidelines as to how governments and organizations in all industries can prepare for attacks by the cybercrime underground.

1.7 Proposed System

The goal of our data analysis framework is to conduct a big-picture investigation of the cybercrime underground by covering all phases of data analysis from the beginning to the end. This framework comprises four steps: (1) defining goals; (2) identifying sources; (3) selecting analytical methods; and (4) implementing an application. Because this study emphasizes the importance of RAT for analyzing the cybercrime underground, the proposed RAT-based definitions are critical to this framework: **Steps 1–2** all contain the RAT elements.

A.Step 1: Defining Goals The first step is to identify the conceptual scope of the analysis. Specifically, this step identifies the analysis context, namely the objectives and goals. To gain an in-depth understanding of the current CaaS research, we investigated the cybercrime underground, which operates as a closed community. Thus, the goal of the proposed framework is to “investigate the cybercrime underground economy.”

B.Step 2: Identifying Sources the second step is to identify the data sources, based on the goals defined by Step 1. This step should consider what data is needed and where it can be obtained. Since the goal of this study is to investigate the cybercrime underground, we consider data on the cybercrime underground community. We therefore collected such data from the community itself and obtained a malware database from a leading global cyber security research firm. Because cybercriminals often change their IP addresses and use anti-crawling scripts to conceal their communications, we used a self-developed crawler that can resolve captcha’s and anti-crawling scripts to gather the necessary data. We

collected a total of 2,672,091 posts selling CaaS or crimeware, made between August 2008 and October 2017, from a large hacking community site (www.hackforums.net) with over 578,000 members and more than 40 million posts.

We also collected 16,172 user profiles of sellers and potential buyers, based on their communication histories, as well as prices and questions and answers about the transactions. The black market uses traditional forum threads (e.g., bulletin boards) instead of typical e-commerce platforms (e.g., eBay, and Amazon). For example, sellers create threads in marketplace forums to sell items, and potential buyers comment on these threads. One of the most significant challenges was therefore converting this unstructured data into structured data.

1.8 Advantages of Proposed System

- Compelling and relevant content will grab the attention of potential customers and increase brand visibility.
- You can respond instantly to industry developments and be seen as ‘thought leader’ or expert in your field.
- This can improve how your business is seen by your audience.
- Positive feedback is public and can be persuasive to other potential customers.
- Negative feedback highlights areas where you can improve.
- Data mining is a technique used to mine out patterns of useful data from large data sets.
- The goal of the proposed framework is to “investigate the cybercrime underground economy.” This Crimeware marketplaces have three key elements, namely
 - 1.actors (e.g., coders, operators, or buyers),
 - 2.value chains, and
 - 3.modes of operation (e.g., CaaS, pay-per-install, crimeware toolkits, brokerage, or supplying data).

1.9 Objective of the Project

This project collected data from the largest hacking community and did not consider other hacking communities. Future studies will therefore need to generalize our findings by investigating a wider range of hacking communities. Second, this study has focused on the CaaS and crimeware available in the cybercrime underground, but much in-depth analysis remains to be done on the configurations of cybercrime networks.

To achieve this goal, we propose (1) a data analysis framework for analyzing the cybercrime underground, (2) CaaS and crime ware definitions, and (3) an associated classification model. In addition, we (4) develop an example application to demonstrate how the proposed framework and classification model could be implemented in practice. We then use this application to investigate the cybercrime underground economy by analyzing a large dataset obtained from the online hacking community. By taking a design science research approach, this study contributes to the design artifacts, foundations, and methodologies in this area. Moreover, it provides useful practical insights to practitioners by suggesting guidelines as to how governments and organizations in all industries can prepare for attacks by the cybercrime underground.

1.10 Organization of Report

This chapter summarizes introduction of project which is elaborately described in later section of report, so the second chapter provides a detailed survey about this projected system which constitutes various paper related to how the performance of Python/Django can be improved. In the third chapter the requirements, constraints listed by the user for designing this application is described and following chapter illustrates design of the application which comprises of sequence diagram, architecture, and so on. Fifth chapter gives insight on implementation part and then testing techniques and test cases used for verifying this application is depicted in chapter six. Analysis of report containing snapshots is present in seventh chapter and finally conclusion and future enhancement is specified in the end along with a references or bibliography.

CHAPTER 2

LITERATURE SYRVEY

2.1 Literature Review

Cybercrime has undergone a revolutionary change, going from being product-oriented to service-oriented because the fact it operates in the virtual world, with different spatial and temporal constraints, differentiates it from other crime taking place in the physical world. As part of this change, the cybercrime underground has emerged as a secret cybercrime marketplace because emerging technological changes have provided organized cybercriminal groups with unprecedented opportunities for exploitation. The cybercrime underground has a highly professional business model that supports its own underground economy. This business model, known as CaaS, is “a business model used in the underground market where illegal services are provided to help underground buyers conduct cybercrimes, such as attacks, infections, and money laundering in an automated manner,”.

Thus, CaaS is referred to as a do-it-for-me service, unlike crimeware which is a do-it-yourself product. Because CaaS is designed for novices, its customers do not need to run a hacking server or have high-level hacking skills. Consequently, the CaaS business model can involve the following roles: writing a hacking program, performing an attack, commissioning an attack, providing an attack server (infrastructure), and laundering the proceeds.

According to this theory, three elements are necessary for crimes to be committed:

- (1) a likely offender,
- (2) a suitable target, and
- (3) the absence of capable guardians against crime.

In a cybercrime context, the “likely offenders” are motivated sellers and potential buyers in the underground market, and the “suitable targets” are the targeted vulnerable organizations. The “absence of capable guardians against crime” is due to organizations failing to take preventive measures against cybercrime. Two types of product or service are available in the cybercrime underground. The first can be either CaaS or crimeware that are related to attack strategy, for example, phishing, brute force, or DDoS attacks, or can be used for spamming or creating botnets, exploits, ransomware, rootkits, or Trojans. Attack

strategies often exploit system vulnerabilities such as application loopholes. In addition, social engineering attacks exploit human vulnerabilities . The most well-known example of such an attack is the use of a “secret question” for password recovery: attackers check into the user’s background to guess the secret question and hence steal the account. Examples include encryption and virtual private network (VPN) services, crypters, and proxies. From the perspective of RAT, the likely offenders are attackers motivated to attack organizations or products that constitute a suitable target. If such targets are attacked, however, both the targets and those who supply their cybersecurity products become aware of the vulnerabilities that made the attack possible, leading them to apply security updates to their software. These updates can be seen as capable guardians against crime, and the preventive measures taken can be identified by looking through each program’s version history. However, this is not the end of the matter, because the attackers will then develop and sell new versions of their hacking tools to combat the guardians, thus re-establishing the third RAT condition, the absence of capable guardians against crime. Such events can also be identified by the version numbers of the hacking tools sold on the black market: since it is an online marketplace, attackers must give detailed explanations to retain their customers’ confidence.

Sl no	Paper Title	Publication name & Year of Publication	Findings of Proposed System (Implemented)	Implementation, Algorithm and methodology used	Remarks (Not yet implemented)
1	Cybercrime underground economy	IEEE Transactions (2018)	Data analytics approach to the system	Naïve Bayes Algorithm MAP functions and methodology	Under Implementation
2	Cybercrime at a Scale	IEEE community (2017)	Deployment of HTTP based botnet commands	HTTP commands and control panels	Deployment Stage
3	Toward a better Measure of business proximity	MIS Quart, Conf. (2016)	Fact sheet Cybersecurity national Action plan	Modeling for Industry Intelligence and facts.	Implemented
4	Bypassing Antivirus	Proc.,18 th Panhellenic Conf.2014	Detection with Encryption	Antivirus bypassing and detection of Encryption	Conference

5	VPN	IEEE potentials (2001)	VPN and its activities	VPN and proxy network models	Implemented
6	CAAS-crimeware	IEEE (2013).	A survey of commoditized crimeware	Crimeware-as-a-service and Crimeware underground market	Conference
7	Cyberterrorism Today	Emerald group publications (2014).	Understand Cyberterrorism	Cyberterrorism and its access towards business	Publication
8	Hacking vulnerability	relevant result (July 6, 2011). BBC news (Feb. 24,2016).	Hacking vulnerability monitoring system.	Hacking vulnerability disclosed and the earlier signal from the underground	Published
9	Myths about malware	www.welivesecurity.com online platform	Malware and its Myths	Myths about malware and exploits	Uploaded
10	Trends in cybercrime	Justic,vol. 20,pp.,2005	The dark side of the Internet	Understanding the dark side of Internet.	Published

Table 2.1 Literature Review

2.2 Conclusion of Review

The data analysis step of the proposed framework involves four steps. Here, we report the data analysis results: CaaS and crimeware classification and market trends, cybercrime market dynamics, and potential hacking targets.

CaaS and Crimeware Classification and Market Trends

Here, we evaluate the accuracy of the proposed classifications. Specifically, we analyze the CaaS and crimeware trends between 2008 and October 2017 based on these classifications. the most common classes overall were botnets (17%) and exploits (17%). The most popular classes in 2017 were botnets (33%), VPN services (20%), exploits (13%), and brute force attack services (7%). To validate our classification model, we used a confusion matrix, a common method of calculating classifier output accuracy. The training and testing datasets comprised 300 and 700 items, respectively. This gave an accuracy of 82.6% with a 95% confidence interval of (70.74%, 81.24%) for identifying the risks posed by CaaS- and crimeware related messages.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Introduction

A **Software Requirements Specification (SRS)** is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application. There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and nonfunctional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

3.1.1 Purpose of the requirements document:

A requirements document is a [document](#) containing all the requirements to a certain product. It is written to allow people to understand *what* a product should do. Purpose and [scope](#), from both a technical and business perspective. Product overview and use cases Requirements, including functional requirements (e.g. what a product should do) usability requirements technical requirements (e.g. security, network, platform, integration, client) environmental requirements support requirements interaction requirements (e.g. how the product should work with other systems) Assumptions Constraints Dependencies.

3.1.2 Project Perspective:

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.

3.2 Functional Requirements:

Functional requirements are used for expressing the behavior of a project, purpose and role each component. This is represented as using inputs, outputs and its behavior based on the specified input. While implementing design phase of system, functional requirements are considered and behavior of the whole project is realized using the use cases. The use case depicts the behavior, relationship between the components or modules of the project and the use case explanation for this project is illustrated in system design chapter.

- Graphical User Interface with the User & Admin Interface.
- Upload/Review Module.
- Chat Modules.
- Download/View Module.
- Feedback Module.
- Data Analysis via Bar/Pie/Spline Chart.

3.3 Non-Functional Requirements:

3.3.1 Practicality

Project practicality is where you use only what is viewed as least needed to meet your goals. A communication system like this needs hundreds of thousands of users to survive and thrive. Therefore, it should be designed to support large numbers of users, e.g., a substantial percent of a town or a campus.

3.3.2 Efficiency

Efficiency measures how well and productively a manager uses his resources to achieve goals. Project management places heavy focus on how to acquire the right project team to perform project tasks and to close project successfully within the agreed constraints.

3.3.3 Cost

Cost management is concerned with the process of planning and controlling the budget of a project or business. It includes activities such as planning, estimating, budgeting, financing, funding, managing, and controlling costs so that the project can be completed

within the approved budget. Cost management covers the full life cycle of a project from the initial planning phase towards measuring the actual cost performance and project completion.

3.3.4 Flexibility

The need for flexibility is to deal with changed circumstances. Flexibility is used to scale back activities needing less effort while diverting resources to areas with unexpected problems.

3.3.5 Modularity

Modularity refers to the concept of making multiple [modules](#) first and then linking and combining them to form a complete system. Modularity enables re-usability and minimizes duplication. In addition to re-usability, modularity also makes it easier to fix problems as bugs can be traced to specific system modules, thus limiting the scope of detailed error searching.

3.3.6 Extensibility

Extensibility is a [software engineering](#) and [systems design](#) principle where the implementation takes future growth into consideration. The term extensibility can also be seen as a systemic measure of the ability to extend a [system](#) and the level of effort required to implement the extension. Extensions can be through the addition of new functionality or through modification of existing functionality.

3.3.7 Reliability

Reliability refers to the probability and or the likelihood that a given product will perform in the way and or manner it was intended to perform in the efforts that have been deemed required of that given product within or under a specific period of time required.

3.3.8 Maintainability

It is the probability that a system or system element can be repaired in a defined environment within a specified period of time. Increased maintainability implies shorter repair times.

3.3.9 Portability

It is a measure of how easily an application can be transferred from one computer environment to another. A computer software application is considered portable to a new environment if the effort required to adapt it to the new environment is within reasonable limits. The phrase "to port" means to modify software and make it adaptable to work on a different computer system.

3.4 System Requirements:

3.4.1 Hardware Requirements

- Processor : Intel i3 7th Gen 2.4 GHz
- Hard Disk : Min 40 GB
- RAM : 4GB or above
- External : Mouse,Keyboard

3.4.2 Software Requirements

- Operating system : Windows 10
- Front End : HTML5, CSS3, JavaScript, Bootstraps
- Coding Language : Python
- Database : MySQL/Sqlite3
- Tool : Pycharm IDE
- Framework : Django web frameworks
- Browsers : Chrome, Mozilla firefox

3.4.2 Software Requirements Specification (SRS) Table

In this CHAPTER we specifically give a brief about Requirement Traceability Matrix which is mainly to check whether the requirements are satisfied or not.

Serial No.	Requirements ID	Requirement In Brief	Requirement Description
1	RID1	Administrator	admin is an business entiry, person to maintain the entire system.
2	RID2	Users	Users is the one who visits the application and can access the basic information.
3	RID3	System Maintainers (Registered User)	System Maintainer is the service receiver
4	RID4	Home Page	When we run the application we get the application home page.
5	RID5	Login Module	User of the system gets login using the credentials (id an password).
6	RID6	Dataset	In this module we collect opinion dataset from the reputed websites
7	RID7	Data Mining	Process of analysing the data from different perspective and summarize it into useful information.
8	RID8	Machine Learning	Used to develop softwares, microsoft technology more developer friendly.
9	RID9	Django, Pycharm	Used to develop browser based application.
10	RID10	Python	More Suitable programming langauage for real time projects.
11	RID11	MYSQL Server	Microsoft technology used to store the data (opinions data)
12	RID12	Browsers	IE, Google Chrome, Firefox

Table 3.4.2 SRS Table

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction to System Design

System design is a substantially important phase in project building and creation stages of whole development cycle. The system design is a mechanism for depicting and representing the overall architecture of the system, interfaces between the different components, the methods and parameters defined for each module and data for a system according to requirements specified by the user.

In order to design a system, first step is to collect the system requirements, functional and non functional requirements , constraints from the user. Second step is designing the system in an abstract manner, this step provides outline of all major components that is required for designing system architecture. Third step is detecting and addressing bottlenecks generated in the abstract or high-level design due to violation of some constraints specified by the user. Next operation is designing system in more elaborate and detailed manner and this step constitutes specifying the methods, parameters, interfaces to application components.

4.2 High Level Diagram

High level design reveals an abstract layout of entire application where abstract HLD pictorially depicts primitive constituents of system to be developed. The architecture of the system, the diagrams depicting flow of relationship, flow of data are all considered as the high level designs and these designs are written using non-technical terms with slight additional technical terms.

4.2.1 System Architecture

System Architecture of application projects a blueprint of entire system in pictorial illustration. In this project the architecture three major components: Upload ,Download and User Chat as shown in the figure 4.2.1

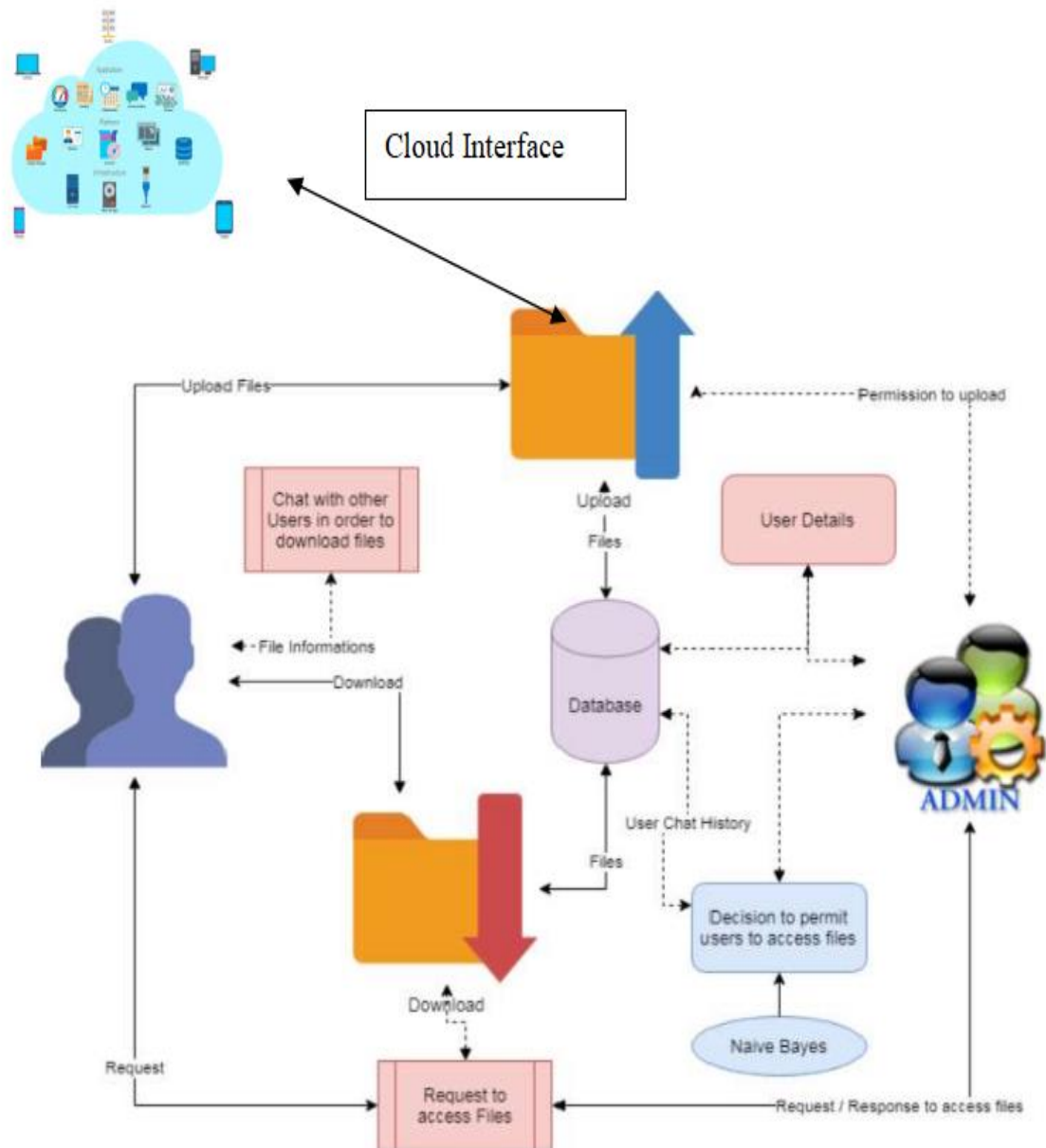


Figure 4.2.1 Architecture of proposed model.

- Step 1:User upload: Users upload their data through any media to the database or cloud server.
- Step 2:Admin process: Admin response for the request and process data.
- Step 3:Decisions to provide access: Using algorithm for decision making and provide access to requested customers.
- Step 4:Download access files: Tracking of files and download the requested files.

5.2.2 Data Flow Diagram

Data-flow diagram is a way of representing a flow of a data of a process or a system and information about the outputs and inputs of each entity and the process itself.

Business Entity/Admin :

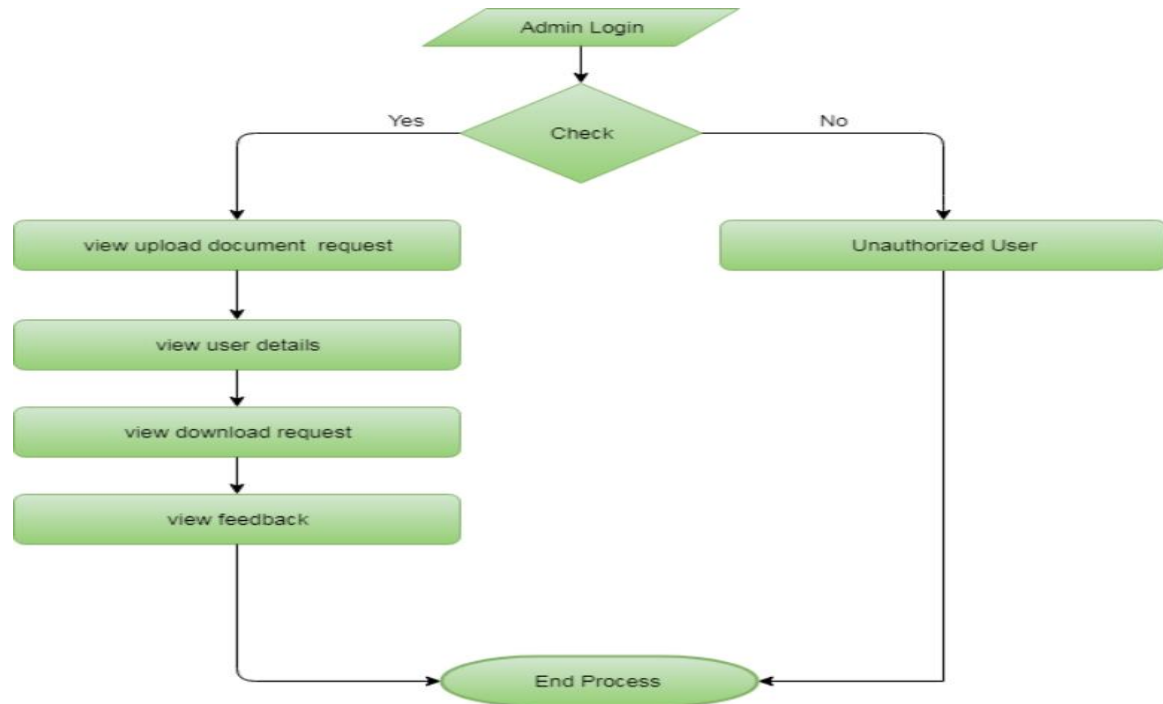


Figure 4.2.2(i):Data Flow diagram for admin module

User:

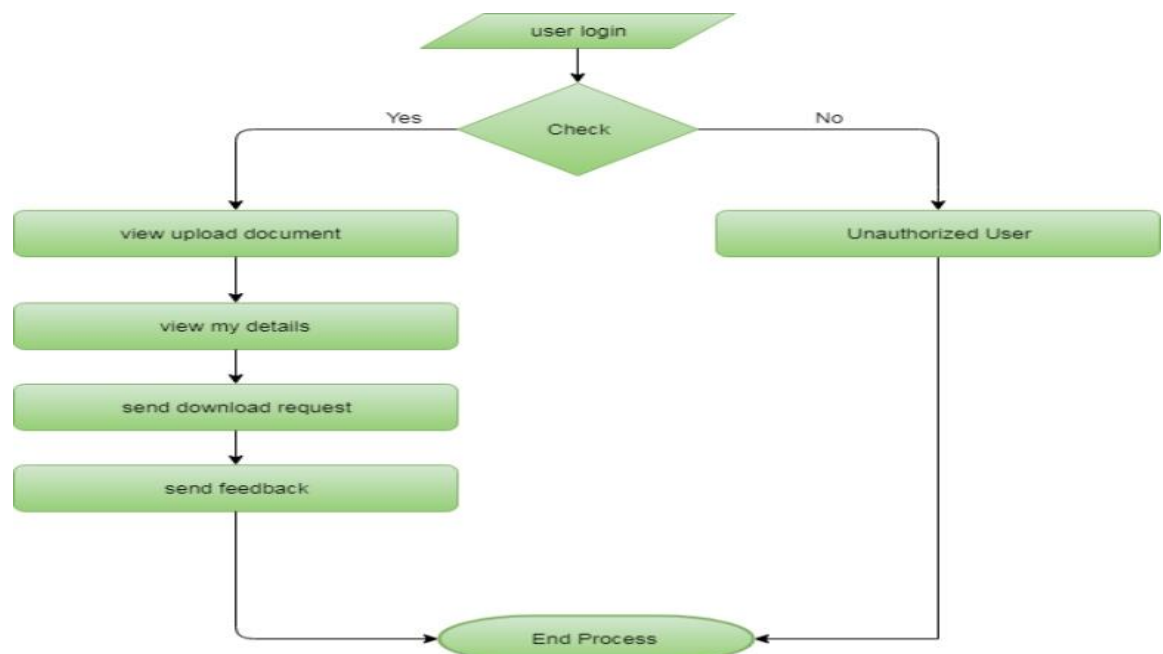


Figure 4.2.2(i):Data Flow Diagram for user

4.3 Low Level Design

Low level design is used describing major components of system in detail and elaborate manner so this technique is also called detailed-design. In this technique the diagram is constructed by iteratively refining the given details, requirements and constraints and also depicts modules, their parameters, methods and relationship among them.

4.3.1 Flow Chart

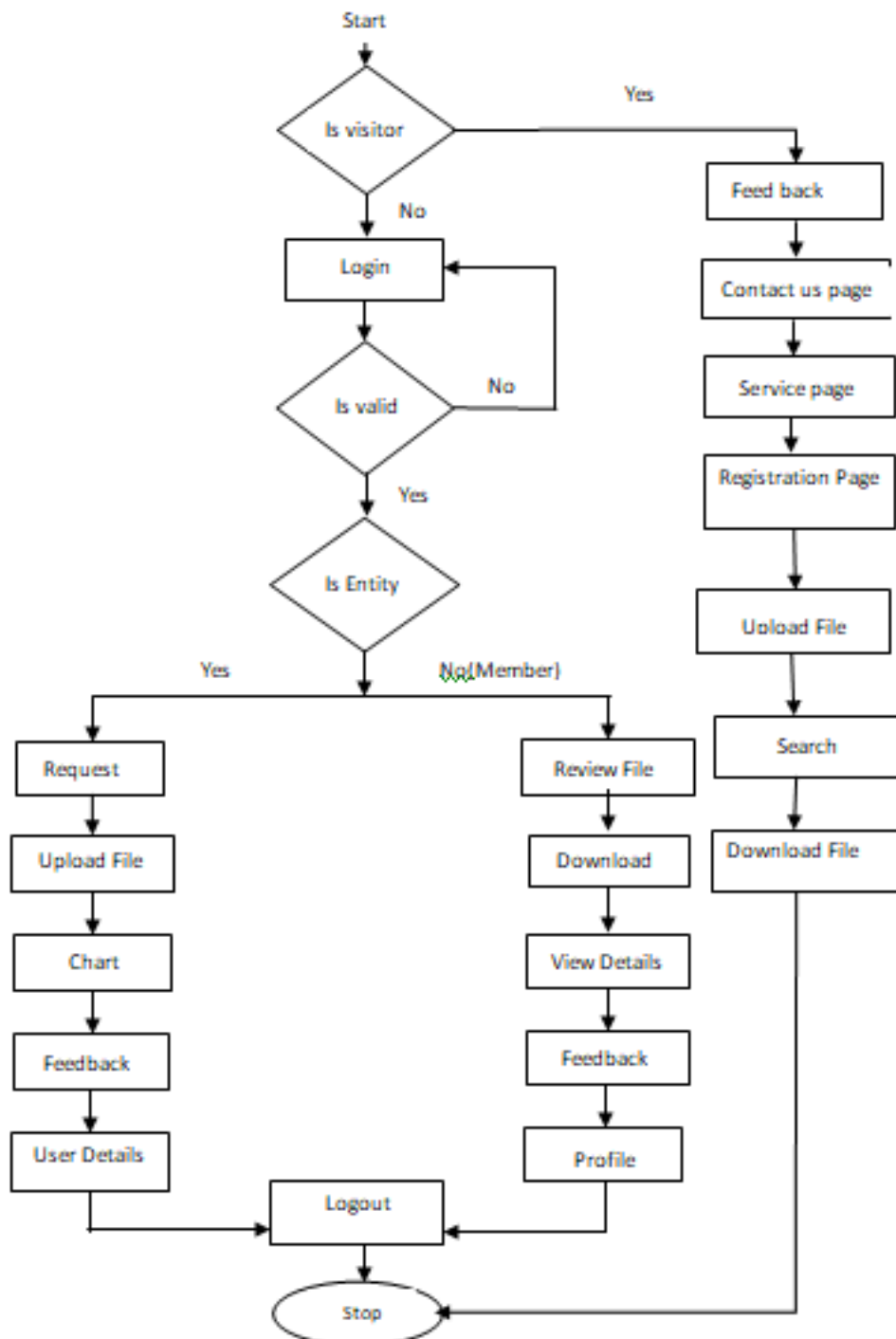


Figure 4.3.1 Flow Chart

4.3.2 Sequence Diagram

a) User

When Admin sign up successfully he/she can view, add and delete eco products category and also he/she can view ,add and delete any eco products, as well as view and respond to customer orders and can also view customer feedback for future upgradation and at last admin also have an option to update his/her password.

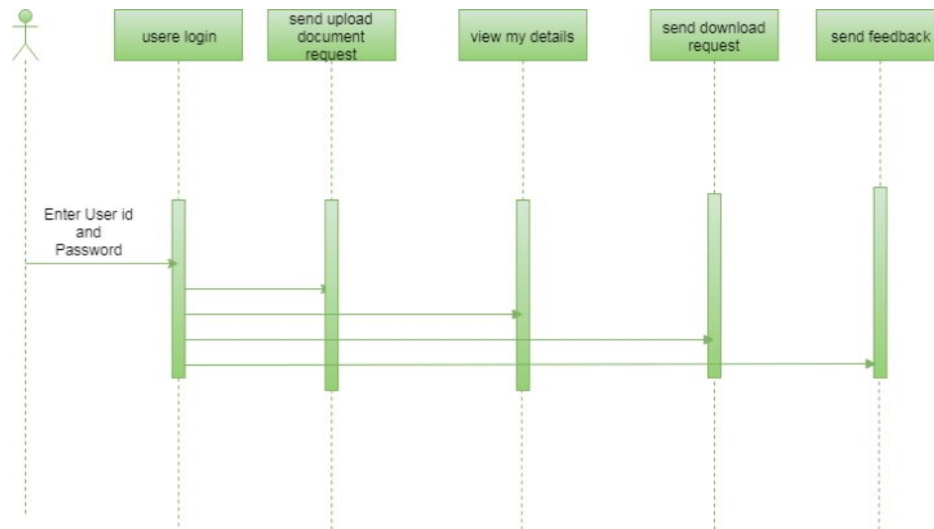


Figure 4.2.2(i): Sequence Diagram For User

b) Admin:

When a user gets signed up, he/she can view or delete review order and can add to place a order , also can view order history and can add his/her feedback and at last he/she can also update his/her profile

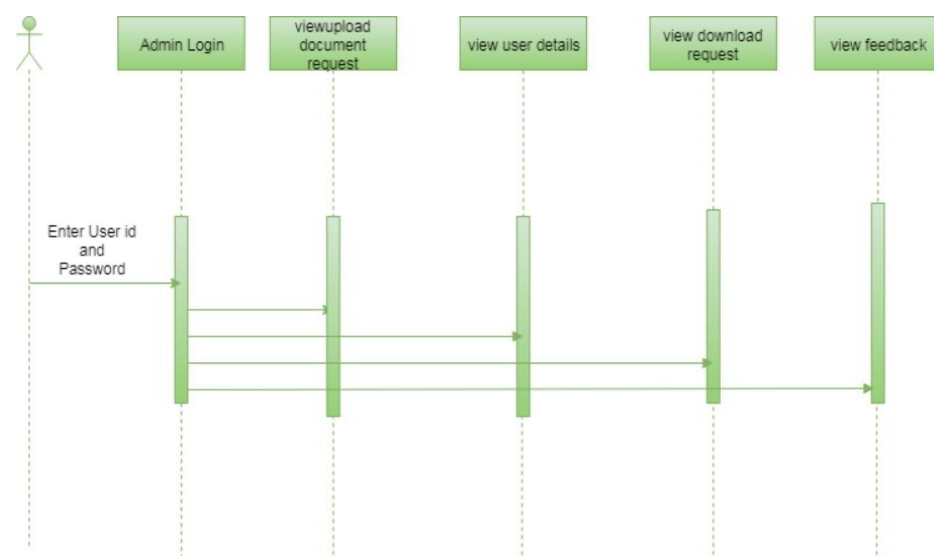


Figure 4.2.2(ii): Sequence Diagram For Admin

4.3.3 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

Business Entity/User:

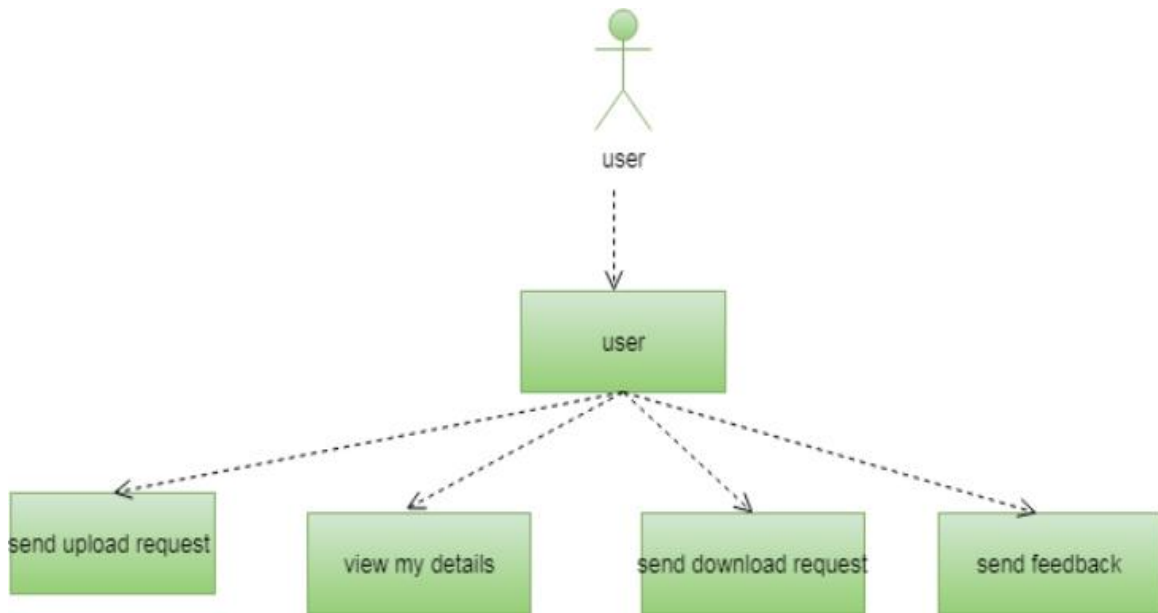


Figure 4.3.3(i) Use Case Diagram for Entity/User

Admin:

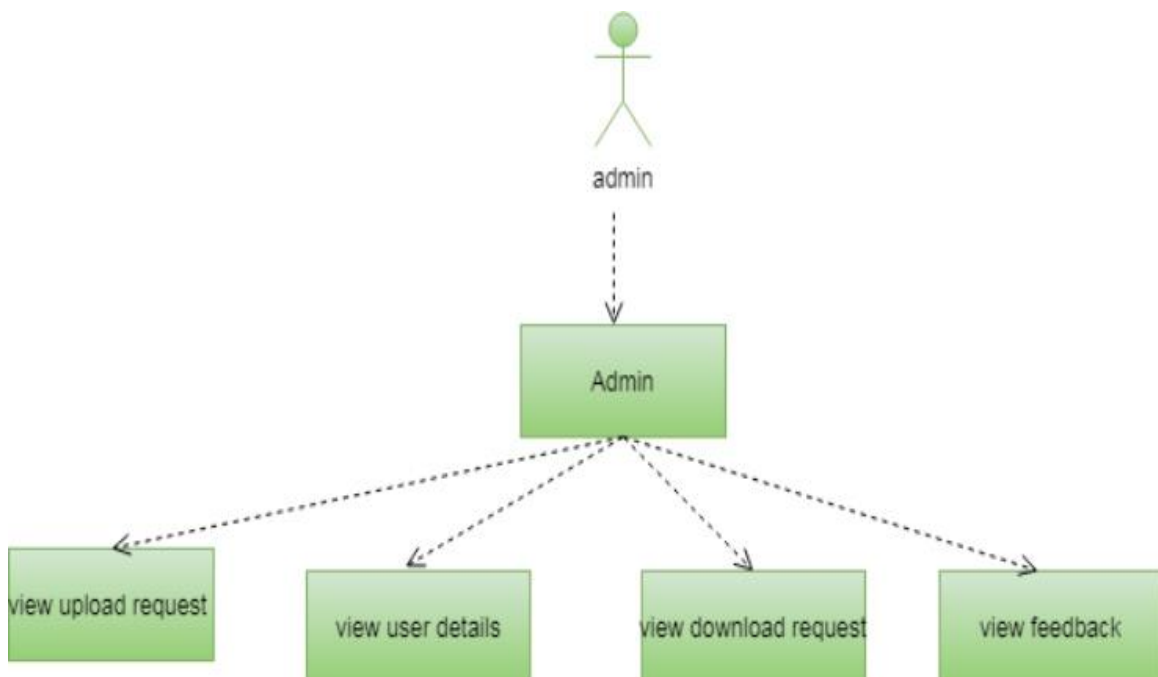


Figure 4.3.3(ii) Use Case Diagram for Entity/User

4.3.4 Activity Diagram

Activity diagram is a kind of behavioral representation which describes how workflows in overall system and is used to describe actions, interactions and activities of system in step-by-step manner. It can also be called as a type of flowchart. Figure 5.5 showcases activity diagram which describes how the workflows between all the modules i.e job tracker, mapping process and reducing job.

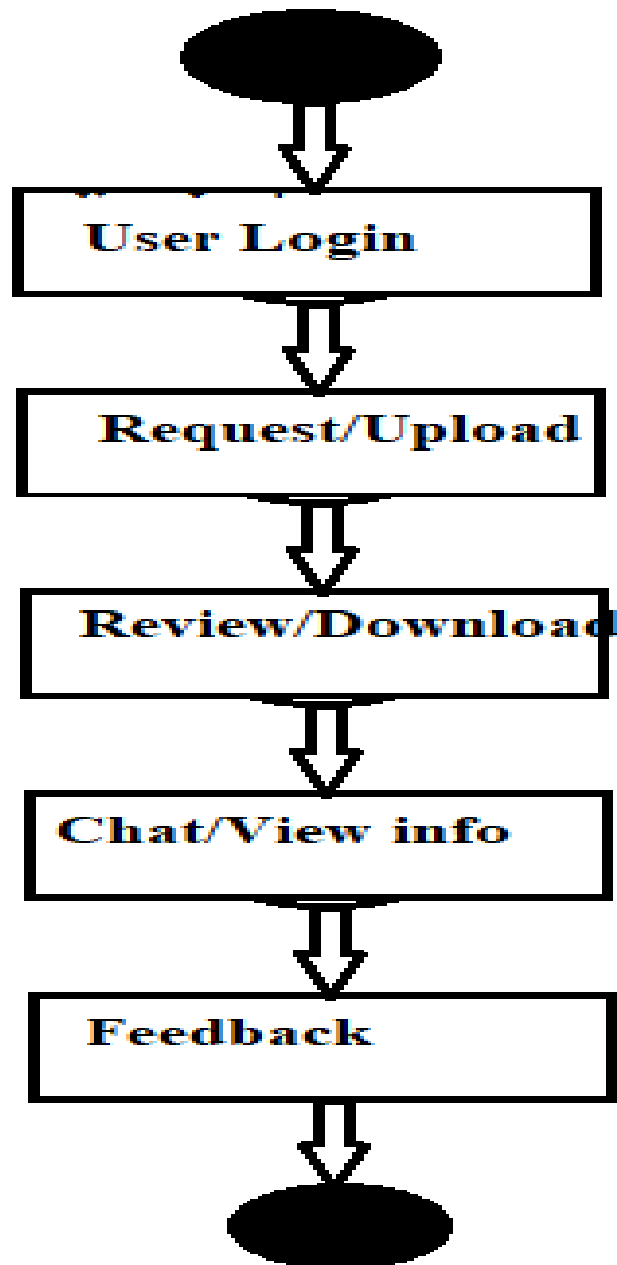


Figure 4.3.4 Activity Diagram

- The victim will not be directly targeted by the attacker. Instead he delivers malicious java scripts while the user visits the website. Here when the attacker injects the malicious java scripts via login page.
- He will be directly allowed to access the database of our website where he can get all important information he needed.
- Finally the user can use the victim's stolen cookie.

5.2 Modules and Methodology

- In contrast, the goal of our data analysis framework is to conduct a big-picture investigation of the cybercrime underground by covering all phases of data analysis from the beginning to the end .
- This framework comprises four steps:

(1) defining goals; (2) identifying sources; (3) selecting analytical methods; and (4) implementing an application.
- The data analysis step of the proposed framework involves four steps. Here, we report the data analysis results: CaaS and crimeware classification and market trends, cybercrime market dynamics, and potential hacking targets.

A. CaaS and Crimeware Classification and Market Trends .

B. Cybercrime Market Dynamics

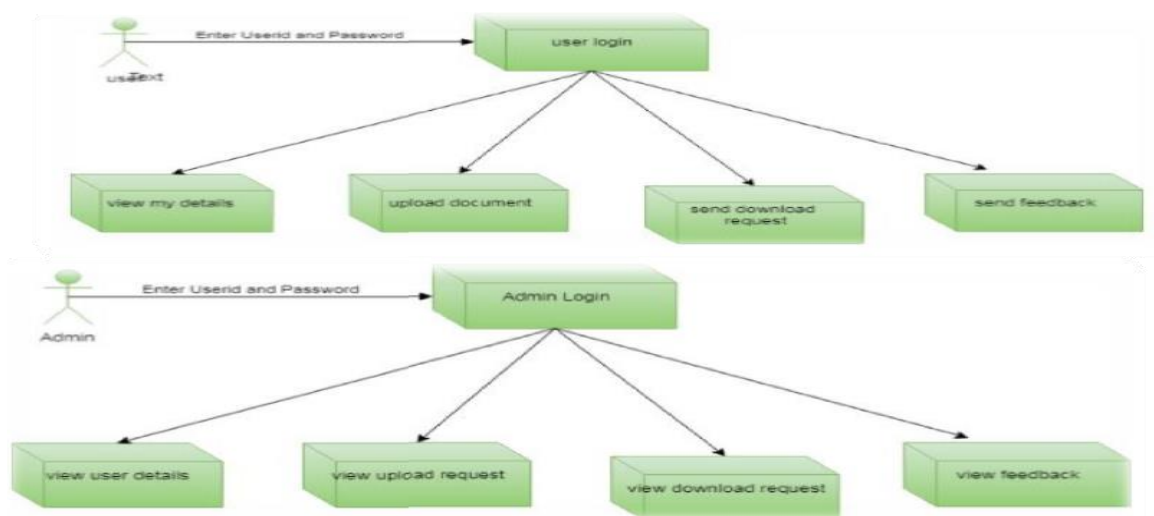


Figure 5.2.1: System Design Modules.

5.2.2 Modules



Figure 5.2.2: Modules for Implementation

Modules that are Implementing this system include the following,

1.Upload Files

Users are allowed to upload the files with the tags given. Once the file is uploaded, then it is sent to approval from admin to publish or make view to other users. These uploaded files can be in any form document, audio or video but not allowed to upload the executable (.exe) files.

2.Conversation Monitoring

Users are allowed to communicate among the other users. This could be monitor by the admin. The malicious conversion likes to threaten the data. In order to protect the cybercrime and prevents from forming cybercrime community. This can be achieved by the help of classification algorithm named naïve Bayes classification.

3.Download Files

The files can be downloading by requesting for the file and once admin approved the files then can be downloadable. The decision to approve files can be taken from the conversation between users. Admin takes the action on download files and approvable status of users. The users are allowed further actions based on the users.

5.Graphical Representations

The analyses of proposed systems are calculated based on the approvals and disapprovals. This can be measured with the help of graphical notations such as pie chart, bar chart and line chart. The data can be given in a dynamical data.

CHAPTER 6

IMPLEMENTATION

6.1 Introduction to System Implementation

Implementation stage of an application creation is actualization of ideas, design and requirement specification into source code. The primary objective of implementation part of building a project is production of source codes with good style and comments when necessary, by applying a proper and best coding technique which is suitable with the help of proper documents. Program codes are created in accordance to the structured coding techniques, which adheres to control flow, so that execution sequence follows the order in which codes are scripted. This makes the code unambiguous and more readable, which eases understanding, modifying, debugging, testing, and documentation of the programs.

6.2 Language Used for Implementation

1.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. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Current Python version using for project is python3.x

2.DJANGO

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

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

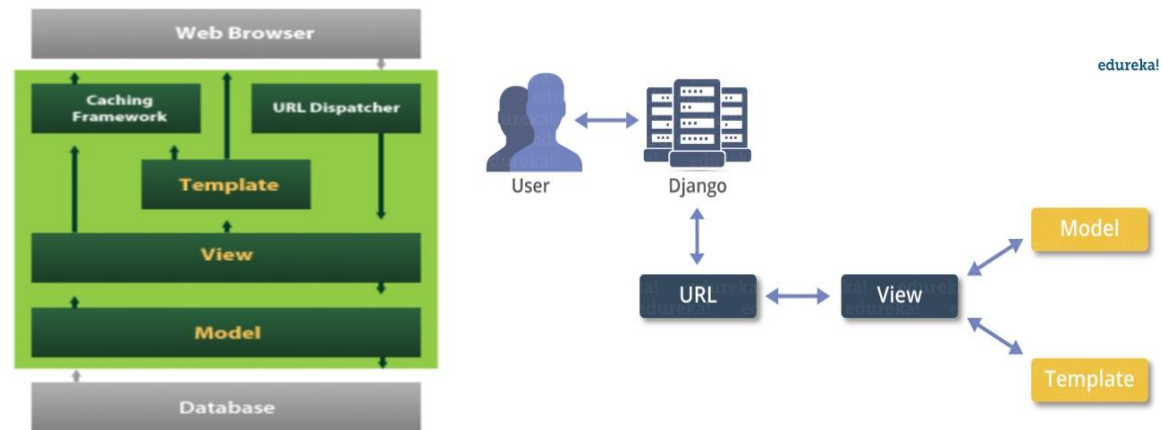


Figure 6.2.2 Django web framework

Django also provides an optional administrative [create, read, update and delete](#) interface that is generated dynamically through [introspection](#) and configured via admin models. Current version used is Django2.2.5

6.3 Algorithm: Naive Bayes Classifier

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis is simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value $P(d_1, d_2, d_3|h)$, they are assumed to be conditionally independent given the target value and calculated as $P(d_1|h) * P(d_2|h)$ and so on. This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

Make Predictions with a Naive Bayes Model. Given a naive Bayes model, you can make predictions for new data using Bayes theorem.

$$\text{MAP}(h) = \max(P(d|h) * P(h))$$

Using our example above, if we had a new instance with the weather of sunny, we can calculate:

$$\text{go-out} = P(\text{weather=sunny} \mid \text{class=go-out}) * P(\text{class=go-out}) \quad \text{stay-home} = P(\text{weather=sunny} \mid \text{class=stay-home}) * P(\text{class=stay-home})$$

We can choose the class that has the largest calculated value. We can turn these values into probabilities by normalizing them as follows:

$$P(\text{go-out} \mid \text{weather=sunny}) = \text{go-out} / (\text{go-out} + \text{stay-home}) \quad P(\text{stay-home} \mid \text{weather=sunny}) = \text{stay-home} / (\text{go-out} + \text{stay-home})$$

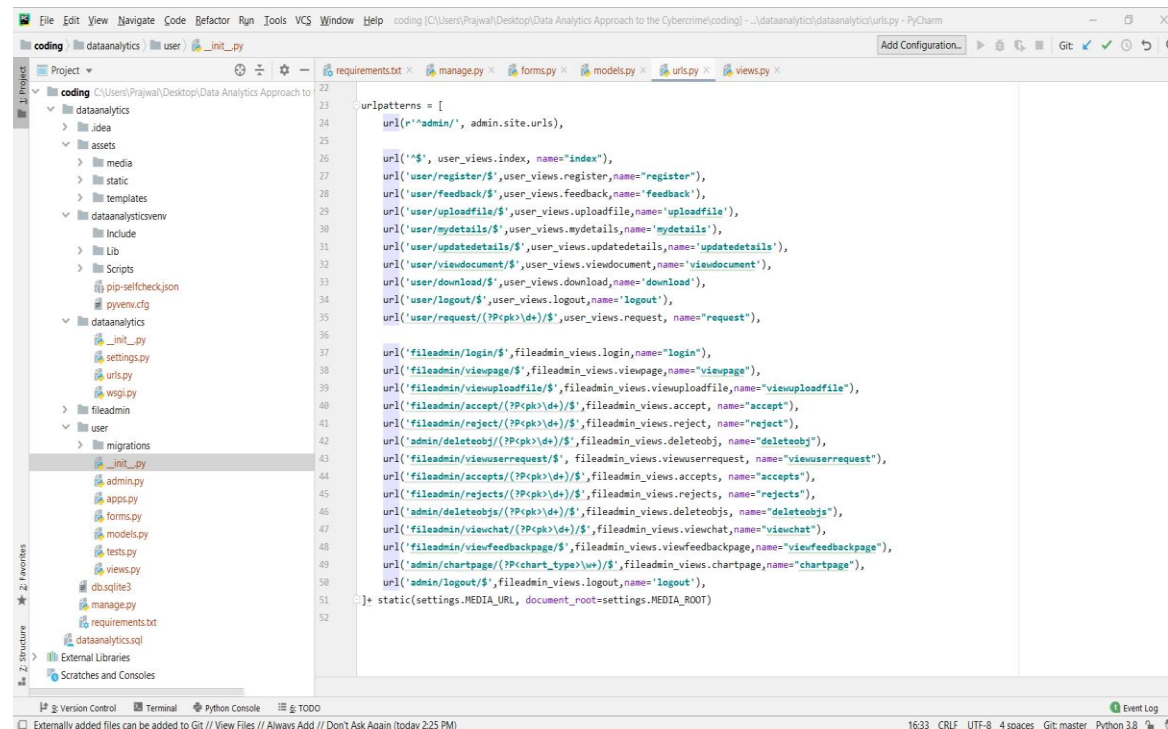
If we had more input variables we could extend the above example. For example, pretend we have a “car” attribute with the values “working” and “broken”.

We can multiply this probability into the equation. For example below is the calculation for the “go-out” class label with the addition of the car input variable set to “working”:

$$\text{go-out} = P(\text{weather=sunny} \mid \text{class=go-out}) * P(\text{car=working} \mid \text{class=go-out}) * P(\text{class=go-out})$$

6.4 Snapshot of Code Snippet

6.4.1 urls.py



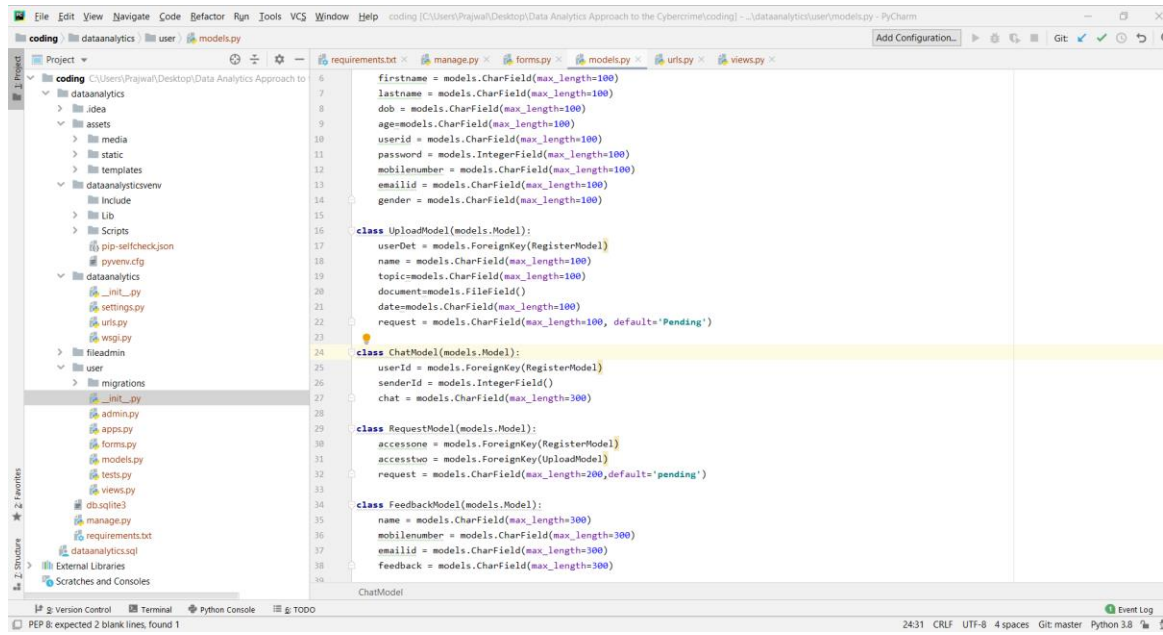
```

22  urlpatterns = [
23      url(r'^admin/', admin.site.urls),
24
25      url(r'^$', user_views.index, name="index"),
26      url('user/register/$', user_views.register, name="register"),
27      url('user/feedback/$', user_views.feedback, name="feedback"),
28      url('user/uploadfile/$', user_views.uploadfile, name="uploadfile"),
29      url('user/mydetails/$', user_views.mydetails, name="mydetails"),
30      url('user/updatedetails/$', user_views.updatedetails, name="updatedetails"),
31      url('user/viewdocument/$', user_views.viewdocument, name="viewdocument"),
32      url('user/download/$', user_views.download, name="download"),
33      url('user/logout/$', user_views.logout, name="logout"),
34      url('user/request/(?P<pk>[0-9]+)/$', user_views.request, name="request"),
35
36      url('fileadmin/login/$', fileadmin_views.login, name="login"),
37      url('fileadmin/viewpage/$', fileadmin_views.viewpage, name="viewpage"),
38      url('fileadmin/viewuploadfile/$', fileadmin_views.viewuploadfile, name="viewuploadfile"),
39      url('fileadmin/accept/(?P<pk>[0-9]+)/$', fileadmin_views.accept, name="accept"),
40      url('fileadmin/reject/(?P<pk>[0-9]+)/$', fileadmin_views.reject, name="reject"),
41      url('admin/deleteobj/(?P<pk>[0-9]+)/$', fileadmin_views.deleteobj, name="deleteobj"),
42      url('fileadmin/viewuserrequest/$', fileadmin_views.viewuserrequest, name="viewuserrequest"),
43      url('fileadmin/accepts/(?P<pk>[0-9]+)/$', fileadmin_views.accepts, name="accepts"),
44      url('fileadmin/rejects/(?P<pk>[0-9]+)/$', fileadmin_views.rejects, name="rejects"),
45      url('admin/deleteobjs/(?P<pk>[0-9]+)/$', fileadmin_views.deleteobjs, name="deleteobjs"),
46      url('fileadmin/viewchat/(?P<pk>[0-9]+)/$', fileadmin_views.viewchat, name="viewchat"),
47      url('fileadmin/viewfeedbackpage/$', fileadmin_views.viewfeedbackpage, name="viewfeedbackpage"),
48      url('admin/chartpage/(?P<chart_type>[a-z]+)/$', fileadmin_views.chartpage, name="chartpage"),
49      url('admin/logout/$', fileadmin_views.logout, name="logout"),
50  ]
51  static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
52

```

Figure 6.5.1 urls.py

6.4.2 models.py



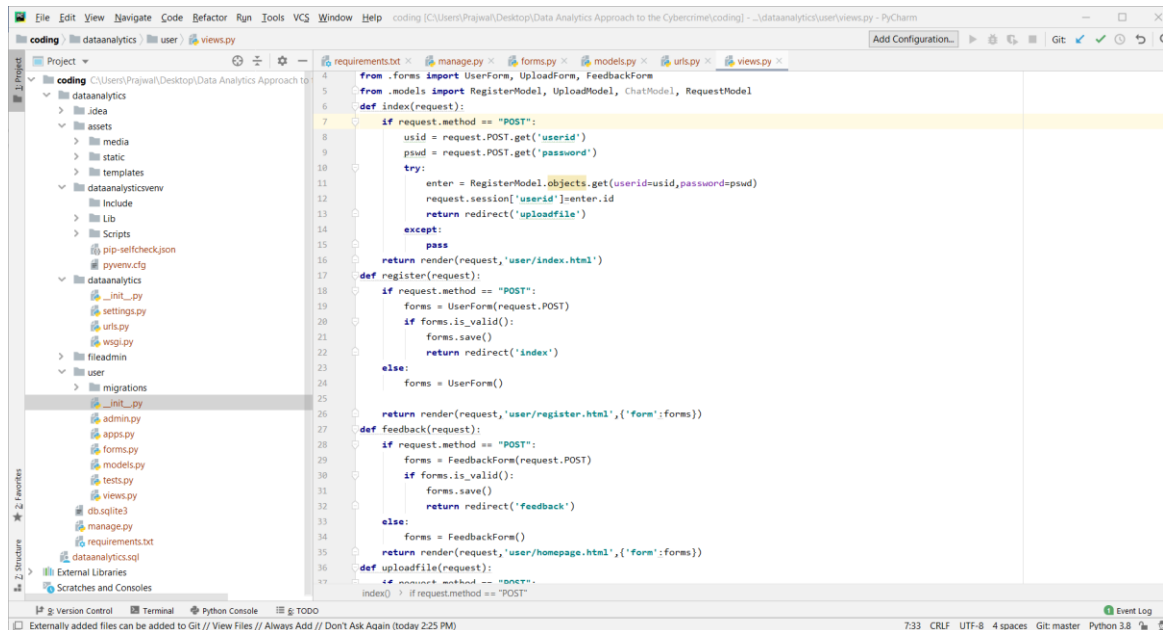
```

6  first_name = models.CharField(max_length=100)
7  last_name = models.CharField(max_length=100)
8  dob = models.CharField(max_length=100)
9  age = models.CharField(max_length=100)
10 user_id = models.CharField(max_length=100)
11 password = models.IntegerField(max_length=100)
12 mobilenummer = models.CharField(max_length=100)
13 emailid = models.CharField(max_length=100)
14 gender = models.CharField(max_length=100)
15
16 class UploadModel(models.Model):
17     userDet = models.ForeignKey(RegisterModel)
18     name = models.CharField(max_length=100)
19     topic = models.CharField(max_length=100)
20     document = models.FileField()
21     date = models.CharField(max_length=100)
22     request = models.CharField(max_length=100, default='Pending')
23
24 class ChatModel(models.Model):
25     userId = models.ForeignKey(RegisterModel)
26     senderId = models.IntegerField()
27     chat = models.CharField(max_length=300)
28
29 class RequestModel(models.Model):
30     accessone = models.ForeignKey(RegisterModel)
31     accesstwo = models.ForeignKey(UploadModel)
32     request = models.CharField(max_length=200, default='pending')
33
34 class FeedbackModel(models.Model):
35     name = models.CharField(max_length=300)
36     mobilenummer = models.CharField(max_length=300)
37     emailid = models.CharField(max_length=300)
38     feedback = models.CharField(max_length=300)
39
40 ChatModel

```

Figure 6.5.2 models.py

6.4.3 views.py



```

4  from .forms import UserForm, UploadForm, FeedbackForm
5  from .models import RegisterModel, UploadModel, ChatModel, RequestModel
6  def index(request):
7      if request.method == "POST":
8          usid = request.POST.get('userid')
9          psud = request.POST.get('password')
10         try:
11             enter = RegisterModel.objects.get(userid=usid,password=psud)
12             request.session['userid']=enter.id
13             return redirect('uploadfile')
14         except:
15             pass
16         return render(request, 'user/index.html')
17     def register(request):
18         if request.method == "POST":
19             forms = UserForm(request.POST)
20             if forms.is_valid():
21                 forms.save()
22                 return redirect('index')
23         else:
24             forms = UserForm()
25         return render(request, 'user/register.html', {'form': forms})
26     def feedback(request):
27         if request.method == "POST":
28             forms = FeedbackForm(request.POST)
29             if forms.is_valid():
30                 forms.save()
31                 return redirect('feedback')
32         else:
33             forms = FeedbackForm()
34         return render(request, 'user/homepage.html', {'form': forms})
35     def uploadfile(request):
36         if request.method == "POST":
37             index()
38         if request.method == "POST":

```

Figure 6.5.3 views.py

CHAPTER 7

TESTING

7.1 Introduction to Testing

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

7.2 TYPES OF TESTS

7.2.1 Unit testing

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

7.2.2 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.

7.2.3 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 centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

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

i. 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.

ii. White Box Testing

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

iii. Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. 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.

7.2 Test Cases Table

Test case ID	Description	Test steps	Expected value	Actual value	OK/Error
1.	Verify login page	Input username and password	Login page	Invalid data	error
2	Verify login page	Input username and password	Login page	Login page	ok
3.	Verify registration page	Registration	User profile	Registration failed	error
6.	Verify registration page	Registration	User profile	User profile	ok
5.	Query is to be posted	Posting query	Query is posted	Unable to post the query	error
6.	Query is to be posted	Posting query	Query is posted	Successfully posted	ok
7.	Files to be upload	Upload file	File to upload	File name is entered	ok
8.	Files to be upload	Upload file	File to upload	Unknown file	error
9.	Files to be requested	Enter request file	File to be send	File name is Request	ok
10.	Files to be reuested	Enter request file	File to be send	File not found	error
11.	Chat Module	Enter text	Negative Text is entered	Possitive Text is entered	error
12.	Chat Module	Enter text	Possitive Text is entered	Possitive Text is entered	ok

Table 7.2 Test cases

CHAPTER 8

RESULT ANALYSIS

8.1 Snapshot with Description

8.1.1 Login Page/Home Page :



Figure 8.1.1 Login Page

- Home Page for User or Admin Login to the System

8.1.2 New User Registration:



Figure 8.1.2 Registration Page

- Registration Page for New Users by entering their basic information.

8.1.3 Upload File Page:

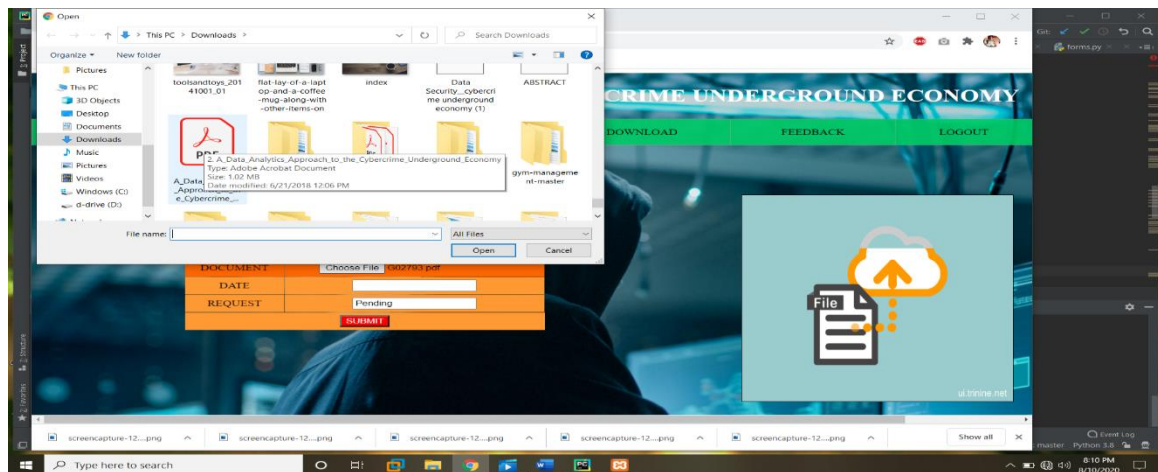


Figure 8.1.3 Upload File Page

8.1.4 View File Uploaded Page:



Figure 8.1.4 View File Uploaded Page

8.1.5 Download Page :

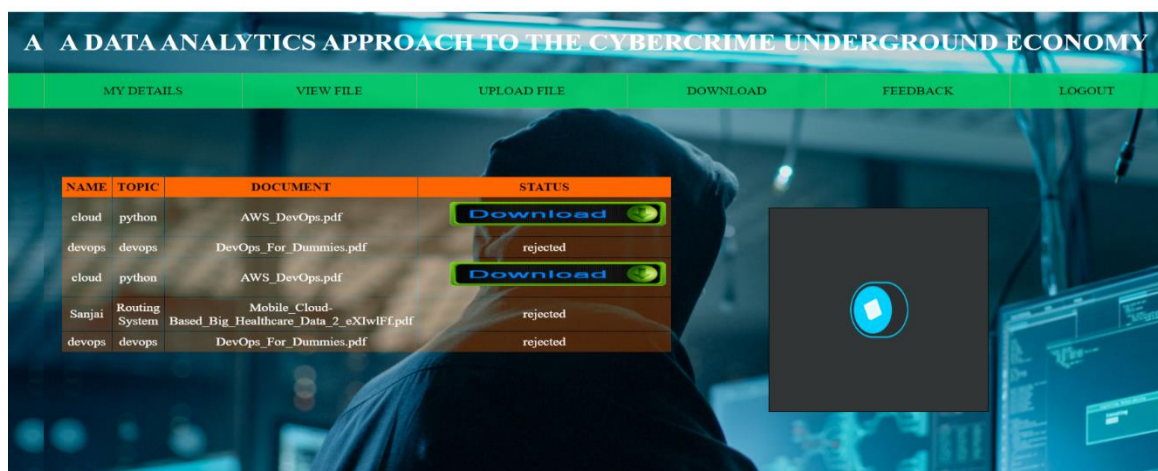


Figure 8.1.5 Download Page

8.1.6 Admin View Uploaded Page :

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY							
USER DETAILS		VIEW UPLOAD FILE		USER REQUEST		CHART	FEEDBACK
FIRST NAME	LAST NAME	DATE OF BIRTH	AGE	USER ID	MOBILE NUMBER	EMAIL ID	
Sabari	Nathan	09-05-1997	21	Sabari	9789672189	sabarinathan@1350@gmail.com	
Sanjai	Kumar	04-11-1996	22	Sanjai	7853974352	sanjai@gmail.com	
Santhosh	Kumar	05-01-1997	21	Santhosh	9674322516	santhosh23@gmail.com	
Suresh	Babu	06-11-1997	21	Suresh	8694326754	suresh@gmail.com	
Somesh	Varan	08-05-1995	23	Somesh	9543742376	sabari43@gmail.com	
Karthik	Raja	06-02-1990	28	Karthik	945173275	karthik@gmail.com	
Soundhar	Rajan	05-03-1996	22	Soundhar	9145725427	soundhar@gmail.com	
Vignesh	Varan	08-04-1992	26	Vignesh	7459287125	vignesh54@gmail.com	
Gokul	Raj	05-02-1994	24	Gokul	9441853851	gokul32@gmail.com	
Mani	Kandan	08-01-1989	29	Mani	9521677421	mani765@gmail.com	
user	user	00/00/0000	21	1	9999999999	user@gmail.com	
user	user	00/00/0000	00	123	9999999999	user@gmail.com	

Figure 8.1.6 Admin View Uploaded Page

8.1.7 Admin View Request Page :

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY							
USER DETAILS		VIEW UPLOAD FILE		USER REQUEST		CHART	FEEDBACK
NAME	TOPIC	DOCUMENT	STATUS	CHAT	REQUEST STATUS	DELETE	
Sabari	Keyset	m1_et4eDAN.txt	accepted	chat	Accept Reject	delete	
Sanjai	Python	Registration_ID_FXgbU9T.txt	accepted	chat	Accept Reject	delete	
Sabari	Keyset	m1_et4eDAN.txt	accepted	chat	Accept Reject	delete	
Santhosh	Java	SIVA_xvIMTG7	accepted	chat	Accept Reject	delete	
Gokul	Python	Prediction_Based_Mobiel_Data_Offloading_1.pdf	accepted	chat	Accept Reject	delete	
Karthik	Security	Coded_Caching_Clusters.pdf	accepted	chat	Accept Reject	delete	
Sanjai	Routing System	Mobile_Cloud-Based_Big_Healthcare_Data_2_eXIwIFF.pdf	accepted	chat	Accept Reject	delete	
Mani	Java	Coded_Caching_Clusters_oYBb9Xm.pdf	accepted	chat	Accept Reject	delete	
Santhosh	Java	SIVA_xvIMTG7	pending	chat	Accept Reject	delete	
Gokul	Hacking	SS_QR_CODE_2.pdf	pending	chat	Accept Reject	delete	
Sanjai	Python	Registration_ID_FXgbU9T.txt	accepted	chat	Accept Reject	delete	

Figure 8.1.7 Admin view Request Page

8.1.8 Feedback Page :

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY			
USER DETAILS		VIEW UPLOAD FILE	USER REQUEST
NAME	MOBILE NUMBER	EMAIL ID	FEEDBACK
Santhosh	9789672189	santhosh43@gmail.com	Thank you,it's very useful for me
Santhosh	9789672189	santhosh43@gmail.com	Thank you,it's very useful for me
Sanjai Kumar	7493674328	talktonoo@gmail.com	it's very well
Suresh Babu	9451734529	smartsuresh@gmail.com	Improve the data base system
Sabari Nathan	9523645741	sabari345@gmail.com	It's good effort
aravana Kumar	9451754336	saravanan12@gmail.com	It's very well
user	9999999999	user@gmail.com	Sir uploading my file

Figure 8.1.8 Feedback Page

8.1.9 Positive Conversation Page :

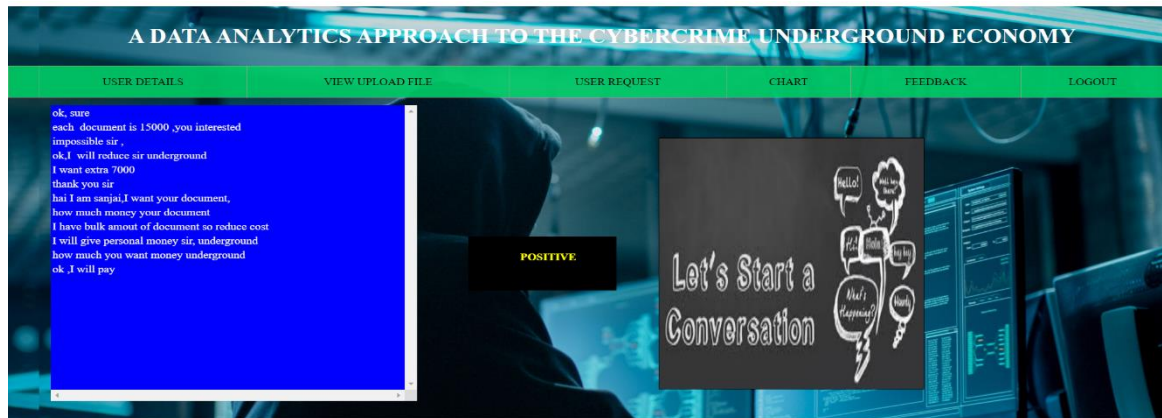


Figure 8.1.9 Positive Conversation Page

8.1.10 Negative Conversation Page :

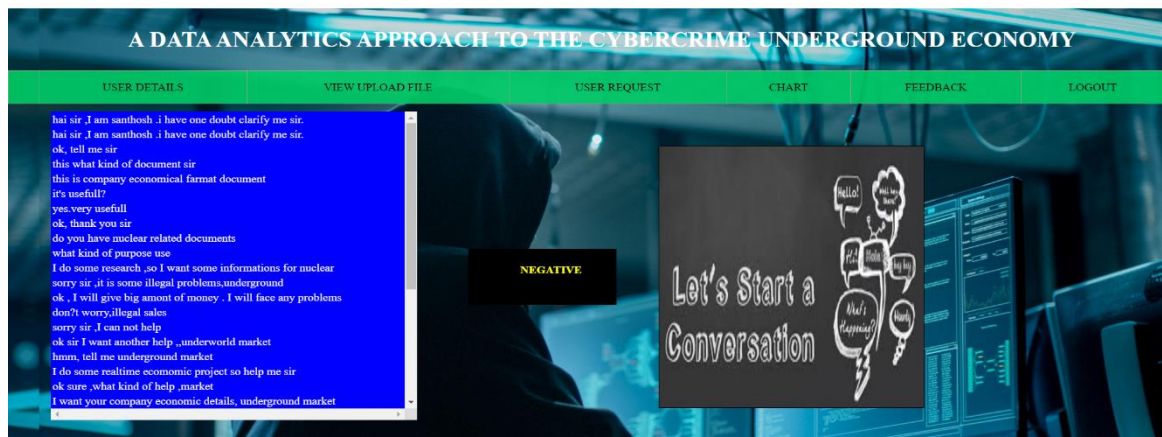


Figure 8.1.10 Negative Conversation Page

8.1.11 CSV file in Excel conversation :

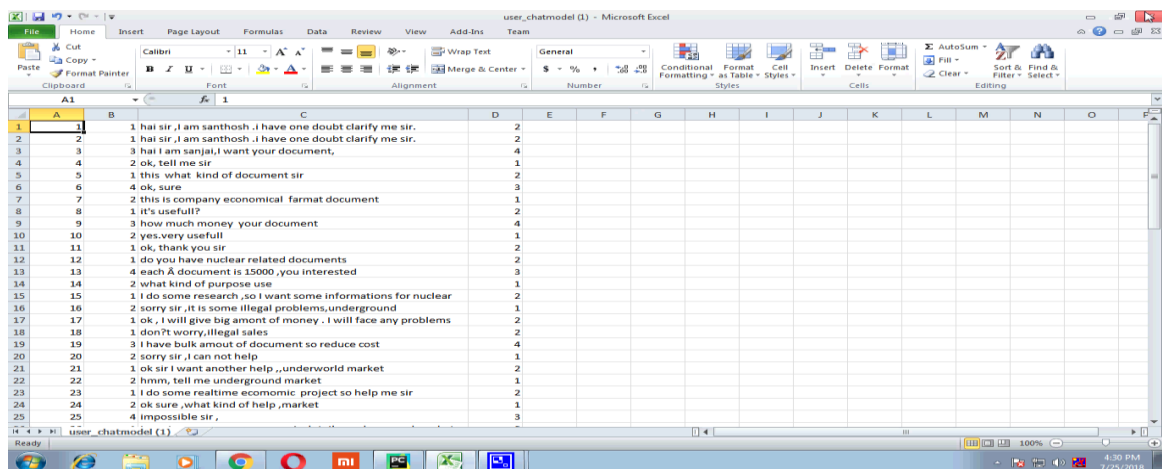


Figure 8.1.11 CSV File in Excel

8.1.12 Data Analysis Graph:

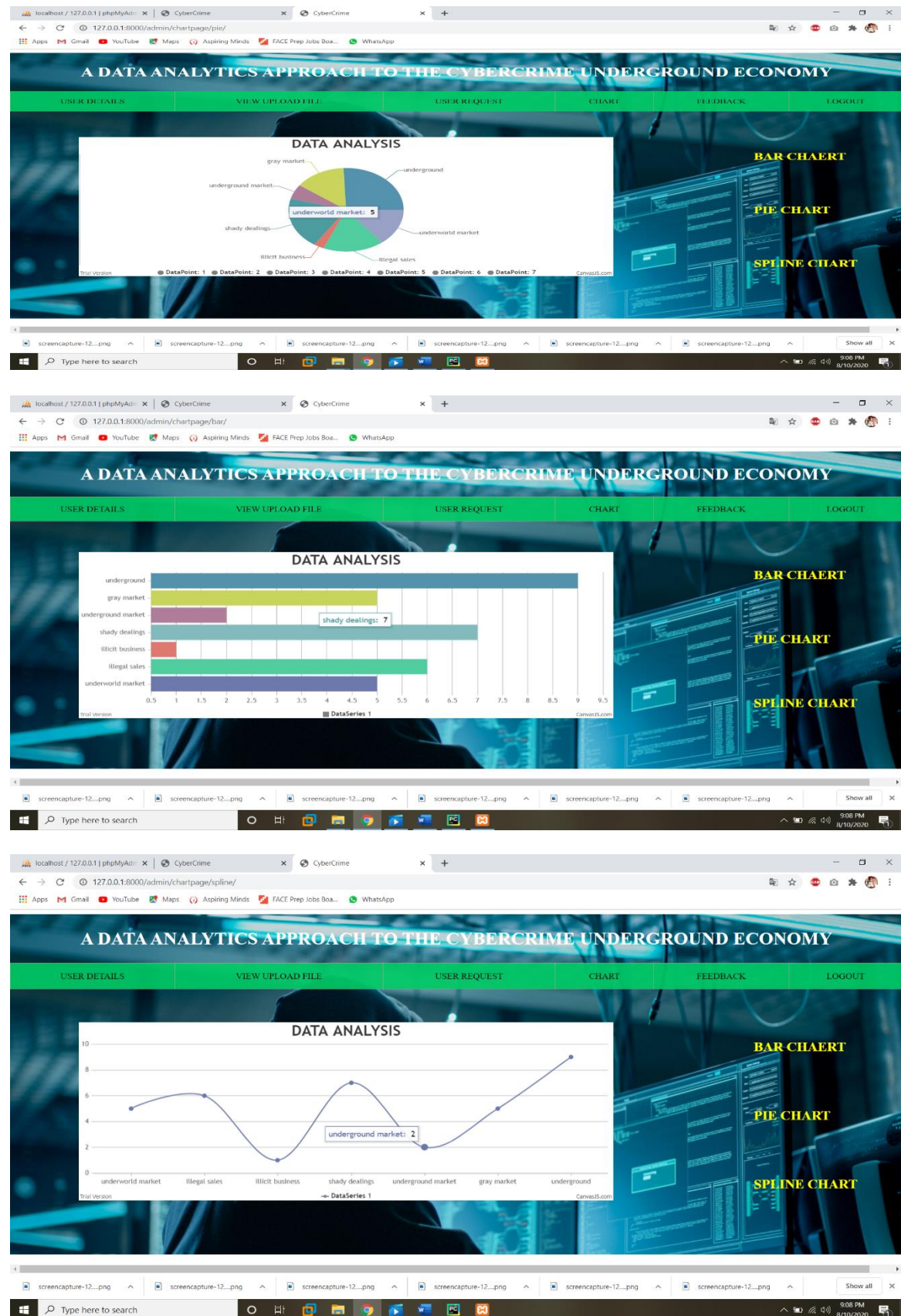


Figure 8.1.12 Data Analysis Graph Chart (Bar,Pie,Spline chart)

CONCLUSION

This study contributes to the DSR literature in a broader IS context in several ways. Because it takes a DSR approach, it contributes to the design artifacts, foundations, and methodologies in this area. First, by creating example front-end applications, we have demonstrated how our design artifacts (the proposed framework and classification model) can be implemented in practice. We have therefore proposed two artifacts: a data analysis framework and a classification model. We have also conducted an ex-ante evaluation of our classification model's accuracy and an ex-post evaluation of its implementation using example applications. In line with the initiation perspective of DSR, these four example applications demonstrate the range of potential practical applications available to future researchers and practitioners. Unlike previous studies that have presented general discussions of a broad range of cybercrime; our study has focused primarily on CaaS and crime ware from an RAT perspective. We have also proposed sets of definitions for different types of CaaS (phishing, brute force attack, DDoS attack, and spamming, crypting, and VPN services) and crime ware (drive-by download, botnets, exploits, ransomware, rootkits, Trojans, crypters , and proxies) based on definitions taken from both the academic and business practice literature.

Looking at the CaaS and crimeware trends, our results show that the prevalence of botnets (attack-related crimeware) and VPNs has increased in 2017. This indicates that attackers consider both the preventive measures taken by organizations and their vulnerabilities. The most common potential target organizations are technology companies (28%), followed by content (22%), finance (20%), e-commerce (12%), and telecommunication (10%) companies. This indicates that a wide variety of companies in a range of industries are becoming potential targets for attackers, having become more vulnerable due to their greater reliance on technology.

FUTURE WORK

Although our study has made several significant findings, it nevertheless has several limitations that will need to be addressed in future studies. These will be able to add more analysis and significant further insights.

First, we only collected data from the largest hacking community and did not consider other hacking communities. Future studies will therefore need to generalize our findings by investigating a wider range of hacking communities.

Second, this study has focused on the CaaS and crimeware available in the cybercrime underground, but much in-depth analysis remains to be done on the configurations of cybercrime networks. Future research could cluster keywords and threats by industry to provide a deeper understanding of the potential vulnerabilities, and it could attempt to discover the network effects involved or the leaders of the cybercrime underground. For example, they should be aware that there are cybercrime underground markets where hacking tools are sold.

Third, this study calls for researchers, companies, antivirus vendors, and governments to collaborate in the fight against cybercrime using civil resources. Rather than acting alone, these groups should unite to maximize their efficiency and effectiveness.

Finally, this study also has important implications for society. Over the last few years, the world has been facing cyber terrorism and cyberwar threats from nation-sponsored attackers. Pollitt defined cyber terrorism as “the premeditated, politically motivated attack against information, computer systems, computer programs and data which results in violence against non-combatant targets by subnational groups or clandestine agents.”

Unlike most cybercrime, which is primarily motivated by monetary gain, cyber terrorists are politically motivated. As a result, governments should, for example, strengthen their ability to protect their citizens in online virtual environments by enhancing their immediate responses to threats such as cyber espionage and cyber terrorism.

REFERENCES

BIBLIOGRAPHY

- [1] **“A Data-Analytics Approach to Cyber Crime Underground Economy”** Jungkook An and Hee-Woong Kim Graduate School of Information IEEE Transactions on Information Forensics and Security (Year : 2018).
- [2] K. Sood, S. Zeadally, and R. Bansal. **“Cybercrime at a Scale: A Practical Study of Deployments of HTTP-Based Botnet Command and Control Panels,”** IEEE Commun. Mag., vol. 55, no. 7, pp. 22– 28.2017.
- [3] Z. Shi, G. M. Lee, and A. B. Whinston, **“Toward a Better Measure of Business Proximity: Topic Modeling for Industry Intelligence,”** MIS Quart., vol. 40, no. 4, pp. 1035–1056, 2016. **“FACT SHEET: Cybersecurity National Action Plan,”** ed: The White House, 2016.
- [4] V. G. Tasiopoulos and S. K. Katsikas, **“Bypassing Antivirus Detection with Encryption,”** in Proc., 18th Panhellenic Conf. on Informatics - PCI '14, New York, New York, USA, 2014, pp. 1–2: ACM Press.
- [5] G. Giacomello, **“Close to the Edge: Cyberterrorism Today,”** in Understanding Terrorism, Emerald Group Publishing Limited, 2014, pp. 217–236.
- [6] A. K. Sood and R. J. Enbody, **“Crimeware-as-a-service—A survey of commoditized crimeware in the underground market,”** Int. J. Crit. Infr. Prot., vol. 6, no. 1, pp. 28–38, 2013.
- [7] A. Majchrzak and S. L. Jarvenpaa, **“Safe Contexts for Interorganizational Collaborations Among Homeland Security Professionals,”** J. Manag. Inf. Syst., vol. 27, no. 2, pp. 55–86, 2010.
- [8] S. W. Brenner, **“Organized Cybercrime? How Cyberspace May Affect the Structure of Criminal Relationships,”** N. C. J. Law & Technol., vol. 4, no. 1, pp. 1-50, 2002.
- [9] R. Venkateswaran, **“Virtual private networks,”** IEEE Potentials, vol. 20, no. 1, pp. 11–15, 2001.
- [10] M.M. Pollitt, **“Cyberterrorism — Fact or Fancy?,”** Comput. Fraud Security, vol. 1998, no. 2, pp. 8–10, 1998.