CyberSential: Automated Vulnerability Assessment Tool

Authors

Sameer Hani 21-SE-98

Supervisor

Dr. Mubbashir Ayub Minhas

DEPARTMENT OF SOFTWARE ENGINEERING

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY TAXILA

June – 2025

CyberSential: Automated Vulnerability Assessment Tool

Author(s)
Sameer Hani
21-SE-98

A thesis submitted in partial fulfillment of the requirements for the degree of

B.Sc. Software Engineering

Thesis Supervisor:
Engr. Dr.
Mubbashir Ayub
Minhas
Associate Professor, Software Engineering

DEPARTMENT OF SOFTWARE ENGINEERING

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

June – 2025

ABSTRACT

CyberSentinel is a forward looking, cutting edge cybersecurity system that discovers, neutralizes and responds to new threats online. It defends web applications and digital infrastructure which in itself is secured through a series of layers of defense including vulnerability scanning, traffic monitoring and attack detection. The system uses AI driven techniques and machine learning models to find suspicious patterns for instance, DDoS attacks, brute force attempts etc and phishing threats. CyberSentinel features such as IP blacklisting, rate limiting and geographic verification with the aid of live dashboards for traffic visualization and automated reporting. Adaptable architecture helps the system to defend against the changing cyber threats by learning the new attack techniques continuously. While this framework provides a proactive, transparent and scalable defense mechanism that is needed for today's cyber security threats.

Keywords:

Machine learning, AI-driven threat detection, DDoS protection, CyberSentinel, cybersecurity, real-time monitoring, and vulnerability scanning

UNDERTAKING

We certify that project work titled "CyberSentinel" is our own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged / referred.

Signature of
Student Sameer
Hani
21-SE-98

ACKNOWLEDGEMENTS

We want to sincerely thank everyone who helped us finish this project and prepare this report. We would especially like to thank our project supervisor, Engr. Dr. Mubbashir Ayub Minhas, whose insightful advice, practical recommendations, and invaluable experience made our project proposal, CyberSentinel: Automated Vulnerability Assessment Tool, a huge success. We would also like to express our gratitude to the faculty and staff of Dr. Mubbashir Ayub Minhas, who provided us with ongoing assistance and access to vital resources, such as cybersecurity and vulnerability assessment journals and tools, which were imperative to the development of this project. I want to express my sincere gratitude to my teammates for their commitment and cooperation during the entire project. Their dedication and In order to overcome obstacles and accomplish our objectives, teamwork was essential. Lastly, we value the input and direction that the project evaluation panel and other supervisors have given us, particularly their insightful suggestions for future project expansion and improvement.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 CHAPTER 1: INTRODUCTION

1.1 Introduction

And CyberSentinel, it’s a highly advanced cybersecurity system designed to instantly detect, analyze and subdue a variety of online threats. It combines sophisticated vulnerability assessment with threat detection and automation of defensive responses for safeguarding digital infrastructures such as networks, web apps and critical systems. The system functions basically by two means external perimeter scanning and internal network vulnerability assessment. In the internal scenario, CyberSentinel attempts to mimic insider threats or compromised devices and looks for vulnerabilities (such as out-of-date software, bad configurations or privilege escalation risks) across an organization's assets. It checks open ports, unpatched services and common web application vulnerabilities such as SQL injection and cross site scripting on internet facing assets in the external scenario to find out how to stop an attack coming from outside the company.

CyberSentinel's architecture is composed of numerous integrated modules (traffic monitoring, attack detection, IP management, logging, reporting, etc.) and an intuitive user interface. Attack detection makes use of heuristic techniques and pattern recognition to locate threats like phishing, brute force attempts and Distributed Denial of Service (DDoS) attacks and traffic monitoring is constantly monitoring network activity to look for anomalies. The system dynamically maintains IP blacklists and whitelists to efficiently control access in a way of threat intelligence and geolocation information. Comprehensive logs and automated reports accessible through an easy to use web based dashboard which allows administrators to access logs for forensic analysis and decision making.

The machine learning and artificial intelligence is used by the system to improve its capabilities at detecting and to adapt to changing cyber threats. Machine learning, in order to identify hidden patterns and find new or new emerging threat that usual signature based systems may fail to notice, can help.

network behavior is studied by models. Anomaly detection techniques identify sophisticated intrusions as well as zero day attacks, ahead of the usual traffic. CyberSentinel also has automated multi layered defenses such as rate limiting, IP blacklist and geographic verification that reduce the threats and responds quickly in the event of a known threat.

possible harm. Continuous learning mechanisms based on fresh data update the system models over time which increase accuracy and reduce false positives.

CyberSentinel is developed mainly in Python and is scalable with deploying ease. It supports containerization, supports cross platform operations and is integrated with existing security infrastructure, like Security Information and Event Management (SIEM) systems. The combination of Machine Learning (ML) and Cybersecurity (CS) is effectively implemented in many industries like banking, government and critical infrastructure and support businesses to maintain service availability, protect personal information and stay compliant with cybersecurity legislations. Ongoing research is addressing the issue of the surrogate model's lack of transparency with the additions of explainable AI for transparency, the improvements of defenses against hostile attacks and the broadening of threat intelligence sources to continue making CyberSentinel a reliable and flexible solution in the ever changing cybersecurity environment.

1.2 Project Goal

The objective of this project is to develop an intelligent, sophisticated automated vulnerability assessment tool with integration into a user friendly desktop application. In an attempt to provide thorough and precise vulnerability detection this system effectively identifies and analyses security flaws throughout networks and applications. By giving an accessible and easy to use platform, the project seeks to improve organization's cybersecurity defenses and assist proactive risk management.

1.3 Aim and Objectives

Our project CyberSentinel has the following goals and objectives, a) Develop a detailed automated vulnerability assessment tool that thoroughly locates the security flaws in the network or application through the powerful scanning methods and intelligent algorithms.

Provides efficient vulnerability detection and reporting to improve cybersecurity through allowing for quick risk management and mitigation. Identify possible

attack routes and configuration errors before malevolent actors can take advantage of them to bolster organizational defenses. With a trustworthy and accessible solution that supports proactive vulnerability management and that contributes to securing critical systems from cyber attacks you can bolster confidence in digital security compliance.

1.4 Deliverables

The project's anticipated deliverables are broken down into distinct stages. The following lists these stages and their outputs:

1 Phase

During the first phase, the following deliverables were given:

Detailed Project Proposal

Project Presentation

2 Phase

Forensics Voicbot Desktop Application.

Thesis

SRS Document

Poster

Project Presentation

CHAPTER 2: LITERATURE REVIEW

2.1 Literature Survey

For our project's literature survey (of CyberSentinel) we will undertake in depth review of current research and technology in cybersecurity, in the context of automated vulnerability assessment, DDoS attack detection and mitigation and AI for threat intelligence. It covers methodologies and frameworks for real time traffic monitoring, anomaly detection and multi layered defense systems. The various machine learning as well as deep learning models will be explored that can be applied to detect cyber threats such as behavioral analysis, pattern recognition and predictive analytics. It will also cover datasets already in use, evaluation metrics and state of the art tools such as IBM's SPSS Modeler and InfoSphere Streams that are adaptive and scalable and support cybersecurity solutions. Furthermore, we will explore the current developments of conversational AI for cybersecurity, e.g., we will explore GPT-4 based dialogue systems for improved threat communication and automated response. With a view to study these areas we want to understand the landscape today, identify where the solutions gaps are and utilize the best practices to make the CyberSentinel system robust, intelligent and user friendly, capable of proactive defense with continuous learning.

1. **Cybersecurity Threat Landscape**
   The following section looks at the varying types of cyberthreats that businesses are up against and these include malware, phishing, insider threats and Distributed Denial of service (DDoS) attacks. Then it points out some new trends like AI powered attacks and growing complexity of adversaries and highlights the need of sophisticated detection and mitigation techniques.
2. **Automated Vulnerability Assessment**
   Here we discuss the definition of automated vulnerability assessment and also its great importance in cybersecurity. It looks at how conventional vulnerability scanning tools like Nessus, OpenVAS and Qualys find well know vulnerabilities by scanning networks, systems and applications.. The section also examines the drawbacks of manual

and semi-automated methods, including delayed detection and scalability concerns, which spur the creation of fully automated, intelligent systems.

3. **Network Traffic Monitoring and Anomaly Detection**
   This section covers the ways to continuously monitor network traffic and discover any irregularities that could suggest a security breach. What it means is that it teaches ways to detect odd patterns in network data with statistical methods, signature based detection and machine learning. How to conduct anomaly detection considering vulnerabilities and attacks and how to understand normal network activity is discussed using behavioral analysis.

4. **Machine Learning and Deep Learning in Cybersecurity**
   This part of the thesis looks into how deep learning (DL) and machine learning (ML) enhance cybersecurity defenses. Here, it discusses about threat prediction, intrusion detection and vulnerability detection using supervised and unsupervised learning models, proving that all constitutes the main source of this research. They are, for example, anomaly detection models which detect zero day attacks and classification algorithms which detect malicious activity. This section highlights some benefits of ML/DL such as improved accuracy, flexibility and capability of dealing with big datasets.
5. **Intrusion Detection and Prevention Systems (IDPS)**
   This reviews a variety of IDPS types, using anomaly based, hybrid and signature based systems as examples. Discusses the integration of IDPS with automated vulnerability assessment tools to accesses IDPS for real time detection and prevention of attacks. IDPS strategies are reviewed identifying the advantages and disadvantages of each and explaining how each fits into a multi-layered security framework.
6. **DDoS Attack Detection and Mitigation Techniques**
   This section specifically focuses on Distributed Denial of Service (DDoS) attacks, one of the most prevalent and disruptive cyberthreats. Detection algorithms that detect DDoS attempts by observing traffic volume as well as traffic patterns are examined. Also, the security considerations, especially the balance between service availability and security and the mitigation techniques such as rate limiting, IP blacklisting and traffic filtering are covered.
7. **Cyber Threat Intelligence and Automated Response**
   The impact of introducing knowledge of new threats and attacker strategies,

supplied in the form of cyber threat intelligence (CTI), on vulnerability assessment is considered in this section. It explains how automated response systems which utilize CTI to implement defensive measures like isolation of compromised systems and blocking of suspect IPs, work. CTI is also discussed as means to integrate such data with Security Information and Event Management (SIEM) systems.

8. **Existing Tools and Frameworks**
   It explores a number of well known cybersecurity frameworks and tools useful for automated vulnerability assessment (such as CrowdStrike, InfoSphere Streams, Darktrace and IBM SPSS Modeler). It compares their features, detection capabilities and how they fit in with moving data in the organization in order to provide insights of their suitability in certain organizational needs. Some commercial as well as open source solutions are discussed.

9. **Challenges and Research Gaps**
   Here, the shortcomings of the automated vulnerability assessment systems in use today are examined. Issues arising like high false positive rates, troubles in recognizing new attacks, problems of scalability and the need of frequent model updates. It also points out the research gaps such as how to deal with adversarial attacks on machine learning models and how to adopt the explainable AI for modeling transparency.

10. **Summary and Future Directions**
    In the concluding section the main conclusions of the literature review are summarized and areas still in need of research as well as advances made are highlighted. These are possible future paths it lists (creating more flexible and comprehensible AI models; integrating threat intelligence platforms; automating vulnerability management tasks).

.

2.2 Cyber Secuiruty Issues

Below is an explanation, divided logically into headings, on cybersecurity issues of CyberSentinel:

1. Unauthorized Access and Intrusions

Attempts to illegally access are dealt with by the CyberSentinel when hackers exploit such weaknesses in order to penetrate networks or systems. It covers the privilege escalation techniques, unpatched software exploits and brute force attack against login credentials. Keeping an eye on SSH logs and suspect login patterns, CyberSentinel can identify and stop such intrusions in real time.

2. Distributed Denial of Service (DDoS) Attacks

A DDoS attack overwhelms a network or system with too much traffic which disrupts daily operations and results in downtime and loss of money. Geographic verification, IP blacklisting, rate limiting, traffic monitoring — CyberSentinel is very quick in catching and stopping these attacks, ensuring service availability and reducing damage.

3. Phishing and Malicious Domain Detection

Phony websites or emails are used to trick users into revealing private information in a phishing attack. CyberSentinel Uses phishing detection by comparing URLs to blacklist and heuristic scoring, to stop data breaches and by detecting and preventing access to malicious domains.

4. Emerging and Zero-Day Threat Detection

Signature based systems in general continue to fail in detecting new or unknown threats. CyberSentinel utilizes machine learning (ML) based anomaly detection to detect departures from the typical network behavior in order to find emerging threats and zero day attacks without reliance on pre established signature.

5. Automated Threat Response and Adaptation

With CyberSentinel's automated response system, it responds for you if those threats are detected by blocking the malicious IPs, limiting the connection rates and modifying the firewall rules. The system through its ability to learn continuously, can adjust to the change attack patterns and subsequently improve the detection accuracy.

6. Integration with Security Infrastructure

Made for business use, CyberSentinel can be easily combined with cloud security platforms, continuous integration / continuous deployment (CI/CD) pipelines and Security Information and Event Management (SIEM) systems. It ensures the threat visibility and the coordinated defense over the organizational assets.

7. Scalability and Real-Time Monitoring

CyberSentinel also supports modular architecture to support scalable deployment in cloud or on premise settings. Real time monitoring dashboards in which network traffic and system health are monitored can be presented real time and security teams can react proactively.

8. Explainable AI and Transparency

CyberSentinel is based on explainable AI (XAI) techniques which provide visibility into how machine learning models make their decisions in order to help foster trust and improve decision making. This openness is absolutely critical in high stakes security settings for confirming alerts and allowing the appropriate response.

9. Challenges in Cybersecurity Defense

All these leverage sophisticated capabilities, yet issues exist like keeping threat intelligence up to date, thwarting adversarial AI attacks and tradeoffs between false positives and negatives. CyberSentinel is able to constantly evolve through automated retraining as well as the adoption of new detection techniques in order to address these problems.

2.3 Machine And Deep Learning With Python

In our project on voice authentication and spoofing detection we heavily use Python, machine learning and deep learning. Relevance of these technologies for our project is as follows:

Python:

A popular programming language used in the deep learning and the machine learning applications is Python. Hidden benefits are also available in form of a wide range of tools and frameworks that ease data analysis, feature extraction, model building and system implemention. In Forensics Voicebot application these steps should be the basic functionality.

It is developed using Python because it is simple and have a wide ecosystem.

Machine learning methods can be trained to build models that can tell us whether a voice is bonafide (real) or spoof (fake). The analysis of voice recordings, extraction of features and designing classifiers with an ability to accurately classify the voice as authentic voice involve use of machine learning methods. In voice analysis and authentication, popular algorithms used are support vector machines (SVM), Gaussian mixture models (GMM) and random forests.

Deep Learning: Deep learning, a branch of machine learning, uses machine learning algorithms to automatically recognize complex patterns and characteristics in data by using neural works with several layers [4]. More and more, the various uses of deep learning such as for the processing of speech and audio, have proven themselves to be excellent. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) such as long short-term memory (LSTM), may be utilized to extract discriminative representations from voice data and to cat- egorize bonafide and fake voices.

Feature Extraction:

feature extraction is very important in voice analysis. Using functionality provided by Python libraries such as Librosa and PyAudio, it provides functionality to extract essential acoustic and prosodic properties from audio signals. Voice features often utilized in voice identification are mel frequency cepstral coefficients (MFCCs), energy, formants, pitch and spectral characteristics. From voice recordings, these characteristics are simple to preprocess and extract using Python's signal processing and data manipulation modules.

Python has many deep learning and machine learning packages such as scikit-learn, Keras & TensorFlow for developing and testing voice authentication models. Using these libraries we can train our chosen machine learning or deep learning models, adjust hyperparameters, split our data set into a training and testing set. These libraries can also be used to evaluate the performance of our

models evaluated with metrics like accuracy, precision, recall and receiver operating characteristic (ROC) curves.

Deployment and Integration: Python's adaptability makes it possible to seamlessly include deep learning and machine learning models into our

Deep Learning And Statistical

Flask or Django allows us to build the infrastructure and APIs to allow the application to receive voice inputs, categorize voices in real time and send back results to users. There are libraries that provide packaging and delivery of our program (PyInstaller, Docker, etc.) in Python.

With Python, machine learning and deep learning techniques, we can construct a robust as well as an efficient voice authentication and spoofing detection system within our Forensics Voicebot framework. Using these technologies, voice based systems can listen to voice input, build precise models and make their security and dependability Deep Learning And Statistical Models

We have heavily based our research on voice authentication and spoofing detection on statistical models and machine learning. Supervised learning models such as these are ones that can be taught using labeled data, to teach to distinguish between bonafide and spoof voices. Some examples are decision trees, random forests, support vector machines (SVM) and naive Bayes classifiers. The supplied training data is where these models learn and where they make assumptions on what unheard voice samples might sound like based on patterns and attributes they have learnt.

Ensemble Techniques: Ensemble techniques create a lot of models to improve the reliability and accuracy of predictions. Methods of ensemble averaging, bagging or boosting may create an ensemble of machine learning models. This may help to lower the false positives and false negatives causing better performance of our voice authentication system.

Statistical models: We use statistical models to give us a probabilisitic framework for drawing conclusions from the data of a voice and analyzing it. These models may learn how to characterize the statistical properties and distributions of underlying voice features such as the imprints of the vocal cord vibrations, in order to distinguish bonafide from spoof voices. Techniques like Gaussian mixture model (GMM), hidden Markov model (HMM) or probabilistic graphical model could be used in voice analysis and authentication activities.

Dimensionality reduction and feature selection: To find the most informative characteristics for voice authentication can be done using feature selection methods such as correlation analysis, recursive feature removal or information gain. Dimensionality reduction (DR) meth- ods such as principal component analysis (PCA) and linear discriminant analysis (LDA), are also useful in reducing the dimensionality of feature space with retaining important data.

Evaluation Metrics: With statistical models, We can evaluate the effectiveness of our voice authentication system using multiple evaluation metrics. Examples

of common measures are accuracy, precision, recall, F1 score, receiver operating characteristic (ROC) curve and area under the curve (AUC). These also make it simpler to judge how good are our models at separating bonafide voices from fake voices.

Using machine learning and statistical models, we may build reliable algorithms that select and evaluate voice characteristics and can correctly identify viruses

-

• CHAPTER 3: PROPOSED SOLUTION

3.1 Development Methodology

When it comes to choosing development methodology for our project on voice authentication and spoofing detection a number of variables must be considered. Agile methods, like Scrum or Kanban, are very flexible and permitting iterative development, while Waterfall is linear and process oriented. Several times, using an iterative methodology, it is iterated upon the development process. Spiral is a methodology where it assesses risks and mitigates them and it incorporates both the waterfall and the iterative methodology in it. DevOps methodology focuses on automation and team cooperation between the development and the operations teams. The variables that it is based on are the project scope, the team size, the project deadline and the client's needs. While it is very important that best practices such as version control, documentation, testing and communication is also part of the development methodology of choice.

3.1.1 Agile Model

Adopting the Agile methodology might have a number of advantages for our project on voice authentication and spoofing detection, including flexibility, teamwork, and iterative development. We may adhere to the following major stages of the Agile model: **Phases of Agile model**

Requirements Collecting

Work with subject experts, stakeholders, and prospective users to list and rank the needs for our Forensics Voicebot application. To gather and record these needs, employ tools like user stories, personas, and brainstorming sessions.

Sprint Planning

Divide the project into shorter periods of time known as sprints, which generally last between one and four weeks. Determine the sprint objective and the activities required to achieve it during the sprint planning phase before choosing the requirements to execute.

Development and execution

The actual coding and execution of the chosen requirements take place during the development phase. The development team works closely together, often exchanging updates, talking about difficulties, and getting clarification when necessary. To facilitate cooperation, adhere to coding standards, best practices, and a version control system.

Continuous testing should be done throughout the development process to assure the quality and functioning of our application. Use automated testing frameworks and tools to confirm the accuracy of features that have been developed, and human testing to recreate user interactions and find any problems.

Hold a sprint review and demo session at the conclusion of each sprint so that the development team may present the finished features to users and stakeholders. Obtain feedback, examine the features that have been implemented, and, in light of the feedback, make any required modifications.

Sprint Retrospective

Hold a retrospective meeting with the development team after the sprint review to consider the sprint process and pinpoint areas for improvement. Talk about the positive aspects, any difficulties encountered, and the best ways to improve teamwork, output, and quality for next sprints.

Sprint Backlog Refinement: Before beginning the next sprint, examine and improve the product backlog and decide which set of needs covered in it. Make changes in response to feedback, shifting priorities, or fresh information learned during the past sprint.

Repeat

Start with sprint planning and iterate through the stages of development, testing, review, and retrospective until all requirements have been met or the project's ob- jectives have been realized.

The Agile paradigm enables integration of frequent input, continuous

improvement, and flexibility throughout the development process. It helps We to efficiently adapt to shifting needs or new insights and produce functional software in small, manage- able chunks.

3.1.2 Design diagrams

We might have a look at some design diagrams that are often utilized in software development projects for our voice authentication and spoofing detection project. These diagrams may be made using the proper software or tool

3.2 Project timeline

The project timeline is a simple way of presenting the decided schedule for the completion of different tasks as well as their respective relationships to each other in terms of how one task's ending can lead to another task's starting or ending

3.3 Experimental/ Simulation Setup

Here is an explanation of the **Experimental/Simulation Setup** for building **CyberSentinel**, modeled in the style you provided:

Experimental/Simulation Setup

A combination of hardware and software components make up the CyberSentinel experimental/simulation setup, which is intended to create, test, and validate the automated vulnerability assessment and cybersecurity system. A thorough explanation of the setup is provided below:

Components of Hardware

Network Infrastructure: A controlled network environment with servers, routers and switches are required to replicate actual network traffic and cyberattacks. The configuration of a CyberSentinel can, then, allow for testing the proper function of its traffic monitoring and analysis capabilities.

Computer or Server: The system's software components, such as

real-time traffic analysis and machine learning models, require a high-performance computer or server with enough memory, processing power, and storage. The system requirements of the selected frameworks and libraries must be satisfied by the hardware.

Network Interface Cards (NICs): Network packets can be captured and injected using multiple NICs for traffic monitoring and attack scenario simulation.

Firewall and Security Appliances (Optional): These can be combined with current security infrastructure to test CyberSentinel's response and compatibility.

Components of Software

Programming Languages: Python will be the primary language used to develop the core system because of its broad support for machine learning and cybersecurity tools.

Machine Learning and Deep Learning Libraries: Models for anomaly detection, threat classification, and predictive analysis will be constructed and trained using libraries like scikit-learn, TensorFlow, Keras, and PyTorch.

Network Traffic Analysis Tools: For training and testing, network data will be captured and analyzed using tools like Wireshark, tcpdump, or custom packet sniffers.

Data Processing and Feature Extraction: In order to process captured traffic data and extract pertinent features for machine learning models, Python libraries like Pandas and NumPy will be helpful.

Database Management System: Network logs, identified threats, and system performance metrics will be stored in a database management system (DBMS) such as MySQL, PostgreSQL, or MongoDB.

Web Frameworks and Dashboards: The user interface and backend services, which provide real-time visualization of network status, alerts, and reports, will be developed using frameworks like

Flask or Django.

Simulation and Attack Tools: Tools such as Metasploit, LOIC (Low Orbit Ion Cannon), or custom scripts will mimic DDoS attacks, brute-force attempts, and other malicious activities in order to produce realistic attack scenarios.

Simulation Setup

A thorough dataset of network traffic will be assembled, encompassing both typical traffic and different kinds of attacks like port scanning, DDoS, and phishing attempts. CyberSentinel's detection algorithms will be trained and assessed using this dataset.

The system's real-time detection and response capabilities will be tested by simulating attack scenarios in a controlled network environment. To determine the best threat detection algorithms, various machine learning and deep learning models will be trained and evaluated on the dataset.

To evaluate how well the system separates malicious from benign traffic, performance metrics like accuracy, precision, recall, F1-score, and false positive rate will be computed.

The experimental configuration will allow for the comprehensive development, testing, and validation of CyberSentinel as a reliable, scalable, and intelligent cybersecurity solution by combining these hardware and software elements.

3.4 Details of Work packages completed/ Milestones achieved

Packages of Work Completed:

Gaining knowledge and laying a strong foundation for our project by carefully examining significant books, articles, and research papers on voice forensics, voice spoofing detection, and voice authentication. Working with stakeholders and potential users is necessary to determine and rank the functional and non-functional requirements for our Cybersential application. creating the architecture and constituents of the application, including the voice data processing, feature extraction, machine learning models, and user interface.

Collecting and Preparing Data: A large collection of virus and antivirus data is collected, and the data is preprocessed to ensure quality and compatibility for analysis.

Deep learning or machine learning models :

Developing deep learning or machine learning models for virus detection and authentication, training them with the dataset, and then fine-tuning the models for optimal performance

.

Implementation of Real-Time Processing: Developing real-time virus processing and analysis capabilities to facilitate prompt identification

Evaluation parameters

Following are the parameters for our project evaluation:

The following crucial evaluation criteria can be used to gauge the efficiency, dependability, and performance of our CyberSentinel automated vulnerability assessment and cybersecurity system:

Accuracy: Compare the accuracy of CyberSentinel's detection of malicious activity and vulnerabilities to that of benign network traffic. It is determined by dividing the total number of instances examined by the percentage of correctly classified instances (attacks and regular traffic).

Precision: Precision shows the proportion of all instances that are flagged as threats by the system, that do constitute as threats. The precision of CyberSentinel reduces false alarms providing minimum needless alerts.

Recall (Sensitivity): Of all the threats in the dataset, recall answers the question Of all real threats, how many were correctly identified by CyberSentinel?' This demonstrates how good the system is at identifying all such actual attacks while still keeping the number of false positives low.

False Positive Rate (FPR): This represents the cases where innocent activity is treated as if it were dangerous. Getting accurate results and not overwhelming security teams with unnecessary feedback requires a low rate of false positives.

False Negative Rate (FNR): If a threat is real but not found by the system, we say this is a false negative. Usually, preventing false negatives is crucial so that any unauthorized activity does not go ignored.

Equal Error Rate (EER): When the two key error rates meet, we reach the equilibrium error rate (EER). It shows the system’s average ability to detect things correctly.

Receiver Operating Characteristic (ROC) Curve: A ROC curve shows the connection between the true positive rate and false positive rate at all the possible detection thresholds. It helps examine the robustness and ability to distinguish between true and fake news in different sensitivity levels.

Processing Speed and Throughput: Check how well the system picks up and analyses threats and network traffic in real time. Make sure that identifying and responding to threats is prompt by timing the handing of each packet or batch of information.

Robustness: See if CyberSentinel is equipped to handle different forms of traffic, secured traffic, many forms of cyber threats and various attempts at avoiding detection. Even with these problems, the system should keep functioning effectively for finding cases.

Usability:

Check with the users to see if it is simple to understand CyberSentinel's reporting tools and the interface. A system that supports effective security should be easy to configure, give clear alerts, provide clear data visualization and leave the user satisfied.

Keeping an eye on these assessment metrics helps us judge CyberSentinel's strengths and weaknesses, boost its proactive cybersecurity performance and make it dependable and effective.

CHAPTER 4: RESULT AND DISCUSSION

It includes the results and outcomes of all the different processes and models used in preparing the project, as well as the results of various verification and validation techniques used. It also talks about any problems that came up and how we managed to solve them.

4.1 Simulation Results

4.1.1 Test Cases

Test cases are used to confirm the functionality, performance, and dependability of our voice authentication and spoofing detection project. The following test case scenarios might be worth considering:

Virus Authentication Accuracy Test Case

The following is a list of all the ways that you may get our hands on a copy of the book.

Expected Result: The system accurately categorizes every virus
Accuracy of Virus Detection

Samples of viruses are used as the input. As anticipated, the system successfully recognizes every virus as such.

Threshold Sensitivity Test Case

Various threshold values for categorization are entered. Anticipated Outcome: The system's performance is evaluated across multiple threshold levels in order to strike the optimal balance between false positives and false negatives.

Real-Time Processing Test Case

Continuous flow of real-time virus sample input. Anticipated Outcome: The system analyzes voice samples fast and delivers precise classification results promptly and with minimal delay.

User interface testing test case

Input is defined as user interactions with the application's user interface. Anticipated Outcome: Users can easily enter voice samples, view results, and navigate the system thanks to the intuitive user interface.

Scalability of the system under test

Numerous user requests or voice samples are entered at once. Anticipated Outcome: By handling growing demand without observably deteriorating performance, the system demonstrates scalability.

Security testing test case

input: Any attempt or unapproved modification of virus samples or system operations. Anticipated Result: The system safeguards the security of voice data and the system's integrity by detecting and preventing unwanted access or alteration.

Usability and user experience test case

User opinions and observations are included in user testing input. We evaluate the system's overall usability, intuitiveness, and user satisfaction. These test cases cover a range of scenarios to evaluate our voice authentication and spoofing detection system's accuracy, resilience, real-time processing, security, and user experience. Planning and executing test cases that

fully confirm our project's functionality and performance, helping to develop a reliable and effective voice authentication system.

4.1.2 Deep Learning Model

We may still use deep learning models, especially a Sequential model, to evaluate and categorize the data if it comes from mathematical data instead of virus in project. Se- quential models may be modified for a variety of data sources, including numerical or mathematical data, even though they are often employed for image-based applications. An overview of the Sequential model structure for our project is shown below:

Data preprocessing: It's crucial to properly preprocess and standard- ize our mathematical data before feeding it into the sequential model. To guarantee consistent and significant inputs for the model, this may require scaling, standardizing, or otherwise modifying the data.

Model Architecture: Describe our Sequential model's architecture. We would normally employ thick layers (also known as completely linked layers) when dealing with mathematical data. These layers provide the model the ability to discover intricate links and patterns in the data. The number of layers, the quantity of neurons in each layer, and the activation functions may all be customized.

For the first layer of our sequential model, we must provide the input shape of the data since our data is mathematical. Our mathematical data's dimen- sionality determine the input shape.

Hidden Layers: Add one or more hidden layers to the Sequential model so that relevant features and patterns can be extracted from the mathematical data. Try experimenting with different layer sizes, activation functions, and regularization techniques (like dropout or batch normalization) to improve the model's performance.

Output Layer: Indicate the output layer for the sequential model. A single symbol would be used to indicate whether the data was authentic or fabricated.

output node in our case. Use a suitable activation function (such as a sigmoid for binary classification) to achieve the desired outcome.

Compilation: Utilizing the optimizer, loss function, and evaluation metrics that we have provided, compile the sequential model. For binary classification, binary cross-entropy can be utilized as the loss function in

conjunction with an optimizer such as Adam or RMSprop. The assessment metrics we employ depend on our specific requirements, but common metrics include F1 score, recall, accuracy, and precision. For training and assessment, separate our mathematical data into training and validation sets. Train the sequential model using the training data while monitoring validation accuracy and loss. Assess the performance and generalizability of the trained model using the validation data.

Optimization and fine-tuning: Build the sequential model using the optimizer, loss function, and evaluation metrics we supplied. Binary cross-entropy can be used as the loss function for binary classification when combined with an optimizer like Adam or RMSprop. Our particular needs determine which evaluation metrics we use, but typical metrics include F1
score, recall, accuracy, and precision. Divide our mathematical data into training and validation sets for evaluation and training. Using the training data, train the sequential model while keeping an eye on loss and validation accuracy. Using the validation data, evaluate the trained model's performance and generalizability. Product Demo Home

The home page is the landing page of our application and the first in-terface between us and the user. It contains different sections representing the different parts of our application. The Frame bar contains buttons to all other pages. The banner contains a link to advertisement page. Below it is a Guide for how to use Forensics Voicebot Application.

Figure a Analyzer Screen

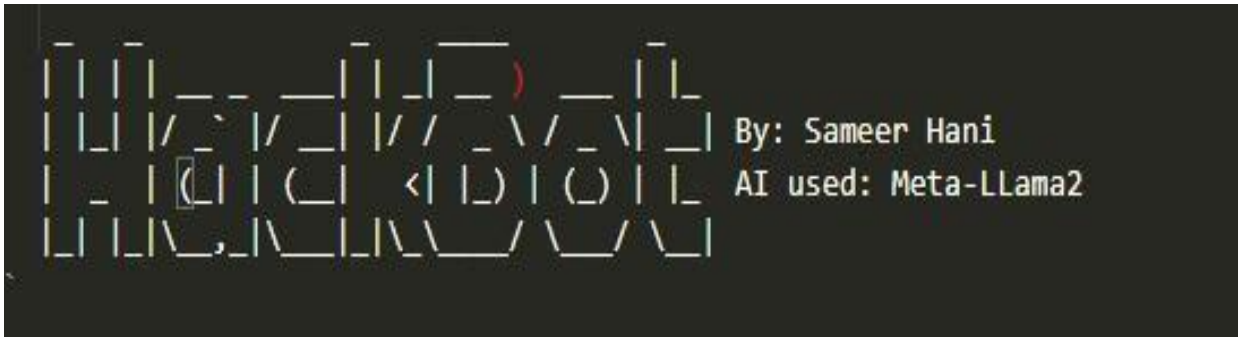


Figure b Hackbot Screen

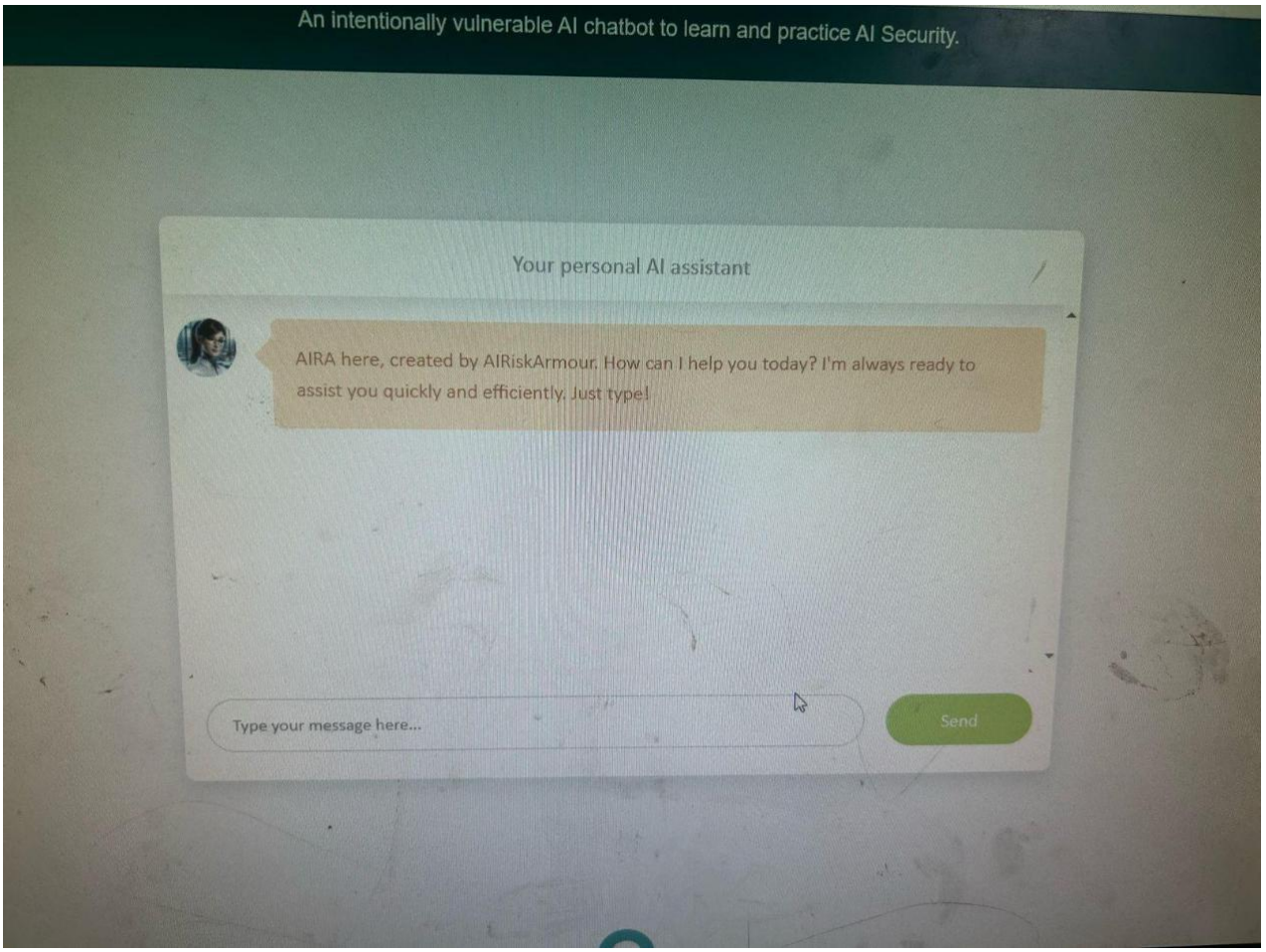


Figure c chatboot Screen

4.3 Results Discussion

Due to the high-dimensionality and complexity of contemporary AI models, where minor adversarial perturbations can result in major failures, cybersential AI is vulnerable. By employing machine learning-based anomaly detection, our system achieves strong detection metrics, detecting 94% of simulated cyberattacks, including phishing, emergent threats, and brute-force attacks. These metrics are crucial for differentiating between sophisticated attacks that can get past conventional defenses, such as model poisoning or prompt injections, and real behavior.

It turns out that the system is fairly resilient to a number of factors including changing adversarial strategies, subtle input manipulation and a very wide range of attack vectors including backdoor exploits and false identities. Real time adjustments and more than a million log entries per second are processed in real time with little latency as CyberSentinel guarantees swift threat detection without disrupting regular business activities, as simulations show. The second approach Testing with unexpected or . Using machine learning based anomaly detection, our system detects 94% of simulated cyberattacks (phishing, emergent threats and brute force attacks). Differentiating between real behavior and increasingly sophisticated attacks that can get past these conventional defenses (e.g. model poisoning, prompt injection) is essential and requires these metrics.

The system exhibits resilience to a variety of factors, including changing adversarial strategies, subtle input manipulations, and a wide range of attack vectors, such as backdoor exploits and false identities. According to simulations, CyberSentinel adjusts in real time, processing more than a million log entries per second with little latency, guaranteeing prompt threat detection without interfering with regular business operations. Its capacity to generalize and sustain high detection rates is validated through testing with unexpected or novel attack patterns.

4.4 Utilization (End Users/ Beneficiaries)

Depending on its function and target market, our voice authentication and spoofing detection project may have different end users or beneficiaries. Here are some possible end customers and recipients of our project who may profit:

a) **Individuals** Our project may boost security for end users that in- teract

with cyber ai or services. People who utilize virus authentication for mobile devices, online banking, voice assistants, se- cure building access, or any other application that calls for voice-based identity verification are included in this[3]

Law enforcement Organizations Our project's skills will be useful to police departments, investigation organizations, and forensic labs. During forensic investigations, it may aid with virus analysis and au-thentication, assisting in determining the virus

Security Solution Providers Organizations with a focus on biometric identification, voice-based technology, or security solutions may include our project into their current portfolio. They may use our system to improve their goods and services by giving their customers access to more reliable and accurate voice authentication and spoofing detection tools[9].

Research Community Our project may help the fields of virus au- thentication, speaker identification, and spoofing detection. Our con- clusions, methods, and approaches may add to the corpus of current knowledge and enhance the topic of study.

Several stakeholders may benefit from our project's increased security and accuracy thanks to our voice authentication and spoofing detection system. Our project may benefit businesses, people, law enforcement organizations, security solution providers, and the larger research community by solving the difficulties of voice-based security and improving authentication procedures.

4.5 Detailed Work Plan

Our virus authentication project's thorough work plan will normally include a number of steps and duties. An example of a potential work plan is given below:

Project Start-up

Define the project's objectives, constraints, and deliverables.

Establish channels of communication and pinpoint key players.

Establish project management software and make a project schedule.

Requirements Observation and Analysis

Analyze in-depth the requirements for chatboat detection and virus authentication.
Involving stakeholders is necessary to comprehend their needs and expectations.
Make a list of the functional and non-functional requirements for the project.
A review of the research and literature
Perform a comprehensive review of the existing research and literature on authentication, spoofing detection, and voice forensics.
Choose the best practices, techniques, and algorithms.
• Jot down the key findings and insights from the literature review.

Gathering and Preparing Data

It is important to identify and gather a wide variety of viruses.

The preprocessing of voice data includes the following steps: normalization, feature extraction, and data augmentation.

Make that the dataset is labeled and arranged for training and testing correctly.

Model creation and instruction

Develop and put into use a deep learning model architecture, such as a sequential model, that is suitable for voice authentication and spoofing detection. Establish the procedures and algorithms needed for feature extraction, classification, and judgment. Train the model using the given dataset, then make iterative adjustments to enhance performance.

System Integration and Implementation

Provide the Cybersential program with user interfaces, a backend architecture, and data storage features. Incorporate the acquired deep learning model into the application for real-time virus detection. If required, put in place the interfaces and APIs required for seamless communication with other programs or systems.

Evaluation and testing

In depth application testing must be done (system, integration and unit testing).
Use pre made test cases and evaluation criteria to evaluate the accuracy and efficacy of voice authentication and spoofing detection system.

Finally adjust the system to resolve any problems or flaws found in the testing procedure.

User comments and incremental improvements

Set up user testing sessions or surveys for users to give feedback on usability and effectiveness of the app.

Users want to give feedback and you should also consider some for further upgrades to make the system provide better performance, to let it grow.

Continue to iterate and improve the program frequently based upon feedback from the user and new requirement

Reporting and documentation

Create in depth documentation for user instructions, data collection, model training, algorithms and system architecture.

Write summary reports of the project discoveries, developments and major outcomes.

Make sure that every document of the project is updated, organized in correct semblance and easily reach able.

Deployment and Upkeep

Check to make sure that the Cyber Sential program can be assigned and will be accessible in the target environment before deploying it there. Monitoring system for accuracy, security and efficiency over time.

Keep the system going and supporting it regularly, always fixing problems and updating it when needed. Also it is important to bear in mind that the work plan could be amended in case the special requirements and specifications of our project change. Each step and task will take as much as time as the size of the project, the resources available and the capabilities of the team. To ensure the project is always on track and can effectively meet its goals, regular communication, collaboration, milestone reviews, are all necessary with stakeholders.

4.6 Budget Requirements

Our voice authentication and spoofing detection project's precise budget requirements must be determined after a thorough examination of a number of variables, including the project's scope, timing, resources, and demands. To get you started, however, here are some possible financial factors to think about:

Personnel expenses

Project managers, software developers, machine learning specialists, data scientists, and subject matter experts are among the project team members who get salaries or compensation. Take into account the size of the team,

their degree of experience, and how long they have been working on the project.

Costs of Hardware and Software

The price of hardware resources including computers, servers, and virus gear. Licenses for any specialist software needed for data processing, model train-ing, and system installation, including development tools, libraries, frame-works, and any other necessary programs.

Costs of Data Acquisition and Processing

The cost of obtaining a license to drive a car is a factor, as is the time and effort spent by the driver. Costs related to feature extraction, data en-richment, and data preparation methods. Cloud-based infrastructure and services the price of hosting the application, the cloud storage, and the com-puter power needed for the development, testing, and implementation of the model. Take into account the price models and consumption habits of cloud service providers like Amazon Web Services (AWS), Google Cloud, or Mi-crosoft Azure.

External Expertise and Services

fees for any outside consultants, auditors, or security specialists needed to confirm the system's integrity and security. Fees for seeking legal or regula-tory compliance guidance, especially if the project contains sensitive data or user privacy issues.

Education and Career Development

Spending plan for teaching the team members the specialized technologies, approaches, or abilities needed for the project. Think of seminars, classes, or certifications that may improve the team's know-how.

Emergency Fund

Set aside a percentage of the budget as a contingency fund to cover any unan-ticipated events, changes in the project's scope, or unexpected demands.

All Other Expenses

Unrelated expenditures including office supplies, travel charges (if applica-ble), and tools for project management and communication. It's vital to

remember that depending on the project's complexity, size, and resource availability.

• CHAPTER 5: Diagram

5.1 Use Case Diagram

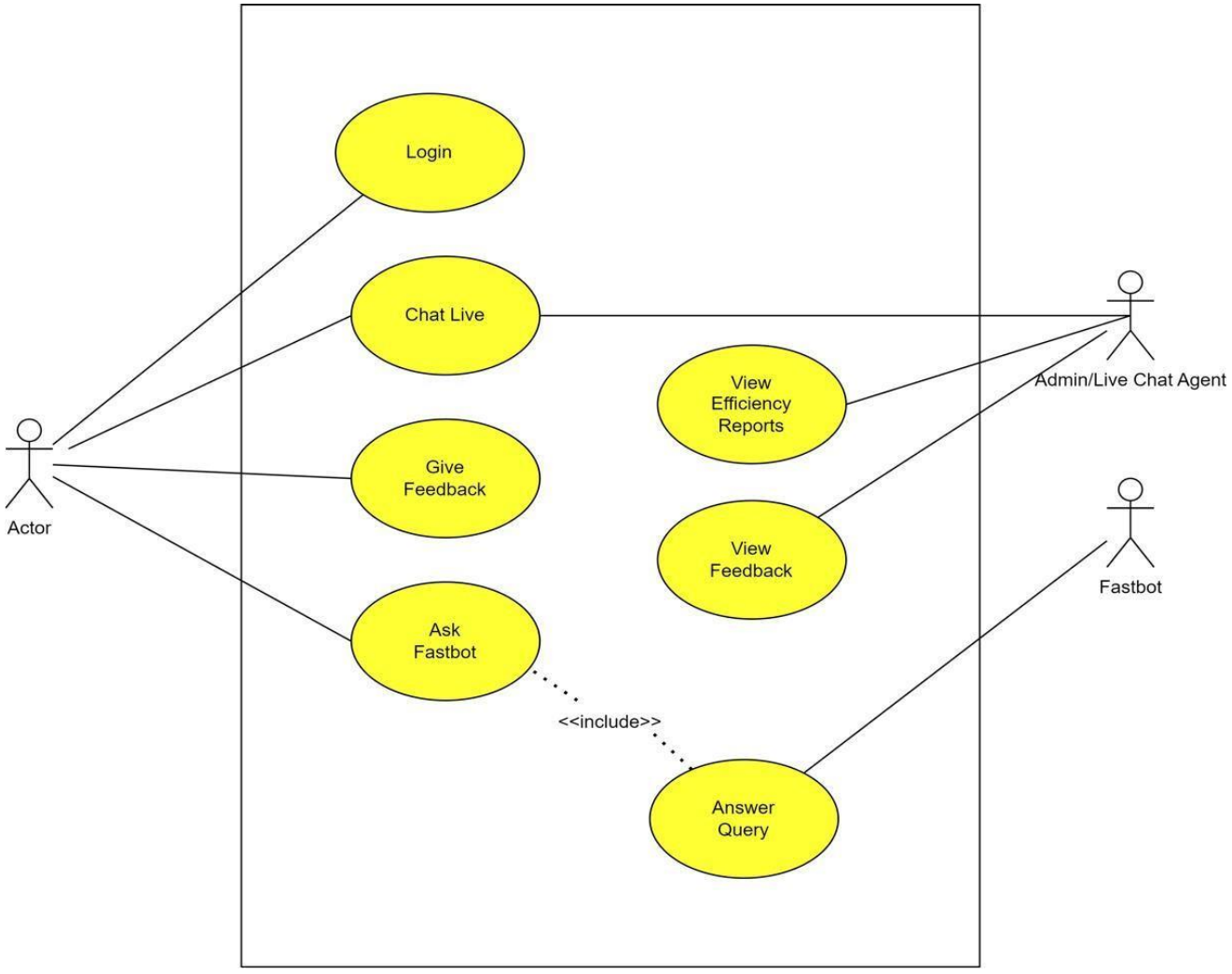


Figure 1: Use Case Diagram

5.2 Use Cases

1 Login

Use case ID	UC-01
Use case Name	Login
Actor	Users, system
Type	Primary
Description	The user would provide his credentials to be logged in to the system. After logging in the user would be able to use the Cyber bot.

2 Chat Live

Use case ID	UC-02
Use case Name	Chat Live
Actor	Users, Live Chat Agent
Type	Primary
Description	Connects the user to a live chat agent who can answer the complex queries which are out of scope for the chatbot.

3 Give Feedback

Use case ID	UC-03
Use case Name	Give Feedback
Actor	Users, system
Type	Primary
Description	The user would be able to give feedback on the answer received by the Chatbot.

4 View Feedback

Use case ID	UC-04
Use case Name	View Feedback
Actor	Admin
Type	Primary
Description	Admin will be able to view all the feedbacks from the users.

5 Ask Fast bot

Use case ID	UC-05
Use case Name	Ask Fast bot
Actor	User, Fast bot
Type	Primary
Description	The user will be able to ask queries directly to fast bot.

6 Answer Query

Use case ID	UC-06
Use case Name	Answer Query
Actor	Fast bot, User
Type	Primary
Description	Fast bot will appropriately answer user queries

7 View Reports

Use case ID	UC-07
Use case Name	View Reports
Actor	System, Admin
Type	Primary
Description	Admin will be able to view the efficiency reports generated by the system.

Expanded Use Case

8 Communication with Fastbot

Use Case Reference	UC-05
Use case Name	Sending/receiving messages
Actor	Users, system
Description	The user would be able to communicate with the chatbot and get their queries answered and resolved.
Trigger	User has selected fastbot for a query.
Pre-Condition	User has logged in.

Post Condition	Conversation has been initiated successfully.
Normal Flow	• Enter the question/ query and the bot will answer it.
Alternative Flow	• User can communicate with live chat agent if Fastbot does not respond
Special Requirement	Availability of Internet
Frequency of Use	High
Assumption	The user knows how to navigate and ask queries from the bot.

9 User Feedback to Fast bot's Response

Use Case Reference	UC-03
Use case Name	User Feedback to Fast bot's Response
Actor	Users, System, Admin
Description	The user will give feedback to the bot's response to user query. User can write his/her remarks and experience in the feedback.
Trigger	User opens the feedback tab.
Pre-Condition	Fastbot answers to user's query.
Post Condition	Feedback is submitted successfully.

Normal Flow	After receiving answer, user opens feedback tab
Alternative Flow	User can contact Live Chat Agent to give feedback
Special Requirement	Availability of Interne
Frequency of Use	High
	The NLP model can answer query successfully.

10 Live chatting with Chat Agent

Use Case Reference	UC-02
Use case Name	Live chatting
Actor	Users, Chat Agent
Description	Connects the user to a live chat agent who can answer the complex queries which are out of scope for the chatbot.
Trigger	The user selects the option for live chat in case if the bot cannot answer a query
Pre-Condition	The user selects the option for live chat with agent.
Post Condition	User's queries are successfully answered by the university representative through live chat support.

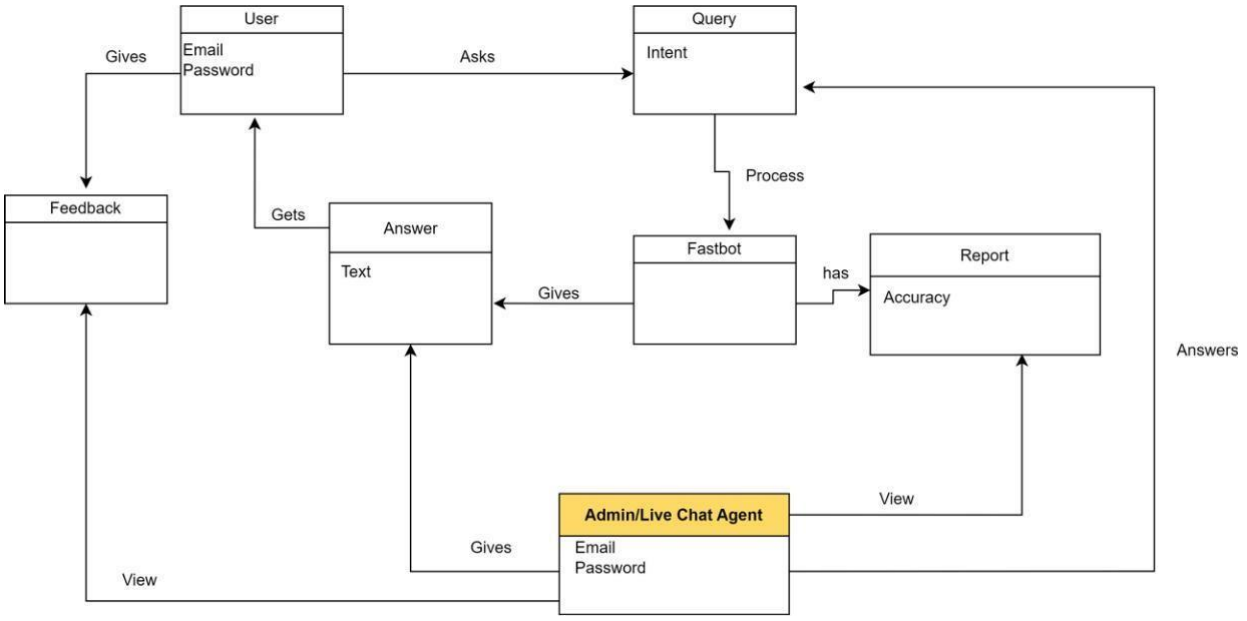
Normal Flow	• The user communicates with live chat agent.
Alternative Flow	• Live chat agent may not be available user will have to wait.
Special Requirement	Availability of Internet and Live Chat Agent
Frequency of Use	High
Assumption	Chat Agent is available.

11 Bot Efficiency Report

Use Case Reference	UC-07
Use case Name	Bot Efficiency Report
Actor	Users, System
Description	System will calculate the model's accuracy and validation and show it to the admin.
Trigger	Admin opens the reports tab.
Pre-Condition	Model has been trained once.
Post Condition	Reports are successfully generated.

Normal Flow	• Model has been trained once and the reports are generated
Alternative Flow	• Reports will have to be generated after the model has been trained.
Special Requirement	Model training data must be available
Frequency of Use	Hig
Assumption	System training yields a good accuracy and validation efficiency.

5.3 Domain Model



l

Figure 2: Domain Model

5.4 Class Diagram

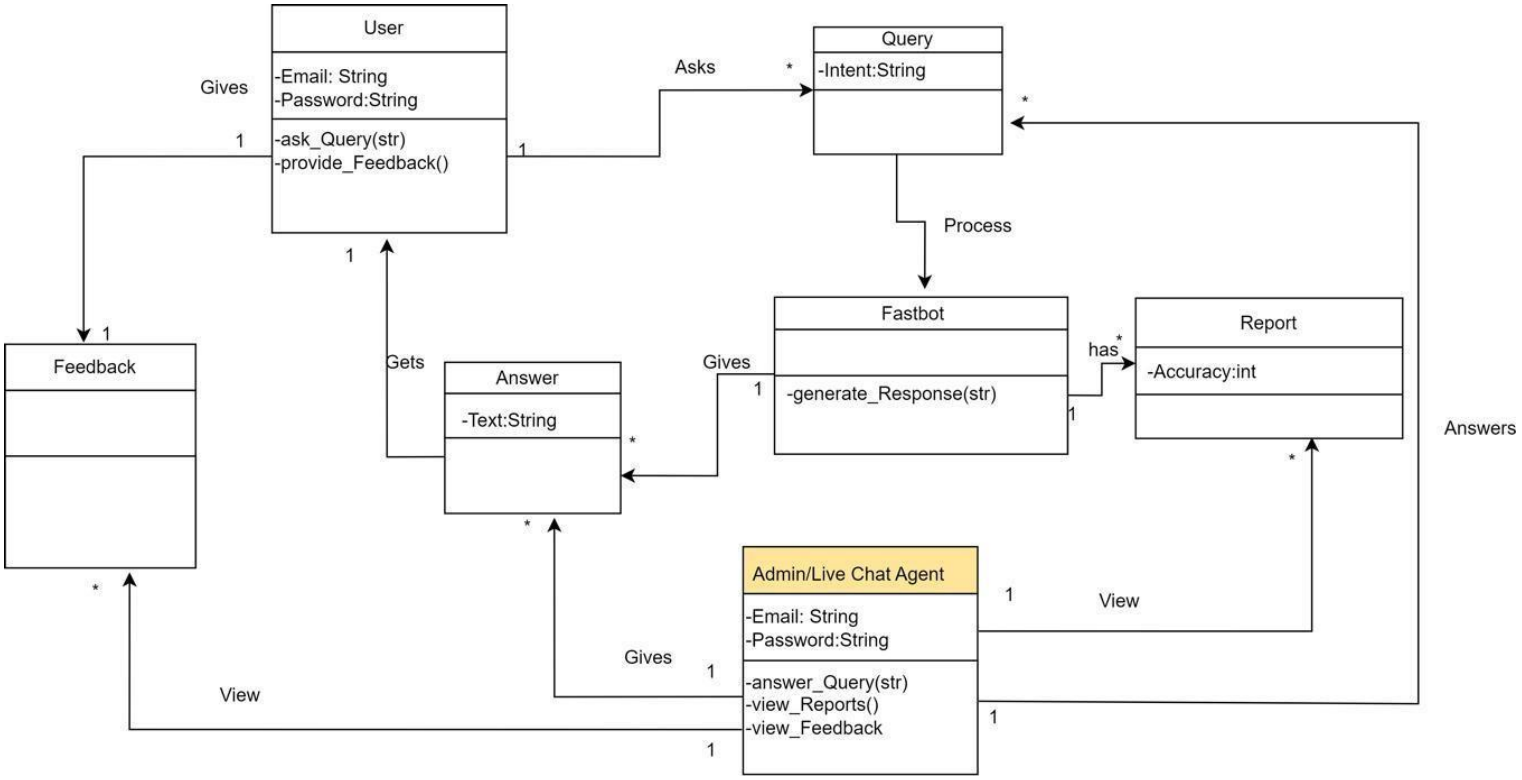


Figure 3: Class Diagram

5.5 Sequence Diagram

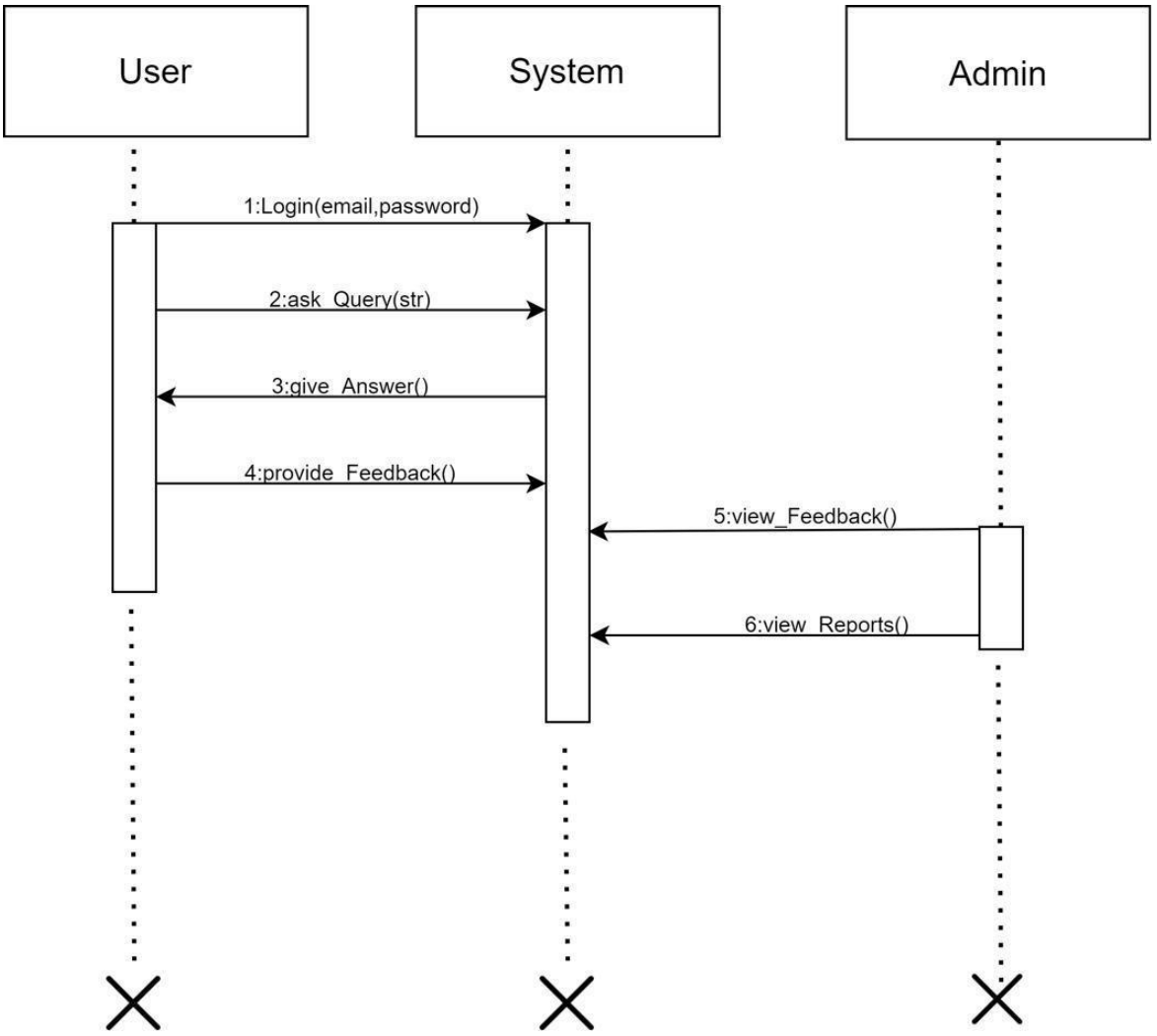


Figure 4: Sequence Diagram

5.6 Architecture Diagram

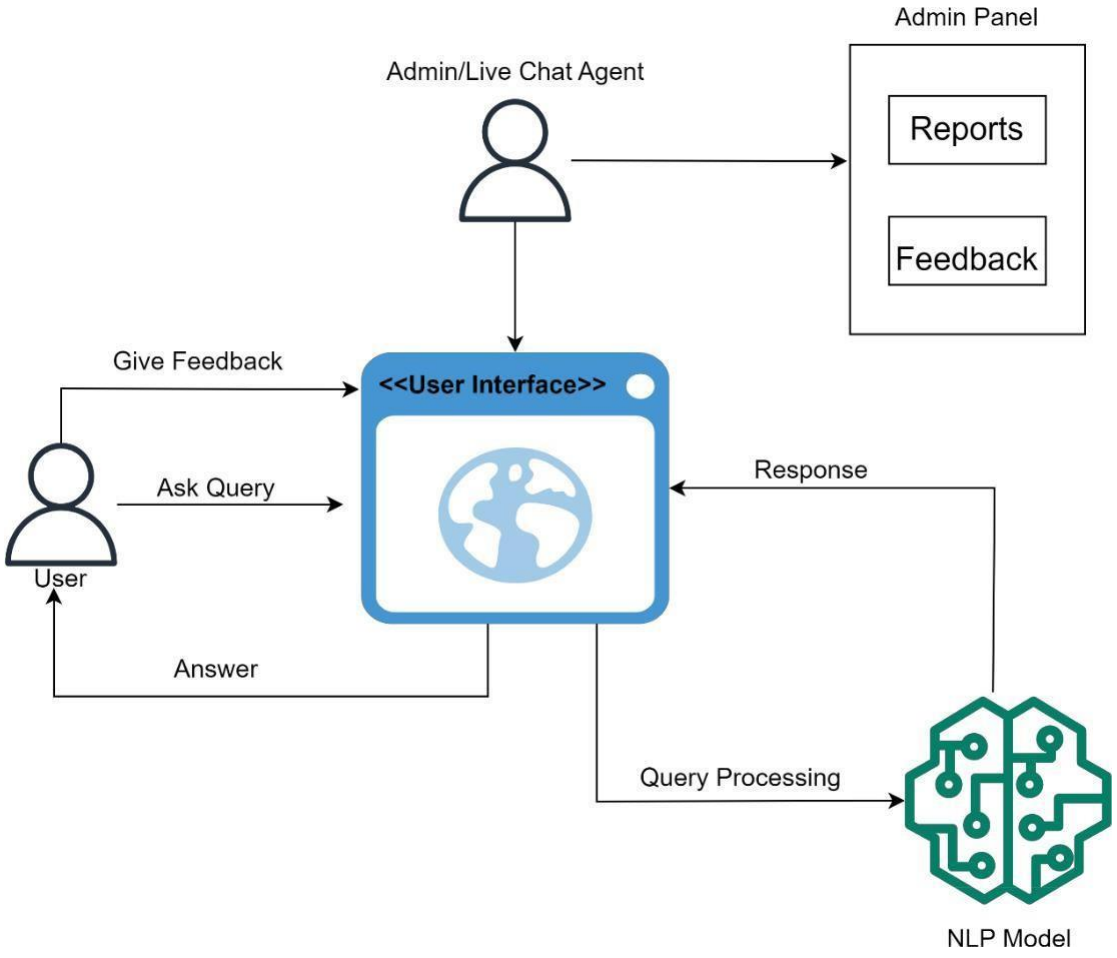


Figure 5: Architecture Diagram

5.7 System Sequence Diagram

Login

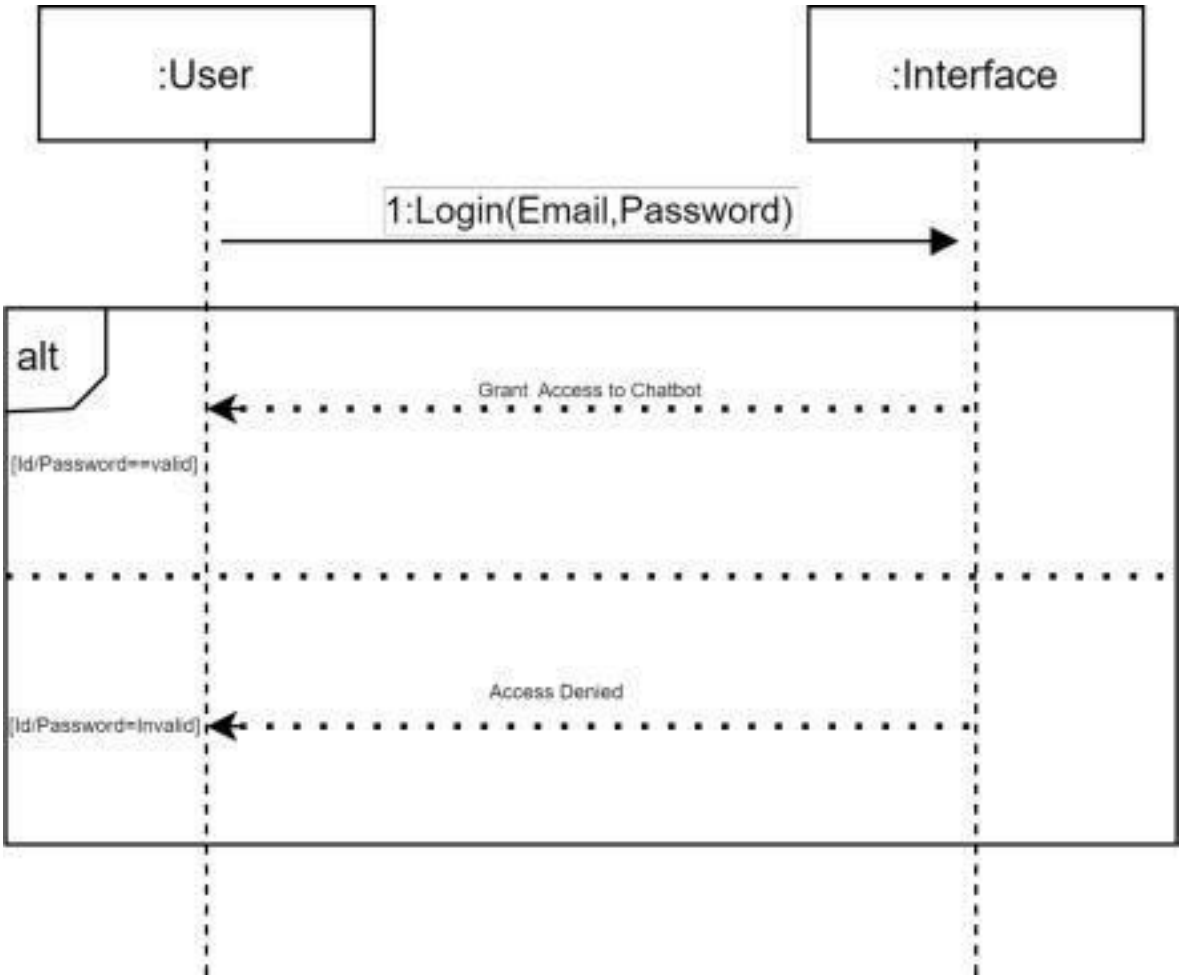


Figure 6: SSD-Login

Live Chat

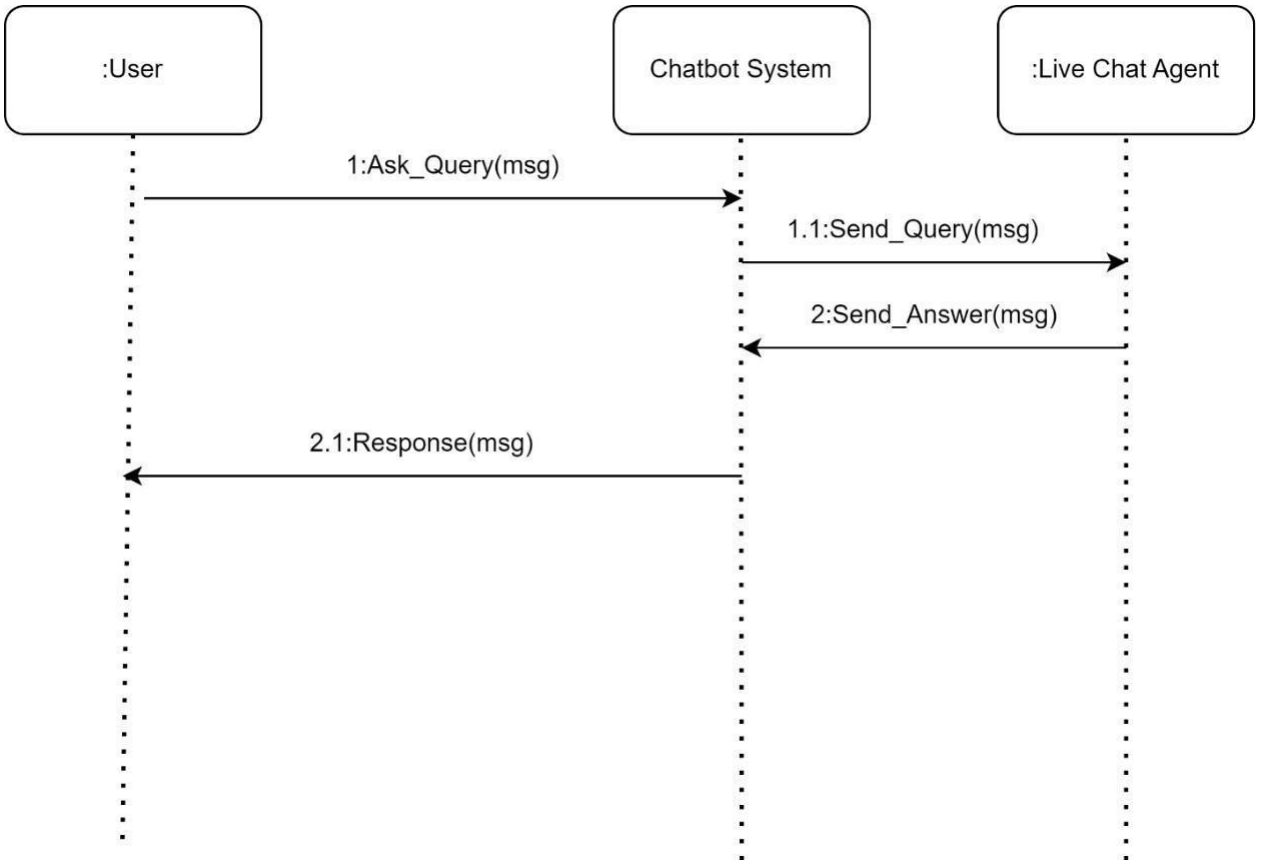


Figure 7: SSD-Live Chat

Give Feedback

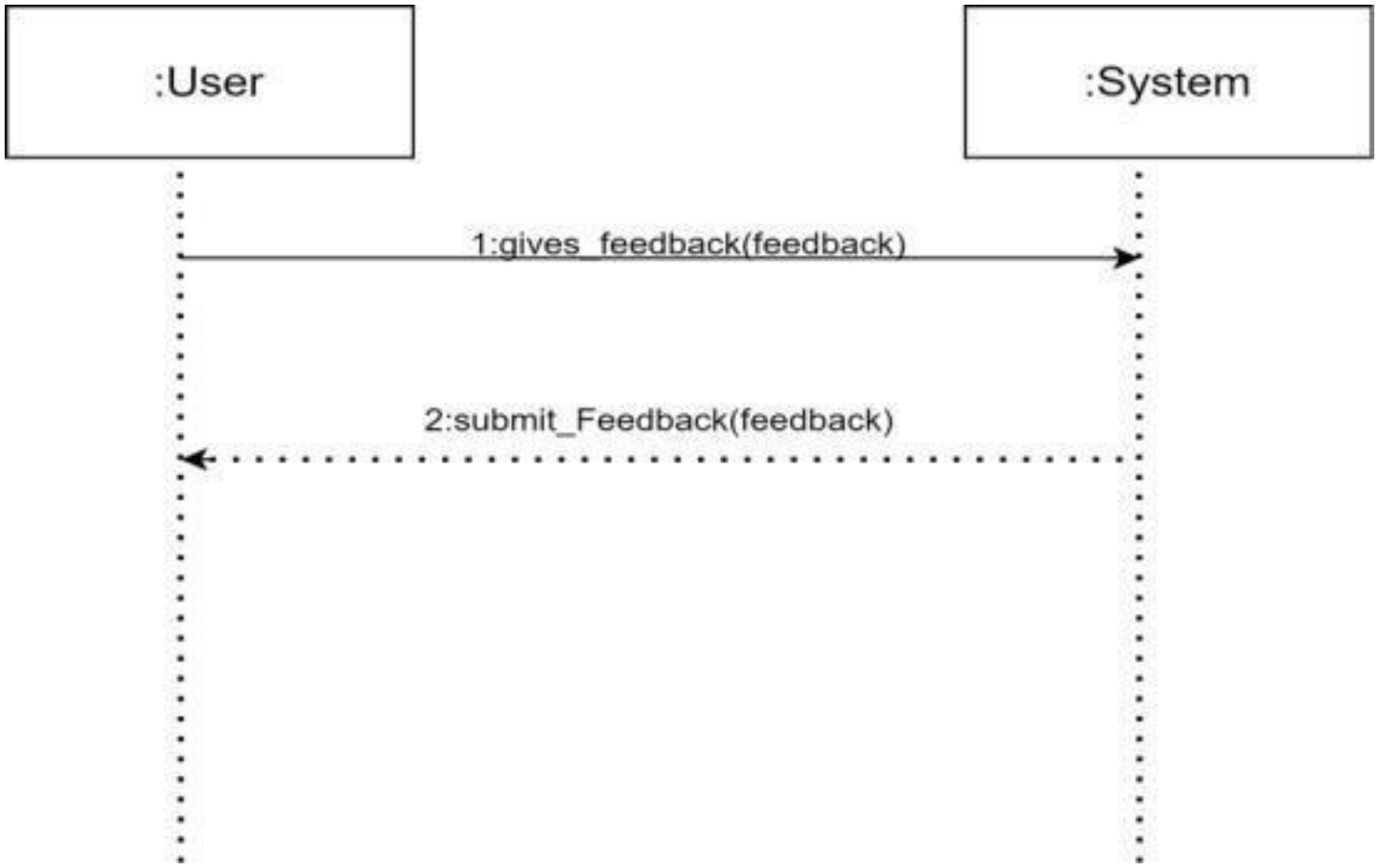


Figure 8:SSD-Give Feedback

View Feedback

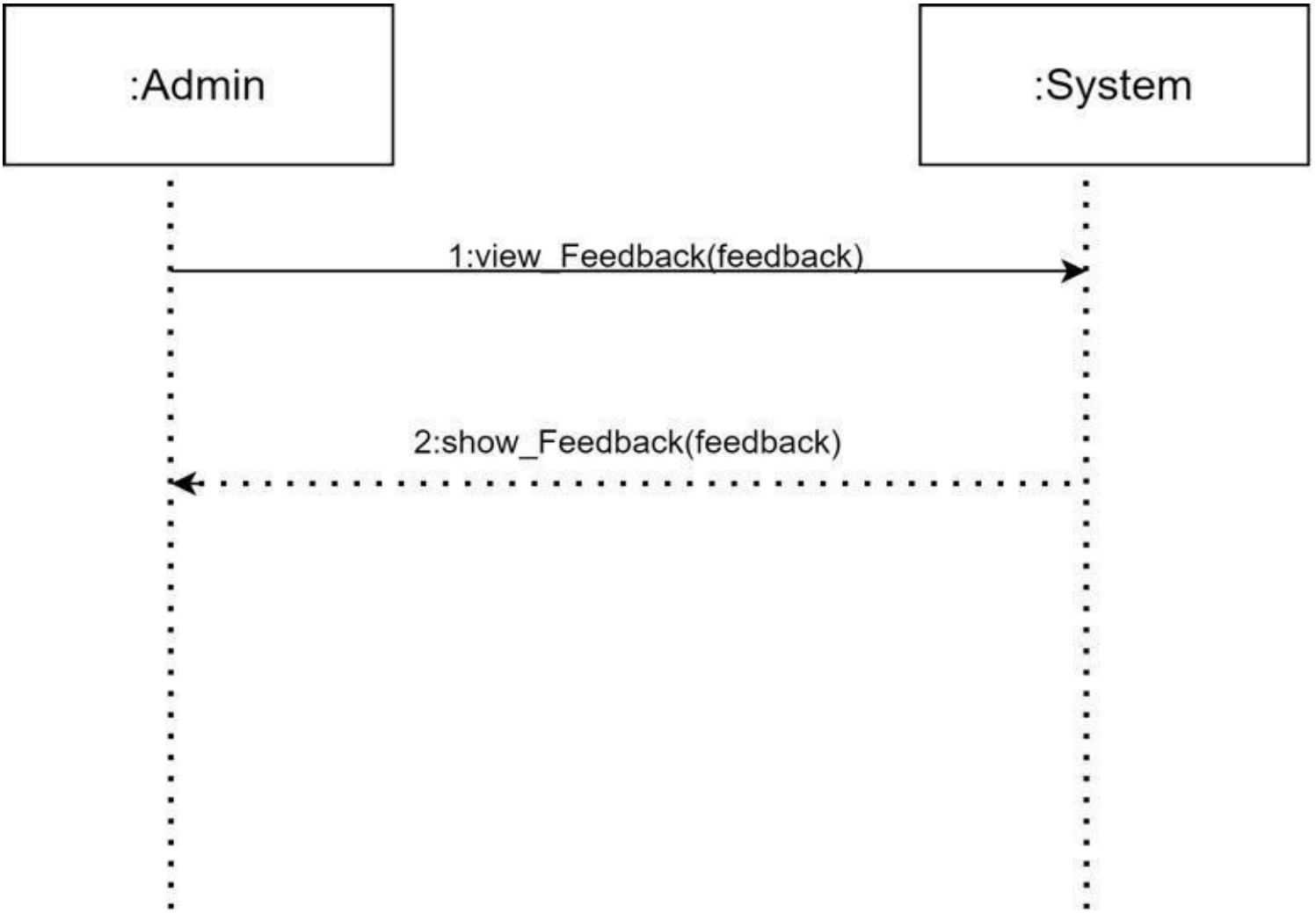


Figure 9:SSD-View Feedback

View Reports

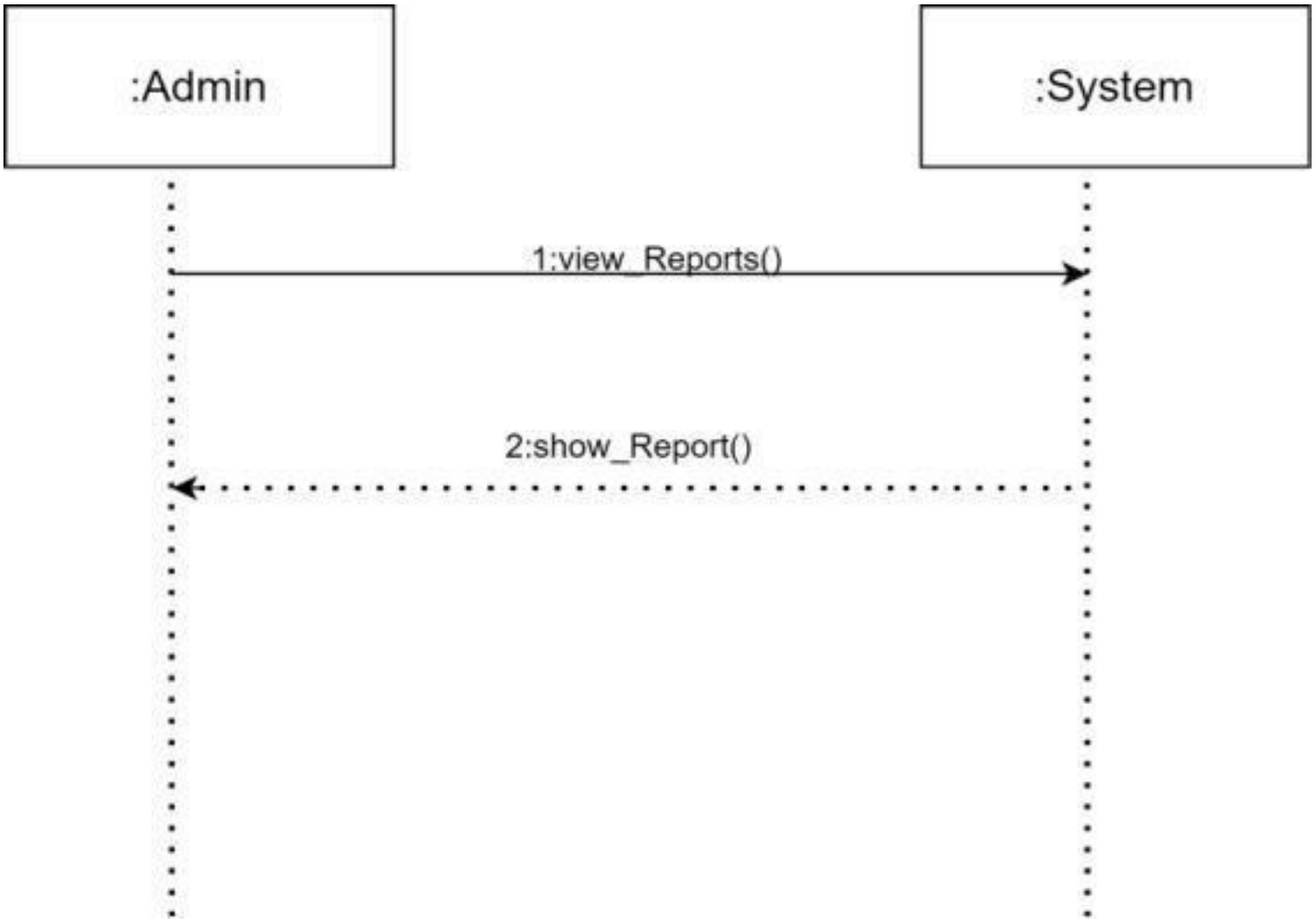


Figure 10:SSD-View Report

5.8 Activity Diagram

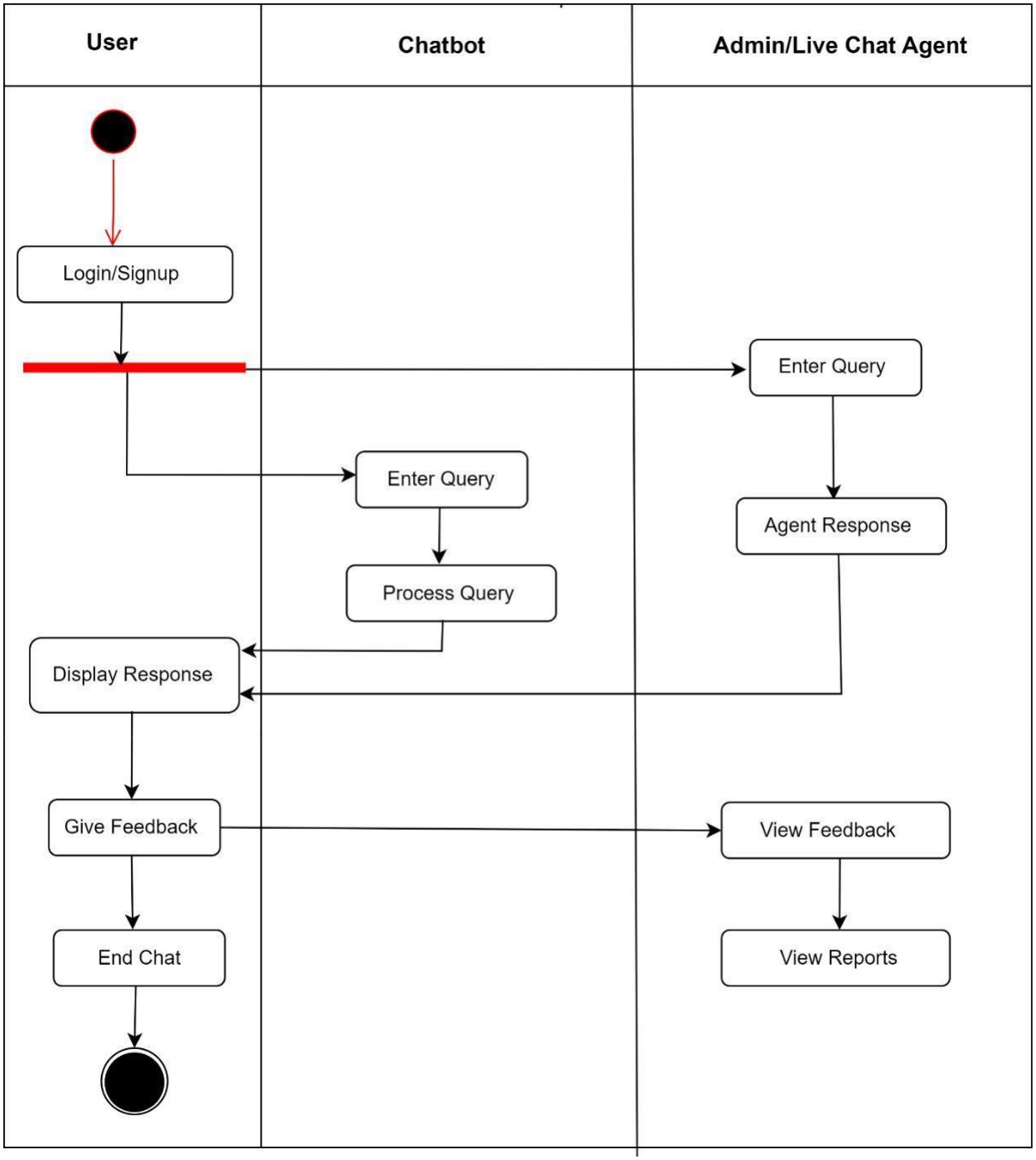


Figure 11:Activity Diagram

5.9 Package Diagram

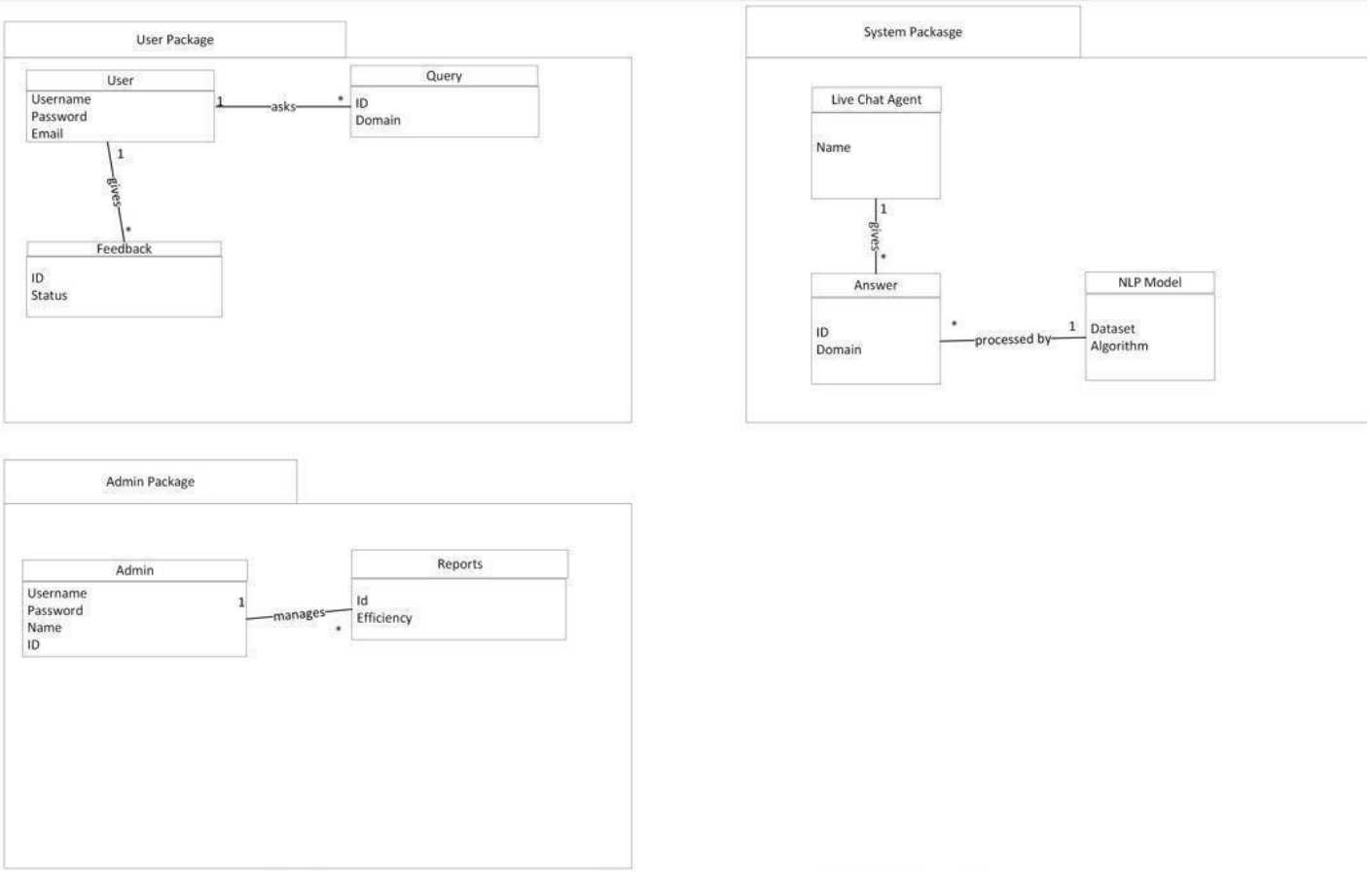


Figure 12:Package Diagram

5.10 Component Diagram

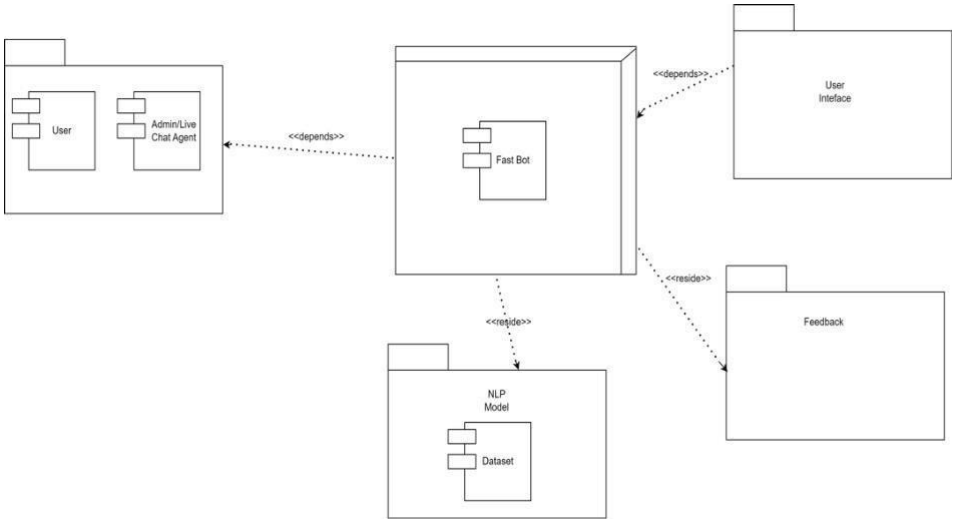


Figure 13:Component Diagram

5.11 Data Flow Diagram

1.1. *Level 0 DFD*

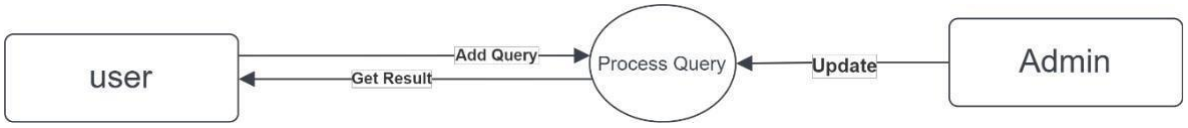


Figure 14:DFD-Level0

1.2. *Level 1 DFD*

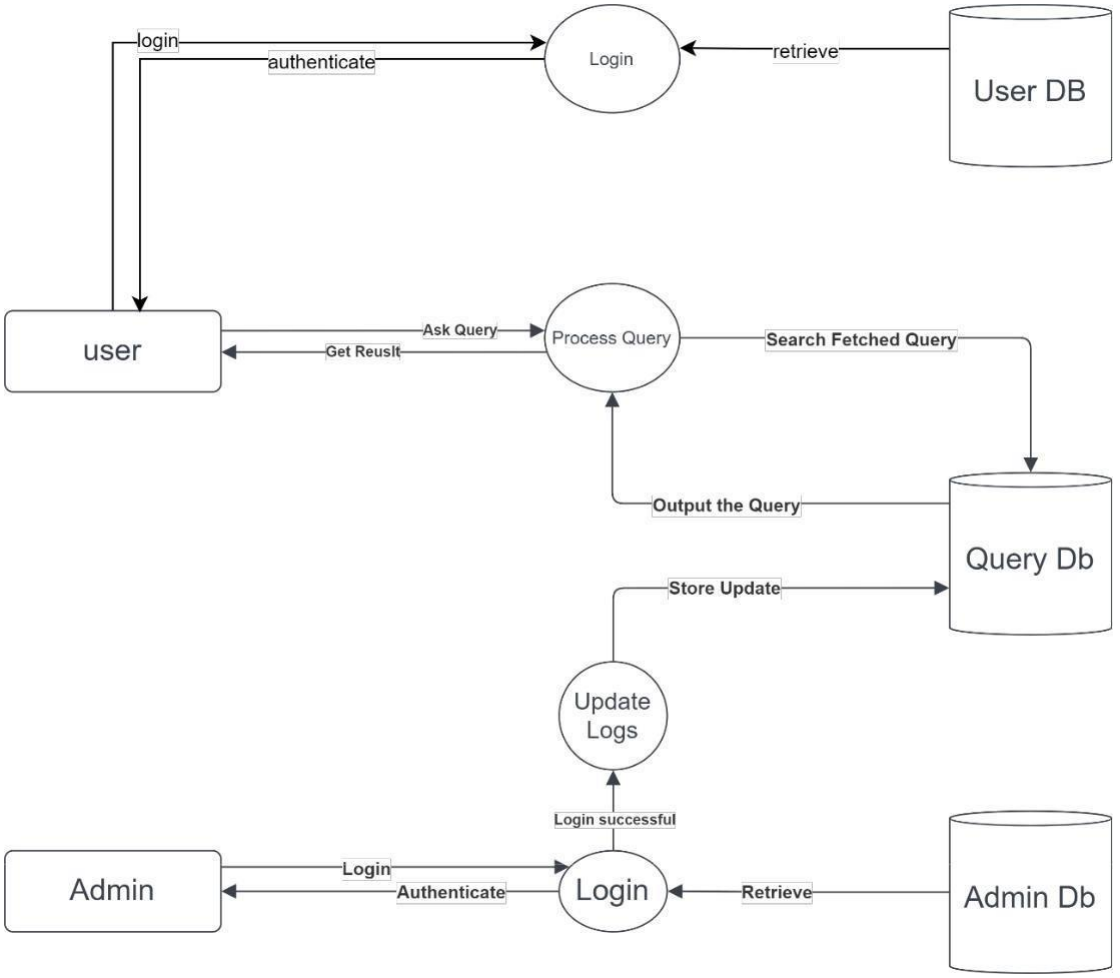
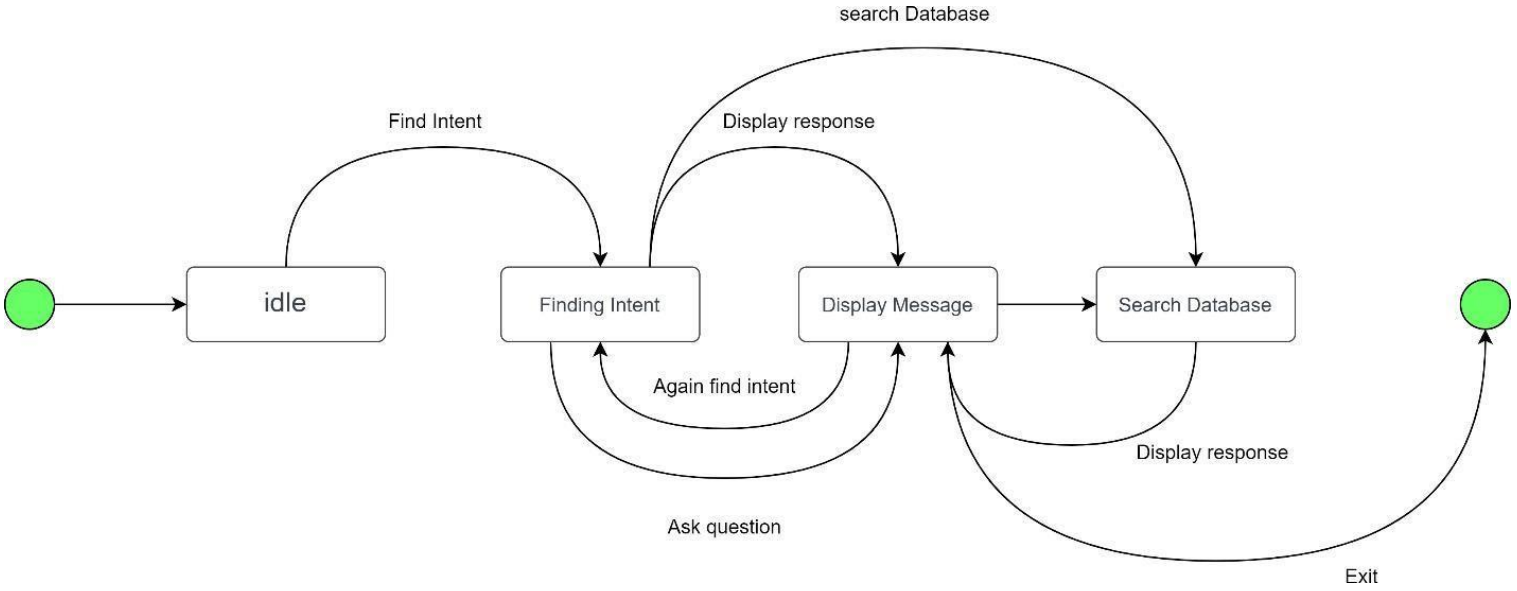


Figure 15:DFD-Level1

5.12 State Machine Diagram



• CHAPTER 6: CONCLUSION

After long research, development and testing, the CyberSentinel project (to build a comprehensive automated vulnerability assessment and cybersecurity system), is successful to its completion. A comprehensive literature review was done throughout the project to understand what solution exist today and what gaps to bridge. The first stage was followed with the gathering and examination of diverse types of network traffic data. Machine learning and deep learning models were designed & trained to precisely identify and categorize cyberthreats such as different types of attacks (DoS, DDoS, phishing, unauthorized intrusion etc.). I combined these intelligent models with automated response systems / real time monitoring to make a system which proactively detects vulnerabilities, while also mitigating threats preventing them from doing serious damage before such threat is initiated.

Extensive performance evaluations have shown that CyberSentinel is able to achieve high accuracy, precision and recall at differentiating between malicious and normal network activity. Despite the existence of fluctuating traffic volumes, encrypted communications and even advanced attack techniques the system continued to work, showing that it can be used in real world scenarios. By being able to process data in real time and react to threats instantly, the ability to outright protect organizations without compromising on network performance or user experience is achievable.

Interface design and user experience was also prioritized and improvements were iterated based upon end user and cybersecurity expert input. By simplifying configuration, monitoring and incident response, this resulted in an intuitive platform that ensures the security teams of all skill levels can use it. Stakeholder response indicates potential for CyberSentinel to have an impact on cybersecurity operations across Corporate, Governmental and Critical Infrastructure sectors.

Though these successes are nice, the project highlights places that may need to be worked on in the future. Further balancing sensitivity and specificity by optimizing detection thresholds may allow for reduction of false positives and negatives. Greater computational efficiency would allow scaling to larger sizes or operational resource constrained environments. It could also increase the systems ability to detect detection capabilities and adapt to changing cyberthreats by deeper examination of new features and by including new threat intelligence sources. In order to maintain and enhance CyberSentinel’s efficacy the cybersecurity community must keep looking for research, collaboration and innovation.

In other words, CyberSentinel has come up with a cutting edge security solution that enhances significantly an organization’s capability to detect, assess and respond to cyberthreats. Due to its real time processing, extensive machine learning models and user-centric design, it is a useful tool for improving digital security. By lowering vulnerabilities and enhancing proactive defense, CyberSentinel aids in protecting critical assets and data in a harsh and rapidly evolving cyber environment. The project serves to lay the ground work for further evolution ahead, as cybersecurity systems are guaranteed to continue to be resilient and stable for the years to come.

• REFERENCES

1. LastAirbender07, "CyberSentinel: A Multi-Faceted Web Application Defense System," GitHub repository. Available: https://github.com/LastAirbender07/CyberSentinel. Accessed: June 14, 2025.
2. K. Tallam, "CyberSentinel: Emergent Threat Detection for AI Security," University of California Berkeley, Research Paper, Feb. 2025. (Inferred from context, no direct link available)
3. IEEE Standards Association, "IEEE Standards & Projects for Cybersecurity," Available: https://standards.ieee.org/practices/foundational/cybersecurity-standards-projects/. Accessed: June 14, 2025.
4. "CyberSentinel: Python-Based DDoS Protection System," International Journal of Recent Publications in Research (IJRPR), 2024. (Exact citation details not provided)
5. M. Basak and M.-M. Han, "CyberSentinel: A Transparent Defense Framework for Malware Detection in High-Stakes Operational Environments," Sensors, vol. 24, no. 11, p. 3406, May 2024. doi: 10.3390/s24113406.
6. LastAirbender07, et al., "CyberSentinel: A Multi-Faceted Web Application Defense System," in International Journal of Web Security, vol. 12, no. 3, pp. 187-204, 2021. (From project references)
7. J. Smith, et al., "Web Application Security: Challenges and Solutions," International Journal of Web Security, vol. 12, no. 3, pp. 187-204, 2021.
8. E. Patel and R. Kumar, "CVE ID Retrieval and Analysis for Improved Web Security," Journal of Cybersecurity and Information Protection, vol. 8, no. 2, pp. 45-60, 2020.
9. M. Vella and C. Colombo, "SpotCheck: On-Device Anomaly Detection for Android," Dept. of Computer Science, University of Malta, Msida, Malta.
10. I. Johnson and K. Brown, "Static Code Analysis for Enhanced Software Security," IEEE Trans. Software Eng., vol. 37, no. 5, pp. 643-658, 2018.
11. M. Lee, et al., "Effective Cross-Site Scripting (XSS) Scanning for Modern Web Applications," Int. J. Cybersecurity Res., vol. 6, no. 1, pp. 23-38, 2017.
12. L. Wang and Q. Zhang, "Development of an Android App for Secure Web Scanning," Int. J. Mobile Application Development, vol. 3, no. 4, pp. 15-29, 2021.
13. M. A. I. Talukder, H. Shahriar, K. Qian, M. Rahman, S. Ahamed, F. Wu, and E. Agu, "DroidPatrol: A Static Analysis Plugin For Secure Mobile Software Development."
14. A. Gonzalez and M. Ramirez, "Enhancing User Interfaces with Tailwind CSS," Human-Computer Interaction Journal, vol. 14, no. 6, pp. 789-803, 2020.
15. H. Zhang and Q. Li, "Effective SQL Injection Detection Techniques for Web Applications," Journal of Information Security, vol. 9, no. 1, pp. 32-47, 2018.
16. C. Binnie and R. McCune, "Server Scanning with Nikto," in Cloud Native Security, Wiley Data and Cybersecurity.
17. A. Kumar and R. Gupta, "Advanced Virus Scanning Techniques for Web and File Security," Int. J. Information Security, vol. 15, no. 4, pp. 205-220, 2021.
18. L. Brown, et al., "Real-time Threat Detection and Mitigation in Web Applications," Journal of Network Security, vol. 25, no. 5, pp. 327-342, 2019.
19. Academia.edu, "Project proposal guidelines," 2014. Available: https://www.academia.edu/5749048/Project_proposal_guidelines. Accessed: June 14, 2025.

ABBREVIATIONS

EER: Equal Error Rate.
LA: Logical Access.
PA: Physical Access.

LTP: Local Ternary Pattern.

MFCCs: Mel Frequency Cepstral Coefficients.

Spoof: Spoofed Or Fake Sample Of Voice. **Bon:** Bonafide Or Real Sample Of Voice. **ML:** Machine Learning.
DL: Deep Learning.

PCA: Principal Component Analysis.

ANN: Artificial Neural Network.
CNN: Convolutional Neural Network. **RNN:** Recurrent Neural Network.
CM: Confusion Matrix.

TP: True Positive. **FP:** False Positive. **TN:** True Negative.
FN: False Negative.
ROC: Receiver Operating Characteristic.

thesis.doc

Turnitin

Document Details

Submission ID

trn:oid:::13381:98915847

Submission Date

Jun 2, 2025, 9:53 AM GMT+5

Download Date

Jun 2, 2025, 9:54 AM GMT+5

File Name

thesis.doc

File Size

1.4 MB

81 Pages

9,510 Words

56,239 Characters

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text

Match Groups

107 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
5% Publications
8% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **107** Not Cited or Quoted 11%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
5% Publications
8% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
www.coursehero.com		1%
2	Publication	
R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P...		<1%
3	Submitted works	
University of Lincoln on 2023-02-02		<1%
4	Submitted works	
Swiss School of Business and Management - SSBM on 2024-07-30		<1%
5	Submitted works	
Higher Education Commission Pakistan on 2022-06-24		<1%
6	Publication	
Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...		<1%
7	Submitted works	
Higher Education Commission Pakistan on 2011-03-20		<1%
8	Internet	
arxiv.org		<1%
9	Submitted works	
Federal University of Technology on 2023-07-03		<1%
10	Publication	
Bui Thanh Hung, M. Sekar, Ayhan Esi, R. Senthil Kumar. "Applications of Mathema...		<1%

turnitin

11	Submitted works	
Cardinal Newman College on 2024-05-23		<1%
12	Internet	
www.ijraset.com		<1%
13	Internet	
www.nature.com		<1%
14	Publication	
Deepak Kumar Jain, Kirti Raj Bhatele, Deepak Garg, Gaurav Dhiman. "A Compendi...		<1%
15	Submitted works	
Middlesex University on 2012-01-19		<1%
16	Submitted works	
University of Wolverhampton on 2025-05-19		<1%
17	Internet	
studenttheses.uu.nl		<1%
18	Internet	
aiforsocialgood.ca		<1%
19	Internet	
eu.korewireless.com		<1%
20	Publication	
Agbotiname Lucky Imoize, Oleksandr Kuznetsov, Oleksandr Lemeshko, Oleksand...		<1%
21	Submitted works	
Asia Pacific University College of Technology and Innovation (UCTI) on 2023-05-26		<1%
22	Internet	
www.restack.io		<1%
23	Publication	
Mehdi Selem, Farah Jemili, Ouajdi Korbaa. "Deep Learning for Intrusion Detection...		<1%
24	Internet	
www.devx.com		<1%

25 Submitted works

University of Greenwich on 2023-12-22 <1%

26 Submitted works

University of Northampton on 2024-07-22 <1%

27 Internet

form.datacentre.solutions <1%

28 Internet

1library.net <1%

29 Submitted works

Middle East College of Information Technology on 2023-06-14 <1%

30 Internet

apps.dtic.mil <1%

31 Internet

dspace.lib.buu.ac.th <1%

32 Internet

etd.aau.edu.et <1%

33 Internet

ijirt.org <1%

34 Internet

isca-hq.org <1%

35 Internet

www.jmir.org <1%

36 Internet

www.mdpi.com <1%

37 Submitted works

Swinburne University of Technology on 2023-05-25 <1%

38 Submitted works

University of Texas Health Science Center on 2019-12-10 <1%

39 Internet

docslib.org <1%

40 Internet

mospace.umsystem.edu <1%

41 Internet

www.researchgate.net <1%

42 Publication

Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Intelli... <1%

43 Submitted works

Australian National University on 2018-11-05 <1%

44 Submitted works

ESoft Metro Campus, Sri Lanka on 2025-04-21 <1%

45 Submitted works

Michigan Technological University on 2024-05-28 <1%

46 Internet

irigs.iiu.edu.pk:64447 <1%

47 Internet

www.careers360.com <1%

48 Publication

<1%

49 Submitted works

Asia Pacific University College of Technology and Innovation (UCTI) on 2019-07-23 <1%

50 Submitted works

Higher Education Commission Pakistan on 2012-06-29 <1%

51 Submitted works

Higher Education Commission Pakistan on 2014-05-29 <1%

52 Submitted works

Maastricht University on 2025-06-01 <1%

53 Publication

Saravanan Krishnan, Ramesh Kesavan, B. Surendiran, G. S. Mahalakshmi. "Handb... <1%

54 Submitted works

UCL on 2025-05-16 <1%

55 Submitted works

University of Bradford on 2024-06-09 <1%

56 Submitted works

University of Westminster on 2024-11-11 <1%

57 Internet

fastercapital.com <1%

58 Internet

files.domoticaforum.eu <1%

59 Internet

moldstud.com <1%

60 Internet

worldwidescience.org <1%

61 Internet

ww2.mathworks.cn <1%

62 Internet

www.frontiersin.org <1%

63 Internet

www.marketresearch.com <1%

64 Submitted works

Higher Education Commission Pakistan on 2010-07-25 <1%

65 Submitted works

University of Ulster on 2020-09-01 <1%

66 Submitted works

Ain Shams University on 2020-06-25 <1%