Shehryar Mughal
01-135221-049

Mubeen Imran
01-135221-067

AI-Based Data Breach Alert System

Supervisor: Dr. Kabeer Ahmed Bhatti

Bachelors of Science in Information Technology

Department Of Information Technology
Bahria University Islamabad

November 2025

Certificate

We accept the work contained in the report titled "AI-Based Data Breach Alert System", written by Shehryar Mughal Mubeen Imran as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Information Technology.

Approved by...

Supervisor: Dr. Kabeer Ahmed Bhatti

_____

Internal Examiner:

_____

External Examiner:

_____

Project Coordinator: Dr. Kabeer Ahmed Bhatti

_____

Head of the Department: Dr. Khurram Shahzad

_____

28th, November 2025

Abstract

The AI-Based Data Breach Alert System keeps an eye on internal communication in a telecom company. This smart system aims to watch user activities, check login patterns, and look at emails, texts, and voice messages using AI. The AI learns to spot content that doesn't look like normal work talk, marks it as "Spam," and tells the system admin right away.

This project wants to build a safe and quick platform for admins and staff to use. The system checks IP addresses to make sure known users can log in from work devices. Once they're in, people can talk in different ways, and AI reads each message to see if it's about work. It turns voice messages into text and checks them too. The system also controls who can open documents. Users must give good reasons to download any PDF files, and the system checks if they're allowed based on their IP.

To wrap up, this project makes internal monitoring automatic by putting together AI-powered communication analysis, IP-based access control, and oversight at the admin level. The admin dashboard brings together all user activity logs in one place, including failed login tries, message groupings, and requests to download documents. This solution's goal is to stop internal data leaks by spotting risky or improper behavior and making sure communication within the organization stays secure.

Acknowledgement

In the Mercy and Goodness of Allah. Allah, the Almighty, is the Praiseworthy. Our Prophet Muhammad (peace be upon him), his family, and all his companions are worthy of our prayers. We would like to express our gratitude to Dr. Kabeer Ahmed Bhatti, our supervisor, for giving us the chance to collaborate with him on the Final Year Project. He is a remarkable supervisor as he guides us on the correct road and assists us in improving. Finally, we would like to express our gratitude to our family and friends for their unfailing prayers and support. May Allah (SWT) guide us on the correct road and assist us in improving as people. Ameen.

Shehryar Mughal
Bahria University
E8, Islamabad, Pakistan

Mubeen Imran
Bahria University
E8, Islamabad, Pakistan

November 2025

List of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
ES6	ECMAScript 6
GUI	Graphical User Interface
HTML	HyperText Markup Language
IP	Internet Protocol
LAN	Local Area Network
LSTM	Long Short-Term Memory
ML	Machine Learning
MNB	Multinomial Naïve Bayes
MSSP	Managed Security Service Provider
NLP	Natural Language Processing
NoSQL	Non-Relational Structured Query Language
NumPy	Numerical Python
PK	Primary Key
REST	Representational State Transfer
RBAC	Role-Based Access Control
SQL	Structured Query Language
SWT	Subhanahu wa Ta'ala
STT	Speech-to-Text
UI	User Interface
UX	User Experience

Chapter 1

Introduction

1.1 Overview

Companies are finding it harder to keep their internal communication safe in fields like telecom where workers often share sensitive info. Watching what users do and checking their messages by hand takes a lot of time and work and often misses threats or things people shouldn't be doing. To fix these problems and protect data better, companies are now using AI and NLP to build AI systems that warn about data breaches. NLP is a big part of these systems. It helps computers understand and look at human language, so they can spot messages that aren't work-related or aren't okay in emails, texts, and even voice recordings that have been written down. With NLP, AI can sort messages into okay or suspicious by looking at certain words how they sound, and what they're about. When the AI works on the text, it uses NLP tricks like breaking words apart, taking out words that don't matter, and finding the main part of words to clean up and make the text the same before sorting it. As AI and NLP get better, people think these AI warning systems for data breaches will become smarter and more trustworthy. They will play a critical role in transforming internal security practices by providing real-time alerts and administrative oversight, enabling organizations to maintain confidentiality, accountability, and compliance in a more efficient and effective manner.

1.2 Background

Many companies in fields handling sensitive data like telecom deal with huge amounts of daily internal communication. They rely on emails, messages, and calls to operate. Going through all these communications by hand takes too much time, is prone to mistakes, and is impossible. Businesses need a smart system to check these messages , catch unusual patterns, and highlight anything that might point to a data leak or misuse of company tools. This creates a demand for AI systems that process large chunks of unstructured communication and turn it into clear and organized data for managers to look at.

The system gathers details about user login habits assigned IP addresses, message data like text and voice, timestamps, and how often people communicate. It uses Natural Language Processing to pull out key points main ideas, and tone to figure out if the messages are related to work. It also converts voice messages into text and applies the same process to classify them. The system stores all this information in an organized MySQL database, which helps to analyze it later, create alerts, or review it during audits. Internal communication, unlike emails or shared posts, sticks to company rules. This makes it easier to spot differences but changing communication styles and formats still makes it tricky for usual monitoring tools.

To tackle these issues, our AI tool uses a strong mix of machine learning and NLP to find

patterns, spot violations, and sort content into "Spam" or "Not Spam." It checks document access based on registered IPs and the reasons users provide making sure that downloading sensitive files stays under control. This smart solution helps by handling policy violations and offering real-time protection to keep company data secure. Its scalable design boosts security for organizations, lowers the chances of internal breaches, and gives administrators useful insights to better manage how users act.

1.3 Problem Description

This project focuses on creating and setting up an AI-powered system to alert about data breaches. The main goal is to improve an organization's cybersecurity by helping it watch activities in real time and respond to possible breaches. This system works to lower the chances of exposing sensitive data by spotting strange patterns, behaviors, and activities in network traffic and user actions with the help of AI and machine learning methods.

The plan includes making a secure platform that can expand as needed and supports admin-friendly features. It will let admins set monitoring rules, get instant alerts via SMS or email, and view detailed logs and reports on a web portal. To boost how well threats are identified and categorized, the system might also include NLP tools to study logs or textual data related to potential threats.

Other goals are to cut down the time it takes to respond to incidents, boost how IT security teams work, and meet data protection rules. Reaching these aims helps the system give businesses an active security setup that stops big data breaches and keeps information safe and private.

1.4 Objectives

The main aim of creating an AI-Based Data Breach Alert System is to give organizations in the telecommunication industry an effective solution to detect breaches , send alerts, and automate responses to incidents. The detailed goals of this project include the following:

The project's goal is to build a smart system that can analyze system logs, network traffic, and user actions to spot abnormal patterns signaling possible security risks or data breaches. The system uses modern machine learning tools and anomaly detection methods to watch activities in real time. It sends alerts through SMS and email whenever it notices any unusual behavior. The platform is meant to help organizations like telecom firm's large businesses, or schools that need strong cybersecurity defenses and breach alert systems. It keeps a single database to store breach cases, user activity details, and alert logs offering material to analyze and guide management decisions. The system also gives security suggestions and insights to aid IT and security teams in creating stronger ways to protect data. The main aim is to cut down response time, lower the effort needed for manual security checks, and strengthen the organization's protection against today's cyber threats.

1.5 Purpose and Scope

1.5.1 Purpose

Studies reveal that many data breaches remain unnoticed for weeks or even months because companies lack effective real-time monitoring systems. This delay means organizations often discover breaches after serious harm has been done, showing why strong security measures are essential. This project aims to create a tool to track system activity around the clock spot suspicious actions and keep a complete record of events with fast response times. The system is simple for IT staff to use since it sends instant notifications, provides thorough logs, and offers tips built on AI-driven insights. It also benefits organizations by letting them export breach details like CSV files to analyze data and make better choices to manage risks.

1.5.2 Scope

The system collects breach-related information and organizes it into tables and CSV files. This setup helps organizations analyze data and review incidents after they occur. It also provides instant alerts, smart predictions, and risk ratings. Security teams can use these features to react fast to threats and improve their defense measures over time. With a simple dashboard and automated alerts, the platform becomes more practical and easier to use. Businesses like telecom providers, IT companies, and schools can use it to track critical data, identify system weak spots, and observe user behavior. The tool also lets organizations study common breach methods weak areas in their systems, and attacker behaviors to boost their cybersecurity plans .The system improves over time by using feedback from users. This allows it to adapt and offer stronger defense against new threats as they appear.

1.6 Solution Application Areas

The AI-powered breach alert system can connect with telecom companies to detect and alert threats in real time improving the security of sensitive customer and company data. IT and security teams in businesses can rely on this system to run round-the-clock monitoring, detect anomalies, and trigger automated actions, boosting their defenses against threats from both inside and outside. Cloud providers and hosting services can implement the system to track client environments, identify unusual activities, and deliver extra security features to clients. Universities, colleges, and other educational organizations can apply the system to safeguard student records administrative details, and research data while helping IT teams understand hacking attempts and improve institutional cybersecurity measures.

Chapter 2

Literature Review

This chapter explains the work of different researchers and writers in cybersecurity, with a focus on detecting data breaches and creating alert systems. It gives a short summary of tools and methods already in use to track and spot security issues as they happen. The chapter uses past research to understand better how artificial intelligence works together with anomaly detection models to improve security systems. It talks about why this area of work matters how using an AI-based alert system helps find unusual behavior, study patterns, and send quick warnings. The chapter also talks about the need to build a web app to make the system easier to use and more practical for businesses wanting to safeguard their important data.

2.1 Natural Language Processing

This chapter explains the work of different researchers and writers in cybersecurity, with a focus on detecting data breaches and creating alert systems. It gives a short summary of tools and methods already in use to track and spot security issues as they happen. The chapter uses past research to understand better how artificial intelligence works together with anomaly detection models to improve security systems. It talks about why this area of work matters how using an AI-based alert system helps find unusual behavior, study patterns, and send quick warnings. The chapter also talks about the need to build a web app to make the system easier to use and more practical for businesses wanting to safeguard their important data.

The AI-Based Data Breach Alert System uses several interconnected modules to work . It aims to build a single web application to detect threats, classify them, and send alerts using AI technologies like Natural Language Processing. Each module is vital to spot, study, and handle possible security risks as they happen. Below are the NLP methods included in the system:

- **Tokenization:** breaks system logs, emails, or alerts into smaller pieces called tokens. These tokens help the system study text data to find patterns or risky keywords tied to security breaches.
- **Stop Word Removal:** The system removes used words like “the,” “is,” and “and,” since they do not add much to the analysis. This helps to focus more on key terms and crucial words that might signal an anomaly or breach.
- **Lemmatization and Stemming:**Lemmatization and Stemming: Logs and security messages often include different forms of the same word. The system simplifies these to their basic root form. This keeps the analysis consistent and makes it easier to detect patterns.
- **Part-of-Speech Tagging:** Words in logs or alerts are labeled by their grammatical function, like nouns or verbs. This step adds clarity to the data’s structure and helps identify clues about possible threats.

The literature review focuses on studying earlier work in anomaly detection, NLP applications in cybersecurity, and systems for real-time threat alerts. These areas form the base to create an intelligent and responsive platform for detecting data breaches.

Diagram Code:

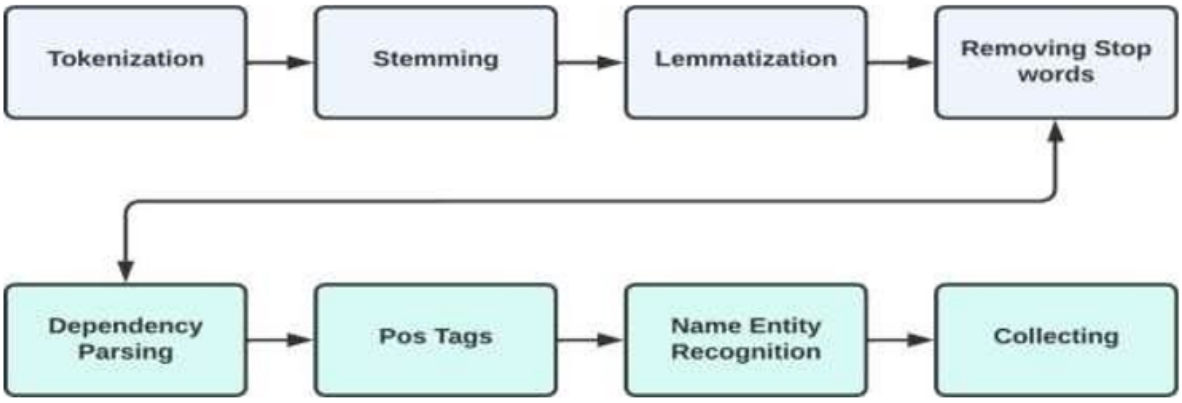

Figure 2.1: Natural Language Processing Flow Diagram

2.2 Feature Extraction

Feature extraction plays an important role in changing raw log and network traffic data into organized formats. These formats help machine learning models understand and work better. People often use methods such as detecting patterns based on frequency, scoring anomalies, or mapping features according to time. These methods make it easier to turn messy security data into useful input information. In our AI-based Data Breach Alert System, we use different steps like cutting out noise, removing repeated entries, and making data more uniform. These steps aim to boost how well the system detects threats and extracts features.

2.3 Similar Applications

Many tools like the AI-based Data Breach Alert System have emerged to handle the growing problems of cybersecurity attacks unauthorized system access, and breaches. These tools use AI and machine learning to spot odd activities, find possible breaches, and send out alerts . This reduces the need to depend on humans to monitor systems. Here are a few popular tools that serve a similar purpose:

- **IBM QRadar:** QRadar is a type of Security Information and Event Management, or SIEM, solution. By using AI and machine learning, it watches for security risks in real-time. It gathers and studies data logs from servers, applications, and network devices. If it notices strange activity or threats, it sends out alerts.
- **Splunk Enterprise Security:** Splunk Enterprise Security applies data analytics to deliver security insights throughout a business. Machine learning models trained on past and live security data help it recognize threats, analyze risks, and create alerts on its own.
- **Darktrace:** Darktrace uses AI-powered anomaly detection to spot unusual patterns in network activity that could hint at malware, insider problems, or data theft. It relies on unsupervised learning, which adjusts to new environments without needing preset rules.

- **Microsoft Sentinel:** Microsoft Sentinel provides a flexible cloud-based system for managing security. The AI built into it cuts down on unnecessary noise, finds genuine security events, and enables quicker and better responses to security issues.

Chapter 3

Requirement Specifications

This chapter gives a detailed overview of what the AI-Based Data Breach Alert System needs to work. It looks at what the system can and cannot do right now and lists the important functional needs alongside other necessary features to make the solution work . To analyze these needs, we studied security systems already in use and spotted where artificial intelligence could improve how well breaches are detected and alerts are sent. The chapter also explains the system's workings with case diagrams. These diagrams show key interactions among users, the system, and admins helping to explain how everything operates together.

3.1 Present System

The way people approach cybersecurity has developed over time. At first, businesses used basic methods like firewalls, antivirus programs, and checking logs to spot possible dangers. IT teams had to spend a lot of time finding and reporting anything unusual making the whole process slow and more about reacting than preventing. As companies grew and their digital systems became more complicated, cyber risks also became more advanced and frequent. To handle these increasing issues many businesses started working with manage security service providers. These MSSPs used simple monitoring systems to find known risks. They provided regular reports and warnings about potential security problems based on set rules. This method showed major weaknesses. It could not spot unknown or changing threats leaving systems open to advanced ongoing attacks and risks from insiders.

3.2 Limitations of present system

Keeping track of network activity and user behavior takes up too much time and does not work well in big companies. Older security systems do not provide live updates and might miss complex or internal threats until after major harm is done. Systems that rely on fixed rules do not adjust or improve to handle new kinds of cyberattacks, which puts organizations at greater risk. Security teams often deal with piles of log data. This makes it hard to tell real threats from a lot of false alarms. A lack of centralized tools and automated reports slows down responses to security problems, which makes the damage from breaches even worse.

3.3 Proposed System

The system called "AI-Based Data Breach Alert System" works as a security tool to monitor and alert issues within telecommunication firms and other big organizations. It uses powerful artificial intelligence and natural language tools to study user behavior, logs, email contents, voice recordings, and how files are accessed. A central database stores this information to check

for threats and analyze it . This system helps system admins a lot by giving them quick warnings about unusual activities. It sorts emails and voice messages as spam or useful, depending on whether they matter to the organization. It also helps understand user actions better with useful insights. The system works best in organized enterprise settings. It offers detailed tools to analyze and report helping IT administrators make informed decisions and take proactive security steps.

3.3.1 Advantages of proposed system

The system watches and studies how users act, communicate, and get into the system in real time. This helps it find possible data breaches and . It uses automatic tools to keep an eye on things and spot threats so IT administrators don't have to do as much manual work. The platform is a single place to check things like login attempts, IP addresses, emails, voice messages, and files that are downloaded. It sends precise alerts to administrators right away making it easier for the organization to handle security risks before they worsen. On top of that, it makes data security better and keeps things following the rules by marking any suspicious or unwanted activities as "Spam" and setting them aside for review.

3.4 Hardware Requirements

The AI-Based Data Breach Alert System needs a laptop or desktop with good enough processing power to handle backend operations and run AI models without problems. It should have at least 4GB of RAM, but having 8GB or more is better for smoother performance of the AI features. A strong internet connection is a must to enable real-time communication between the admin and users over the network. The system also works best with a single network setup like an in-office LAN or Wi-Fi. This setup helps track all user systems by registered IP addresses.

3.5 Software Requirements

The system relies on MySQL to keep organized data about users, IPs, login activity, emails, voice transcripts, and PDF-related actions. Developers use Python to build backend functions, create AI models, and implement NLP features. To access the web interfaces for users and administrators, you need a web browser, and Google Chrome is the preferred option. For coding and managing HTML templates, a text editor is necessary, with Visual Studio Code being the suggested choice.

3.6 Models and Integration

The AI-powered system that alerts data breaches works by gathering user activity data. It collects things like login tries, IP addresses, emails, voice recordings, and uploads of PDFs. The system uses NLP techniques to clean and prepare this data. A Flask backend API ties together all the different parts of the system. It changes voice recordings into text using the Whisper Speech-to-Text Model. Emails messages, and transcripts get processed through TF-IDF vectorization. After processing, the system uses a trained LSTM Deep Learning Model to check if the content is Spam or Not Spam.

The system uses the MySQL database to store activities, logs, and classification results keeping it connected to the backend to monitor everything in real time. It sends automated alerts when it notices suspicious actions like unauthorized IP access repeated failed logins, or spammy

content. By combining machine learning models, a Flask API, and the structured MySQL database, the system can detect and react to possible security issues as they happen.

3.7 Intended Audience

- **System Administrators and IT Security Teams:** System administrators and cybersecurity professionals form a key group using the AI-Based Data Breach Alert System. This tool helps them track user actions across the company network. It identifies unauthorized access, flags suspicious communications like emails or voice messages, and sends instant alerts. Security teams can prioritize critical threats and enforce policies instead of spending time on manual checks. It boosts response speed and cuts the chances of data breaches.
- **Small and Large Businesses:** This system helps both small businesses and big companies. Small businesses, which often do not have teams for cybersecurity, can rely on the system's automation and smart tools to keep basic security in place and track what users do inside the network. Larger companies, with their complicated networks and many employees, can use it to handle tasks like logging activities spotting unusual behavior, and sending alerts about breaches on a bigger scale. This helps them follow rules and cut down on risks from inside the organization.
- **Telecommunication Companies:** Telecom companies rely on this system because they manage vast amounts of data and handle sensitive customer information. The system secures internal communications, monitors IP-based login sessions, and checks user activities like emails, calls, or file usage. It catches any suspicious or rule-breaking actions. This strengthens how they handle data and makes their internal security practices better.

3.8 Limitations of project

The project has some limits. Picking out accurate features from processed log data and user messages like emails or voice notes can be tough. This can lower how precise the system is at classifying threats. The system needs a good internet connection because it depends on sending and receiving data in real-time between users' computers and the main server. Without a steady connection, it might miss or fail to monitor user actions. Developers built the system with open-source tools like Python and MySQL. These tools cut costs but might not support some advanced features needed for bigger enterprises. It can be hard to measure how well each extracted feature works in AI classification to address new or changing threats. On top of that, the team has 12 months to complete the project, which could limit deep model training thorough testing, or adding all the planned features.

3.9 Functional Requirements

The AI-driven Data Breach Alert System helps track user actions, examine communications, and notify admins . Its main features involve:

1. **Tracking User Logins:** Monitor login attempts along with the IP addresses used. Record failed attempts and show them on the admin panel.
2. **Analyzing Emails and Voice:** Use AI to examine emails and voice message transcripts. Sort them as "Spam" or "Not Spam" depending on workplace importance.

3. **Controlling PDF Access:** Before downloading files, users need to verify their IP and provide a valid reason. Track and log all file uploads and downloads.
4. **Admin Panel:** Offer a single interface for admins to watch over activity, get notifications, and check patterns using logs and graphs.
5. **Real-Time Alerts:** Inform admins right away when suspicious or unauthorized user actions occur.
6. **Feedback Collection:** Let users give feedback to improve the system.

3.9.1 Client Module: Getting IP and System Details

- **Description:** The system collects the user's active IP address and system details during login. This helps maintain secure access and track devices. This info confirms the validity of the login and keeps an eye out for unapproved access attempts.
- **Output:** The system provides the user's IP address, device or system ID, and the time of login.
- **Processing:** When users log in, the system gathers their IP and system details and saves them in the database to allow the admin to review later.
- **Input:** Users provide login details and try to access the system.
- **Primary Actors:** User or client.

3.9.2 Client Module: Techniques to Parse Email Voice, and Files

- **Description:** Users engage with the system by sending emails recording voice messages, or uploading files. The system applies parsing and language processing methods to review the content and check it for relevance or security concerns.
- **Output:** Analyzed email contents, voice message transcripts, and data from file details marked as either "Spam" or "Not Spam."
- **Processing:** The system reviews email content, converts voice messages into text, and examines file details through AI tools to recognize suspicious or unrelated material.
- **Input:** Users send emails, leave voice messages, or upload files.
- **Primary Actors:** Users or clients.

3.9.3 Client Module: AI-Powered Communication and Activity Sorting

- **Description:** The system examines emails, voice recordings, and patterns in file access that users provide. Using AI tools, it decides if the material relates to work or if it might be spam. It categorizes the content and notifies the admin about anything unusual.
- **Output:** Results showing if content is spam or not, along with alerts about odd messages or actions that break rules.
- **Processing:** The system reviews text and voice data converted into text by NLP. It checks for relevance, runs classification, and either logs the data or informs the admin about irregularities, if necessary.
- **Input:** Content made by users like emails, voice messages, or file requests.
- **Primary Actors:** Users or clients.

3.9.4 Admin Module: Handling Applicant Data

- **Description:** Admins go to the "Applicant Data" area to check user info in a table. This includes basic details, skills possible job roles, and scores. They can save this info as a CSV file to analyze it later.
- **Output:** Applicant info displayed in a table that can be downloaded as a CSV file.
- **Processing:** Admin opens the applicant data section and views or downloads the information.
- **Input:** Admin logs in and requests access to view or download applicant details.
- **Primary Actors:** Admins.

3.9.5 Admin Module: Watching User Activity and Communication Records

- **Description:** Admins use the dashboard to check user login records, review email and voice message categories, and view requests to download files. This tool helps them monitor activities live confirm AI choices , and record any unusual actions.
- **Output:** Admins can view logs of user actions, message categories (Spam or Not Spam), voice-to-text transcripts, and file access details.
- **Processing:** The admin logs in, picks a specific module like login records, email and voice tracking, or file tracking, to dive into a detailed review.
- **Input:** Admin provides login credentials and chooses specific data to review or confirm.
- **Primary Actors:** Admin staff.

3.9.6 Admin Section: Creating Pie Charts and Visual Summaries

- **Description:** The system creates visual summaries like pie charts and bar graphs inside the admin panel to show critical security data. This data covers things like login attempts how much communication is spam versus not, frequency of file access, IP-based login patterns, and levels of user activity.
- **Output:** Charts that provide security data and show stats about user activities.
- **Processing:** The system takes saved logs and makes graphs using the chosen options.
- **Input:** Admin chooses which metrics or data type they want to see as visuals.
- **Primary Actors:** Admin.

3.9.7 Feedback Section: Collecting User Input and Ratings on the System

- **Description:** Users share their opinions on how well the system works. This includes how good spam detection is how simple it is to use, and how fast it responds. They give ratings between 1 and 5 and can also write extra comments to suggest changes or new ideas.
- **Output:** The system collects ratings and written feedback from users.
- **Processing:** It saves all the feedback and scores in the database so admins can review them and make changes or improvements later.
- **Input:** Feedback forms filled out by users.
- **Primary Actors:** User or client.

3.9.8 Feedback Module: Showing Overall Ratings and Previous User Comments

- **Description:** The system uses feedback from users to create an overall performance score. It looks at how accurate the system is how well it blocks spam, and how easy it is to use. It keeps a record of old user comments and shows them to admins so they can check how the system performs and find ways to make it better.
- **Output:** The system shows an average rating score and saved user comments.
- **Processing:** It calculates the average of all user ratings then shows this score and old comments on the admin's dashboard.
- **Input:** Users provide feedback ratings and leave comments.
- **Primary Actors:** Admin and users.

3.9.9 Non-Functional Requirements

Here are the non-functional requirements for the AI-Based Data Breach Alert System:

- **Performance:** The system needs to process user activities, analyze messages, and create alerts in real-time . It has to perform to catch breaches and provide a smooth experience without delays.
- **Availability:** It must stay accessible to authorized users and administrators inside the organization's network. The system should remain available at all times when connected to reliable office internet systems.
- **Security:** The system has to include strong security steps to protect sensitive data. This includes features like IP checks, user logins, encrypted communication, and limited access to confidential logs and user information.
- **Reliability:** The system needs to monitor and report activities without constant failures. Stable and well-tested technologies paired with data backups and logging tools help ensure reliability.
- **Maintenance:** The development and IT security team handle system upkeep, updates, and AI model upgrades. They perform regular checks and make updates to keep the system running well and adapting to threats.

3.10 Use Cases

A diagram shows how users might interact with the AI-Based Data Breach Alert System and explains how both users and admins work within the system. It shows users taking actions like logging in, sending messages or emails, and uploading or downloading files. At the same time, it describes how admins keep track of and review these activities. The diagram highlights the main features of the system. These include tracking activities classifying content with AI as spam or not spam, converting voice to text, and controlling access to PDFs. This diagram acts as a guide to show how the system works and how users and admins connect with the smart alert framework.

3.10.1 General Use Case

The following Section gives a view of general use case of AI-Based Data Breach Alert, with the help of use case diagram and table.

Table 3.1: Use-Case 01: User Registration

Use-Case Name	Main Use Case
Actor	User / Admin
Roles	Users engage with messaging, file management, and notification features. Admin oversees user actions, gets alerts created by AI, and handles system management.
Description	This scenario shows the key ways users interact with the system. They log in, send messages, check emails, access PDFs, and handle voice messages. The admin manages the tasks, checks spam reports, and keeps the network secure.
Pre-condition	Both Users and Admin need to log in using the organization's secure internal network. Users also need to register and get a system-assigned IP address.
Post-condition	All actions are recorded by the system, which also categorizes texts, emails, and voicemails as "Spam" or "Not Spam" and displays the activity on the admin dashboard for monitoring.

Table 3.2: Use-Case 01: Main Use-Case

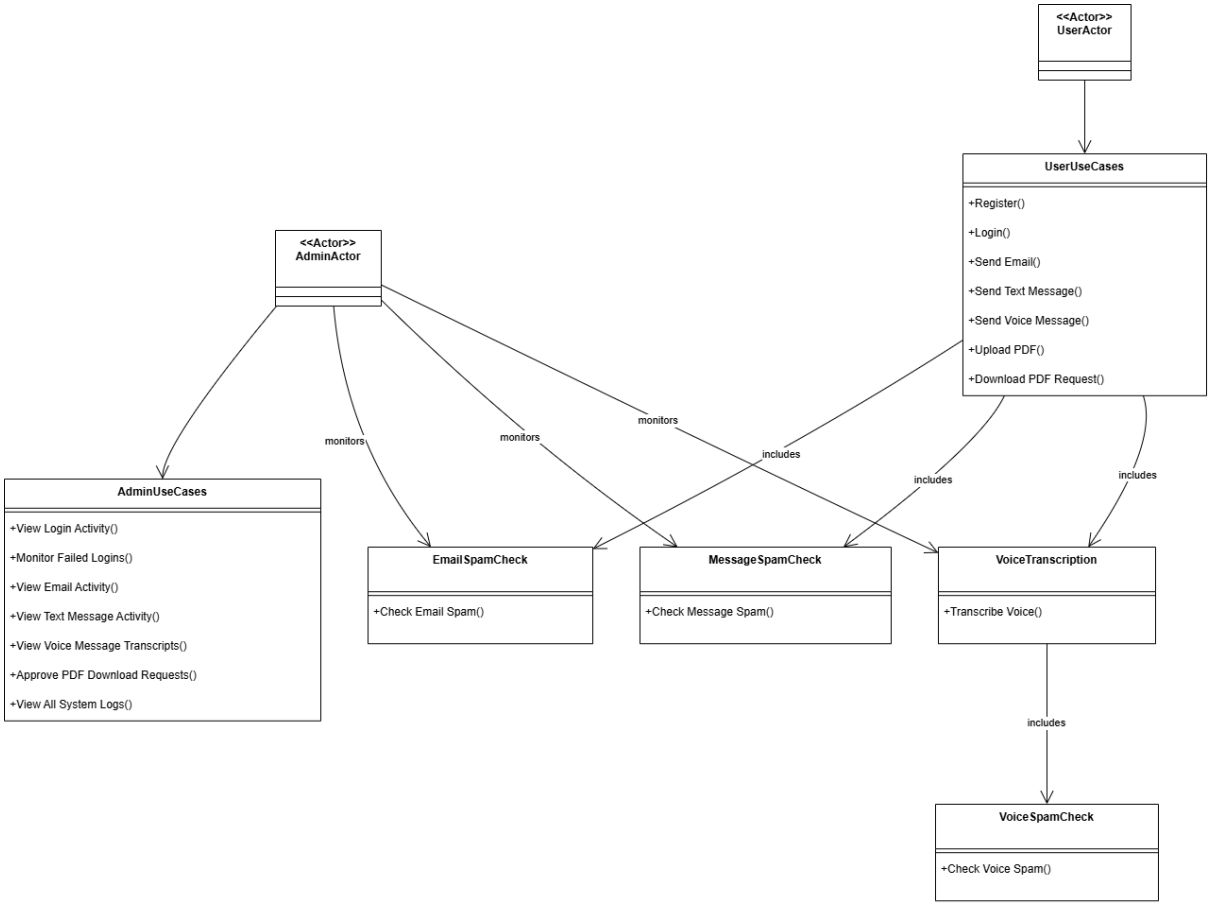

Figure 3.1: Main Use Case Diagram

3.10.2 User Registration

Table 3.3: Use-Case 01: User Registration

Use-Case ID	UC01
Use-Case Name	User Registration
Priority	High
Primary Actor	User
Description	Tracks and displays all login attempts (successful/failed) with IP addresses.
Basic Flow	User logs in; system records attempt with IP and shows it on the admin dashboard.
Alternative Flow	Incorrect credentials; system logs failed attempt and flags it.
Pre-condition	User accesses login page.
System Reaction	Records login attempt and updates dashboard.
Post-condition	Admin can monitor login activity.

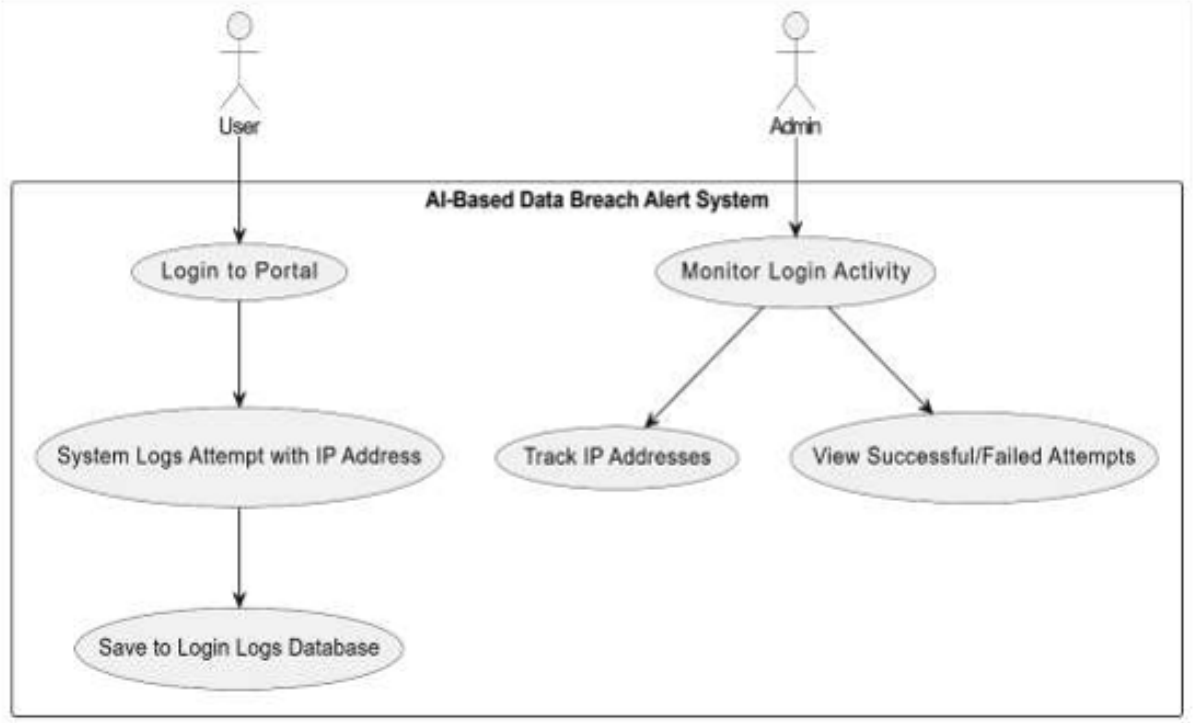

Figure 3.2: Use-Case Diagram 01: User Registration

3.10.3 Send Email

Table 3.4: Use-Case 02: Send Email

Use-Case ID	UC02
Use-Case Name	Send Email
Priority	High
Primary Actor	User
Description	User sends an email. AI classifies it and notifies admin.
Basic Flow	Email is sent; the AI classifies it and updates the admin.
Alternative Flow	Email is flagged as spam; admin is notified.
Pre-condition	User is logged into the portal.
System Reaction	Email is analyzed and classified.
Post-condition	Admin sees the classification result.

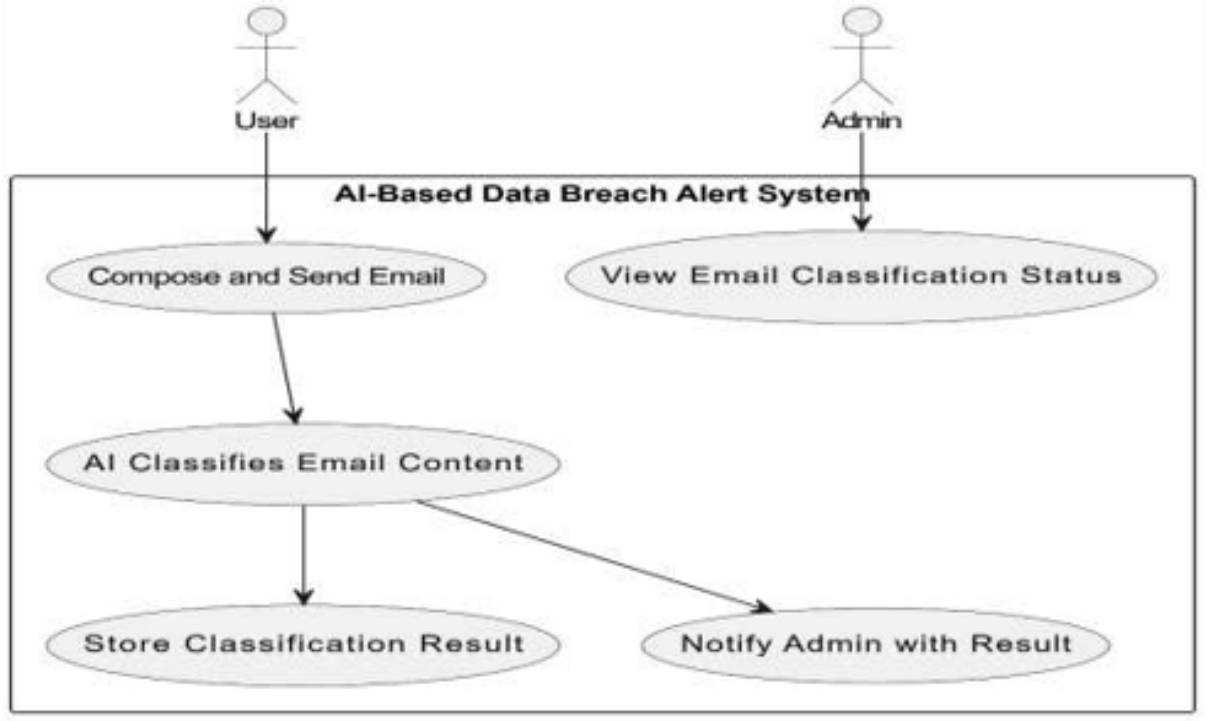

Figure 3.3: Use-Case Diagram 02: Send Email

3.10.4 Send Text Message

Table 3.5: Use-Case 03: Send Text Message

Use-Case ID	UC03
Use-Case Name	Send Text Message
Priority	High
Primary Actor	User
Description	User sends a message; AI classifies it and alerts admin if needed.
Basic Flow	Message is sent; AI classifies it.
Alternative Flow	Flagged as irrelevant; admin is alerted.
Pre-condition	User must be logged in.
System Reaction	Message is classified and stored.
Post-condition	Admin sees message classification.

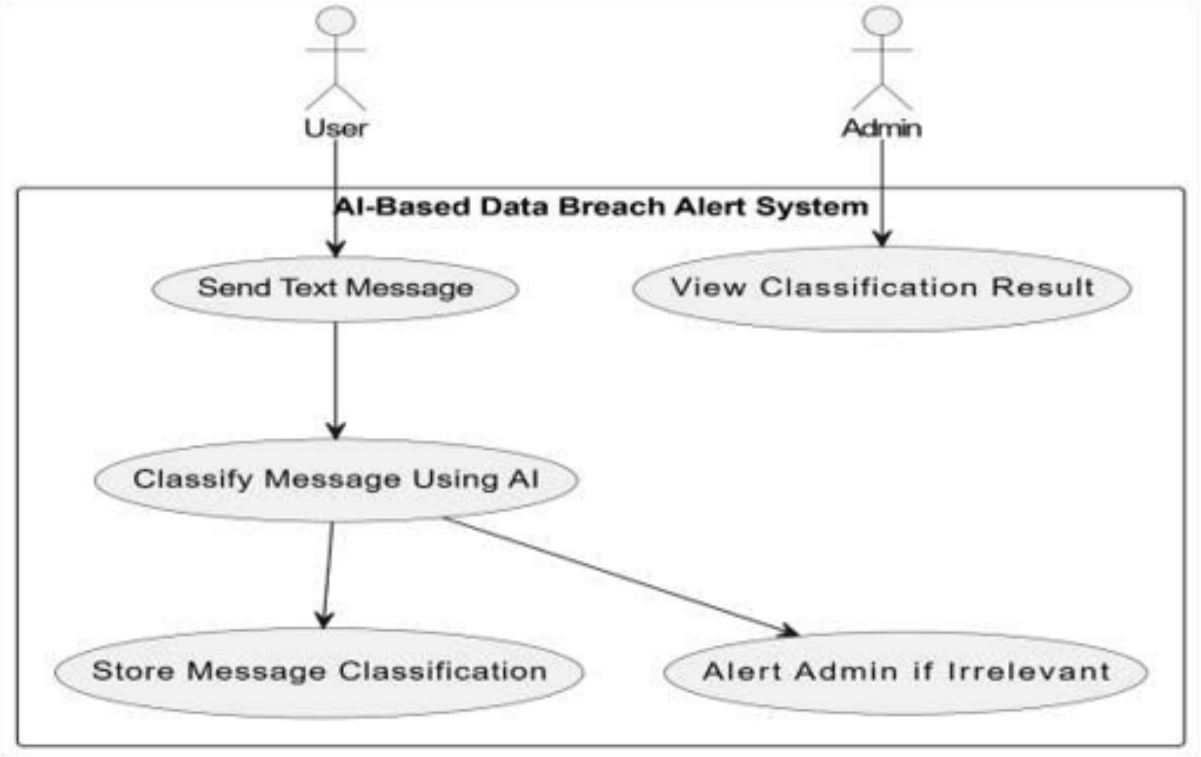

Figure 3.4: Use-Case Diagram 03: Send Text Message

3.10.5 Send Voice Message

Table 3.6: Use-Case 04: Send Voice Message

Use-Case ID	UC04
Use-Case Name	Send Voice Message
Priority	High
Primary Actor	User
Description	Voice input is transcribed and classified for admin monitoring.
Basic Flow	User records and sends voice; system transcribes and classifies.
Alternative Flow	Transcription fails; retry option is given.
Pre-condition	User provides voice input.
System Reaction	Transcription is generated and classified.
Post-condition	Admin sees voice message content and classification.

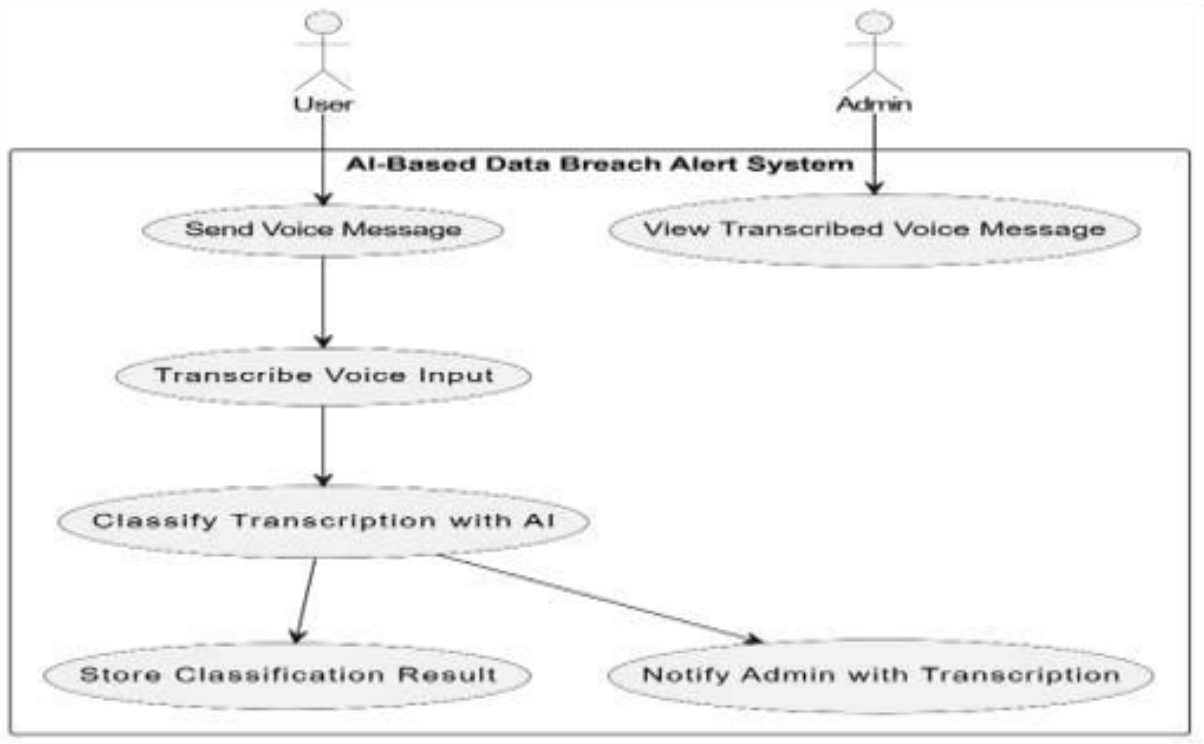

Figure 3.5: Use-Case Diagram 04: Send voice Message

3.10.6 Upload Pdf

Table 3.7: Use-Case 05: Upload PDF

Use-Case ID	UC05
Use-Case Name	Upload PDF
Priority	Medium
Primary Actor	User
Description	User uploads PDF; admin reviews it.
Basic Flow	Users upload documents; system stores and displays them to admin.
Alternative Flow	File rejected due to format error.
Pre-condition	User is logged in.
System Reaction	Stores and presents file to admin.
Post-condition	PDF is uploaded and logged.

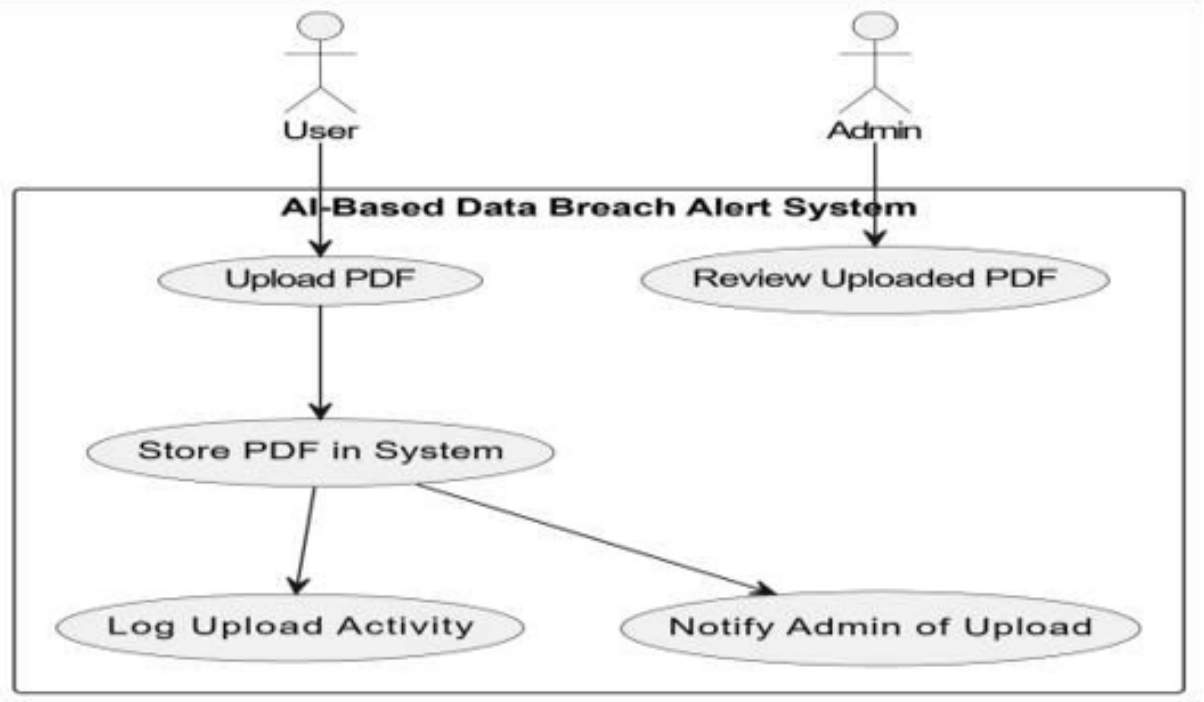

Figure 3.6: Use-Case Diagram 05: Upload Pdf

3.10.7 Download Pdf with Reason

Table 3.8: Use-Case 06: Download PDF with Reason

Use-Case ID	UC06
Use-Case Name	Download PDF with Reason
Priority	High
Primary Actor	User
Description	User requests a PDF download with valid reason and IP check.
Basic Flow	The system verifies IP and reason, then sends request to admin.
Alternative Flow	Access denied due to invalid IP or reason.
Pre-condition	User is logged in with registered IP.
System Reaction	Sends download request for admin approval.
Post-condition	PDF downloaded or rejected.

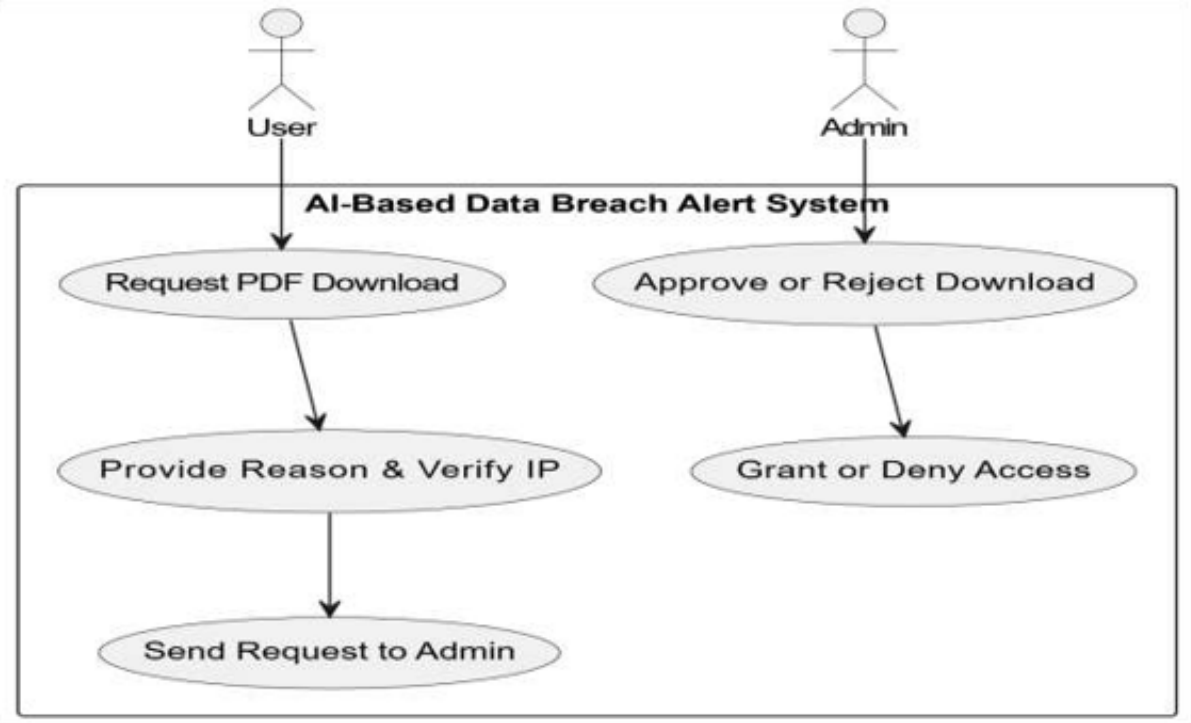

Figure 3.7: Use-Case Diagram 06: Download Pdf with Reason

3.10.8 Admin Dashboard Monitoring

Table 3.9: Use-Case 07: Admin Dashboard Monitoring

Use-Case ID	UC07
Use-Case Name	Admin Dashboard Monitoring
Priority	High
Primary Actor	Admin
Description	Admin monitors all system activities.
Basic Flow	Admin sees login logs, spam flags, and file requests.
Alternative Flow	None.
Pre-condition	Admin is logged in.
System Reaction	Fetches and displays logs.
Post-condition	Dashboard reflects real-time data.

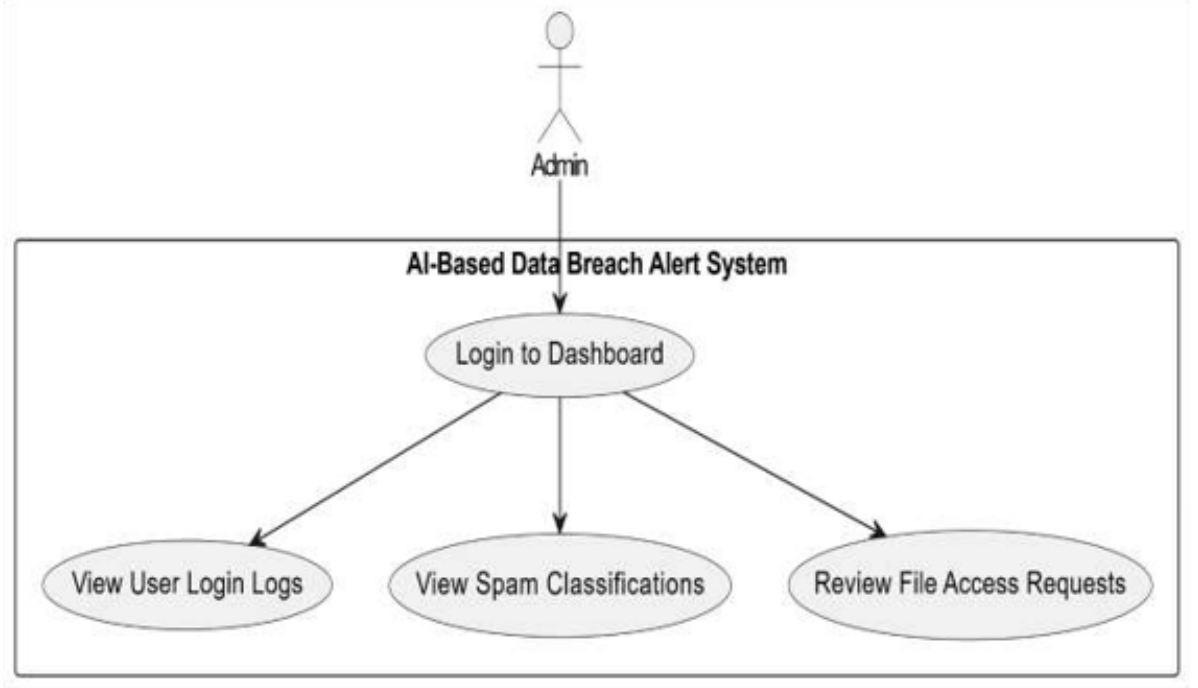

Figure 3.8: Use-Case Diagram 07: Admin Dashboard Monitoring

3.10.9 Voice to Text Transcription

Table 3.10: Use-Case 08: Voice-to-Text Transcription

Use-Case ID	UC08
Use-Case Name	Voice-to-Text Transcription Module
Priority	Medium
Primary Actor	System
Description	Transcribes user voice messages and outputs text.
Basic Flow	System receives voice, converts to text, and stores result.
Alternative Flow	Fails to transcribe; error message is shown.
Pre-condition	Voice message received.
System Reaction	Speech-to-text processing.
Post-condition	Text output stored for admin review.

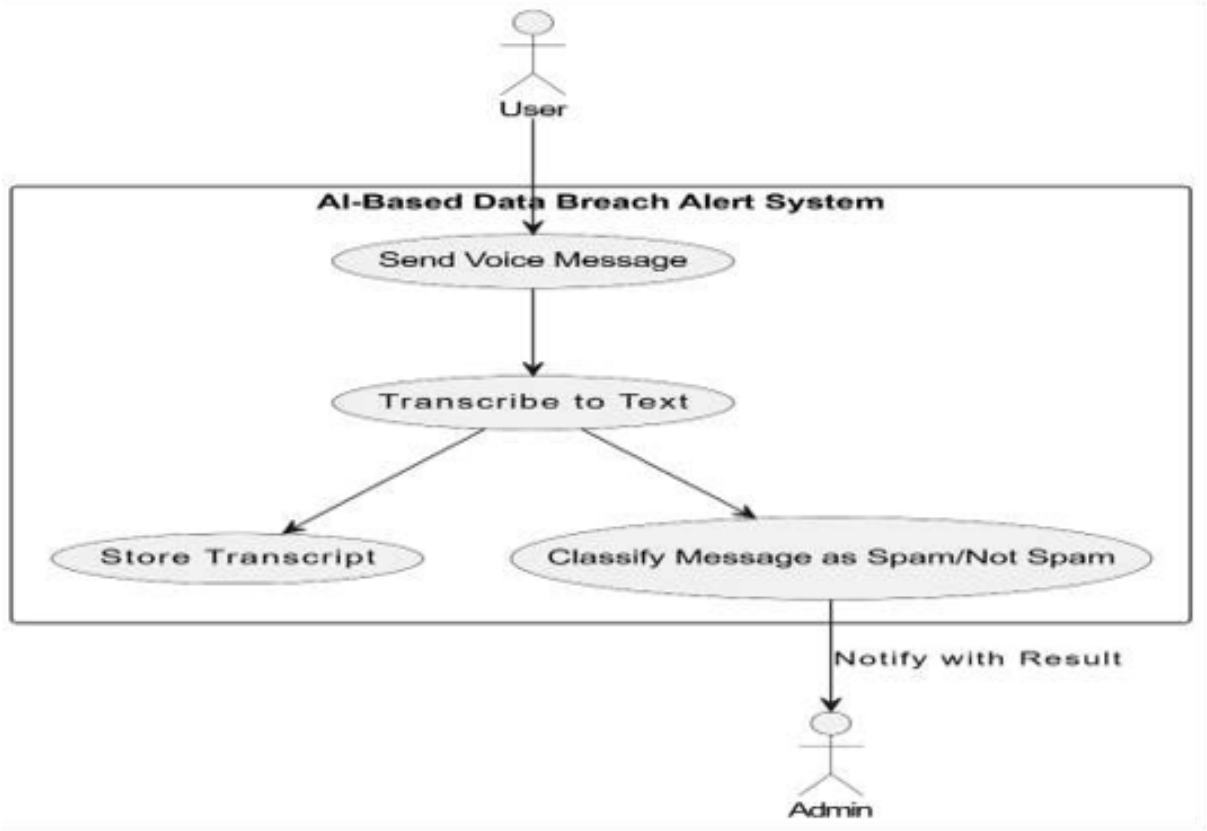

Figure 3.9: Use-Case Diagram 08: Voice to Text Transcription

3.10.10 AI Text Classification

Table 3.11: Use-Case 09: AI Text Classification

Use-Case ID	UC09
Use-Case Name	AI Content Classification Module
Priority	High
Primary Actor	System
Description	AI classifies content as 'Spam' or 'Not Spam'.
Basic Flow	Content is passed to the classifier; result is produced.
Alternative Flow	Uncertain classification; admin review required.
Pre-condition	Text content is available.
System Reaction	Runs classification model.
Post-condition	Classification result sent to admin.

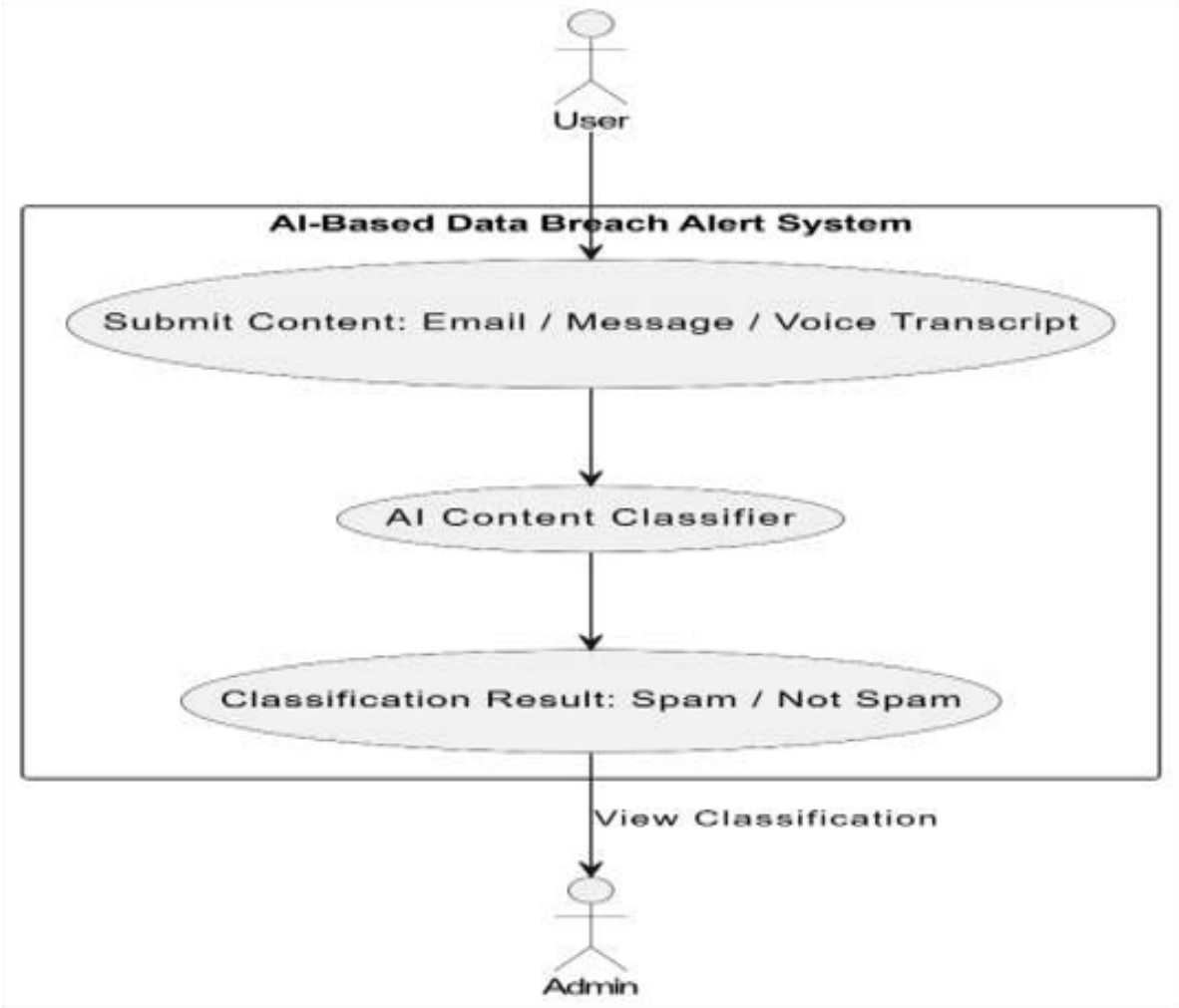

Figure 3.10: Use-Case Diagram 09: AI Text Classification

3.10.11 IP-Based Access Control

Table 3.12: Use-Case 10: IP-Based Access Control

Use-Case ID	UC10
Use-Case Name	IP-Based Access Control
Priority	High
Primary Actor	System
Description	Verifies access based on registered IP addresses.
Basic Flow	System checks IP before granting access.
Alternative Flow	Access denied due to invalid IP.
Pre-condition	User is logged in and requested a secure action.
System Reaction	Checks IP validity.
Post-condition	Grants or denies access.

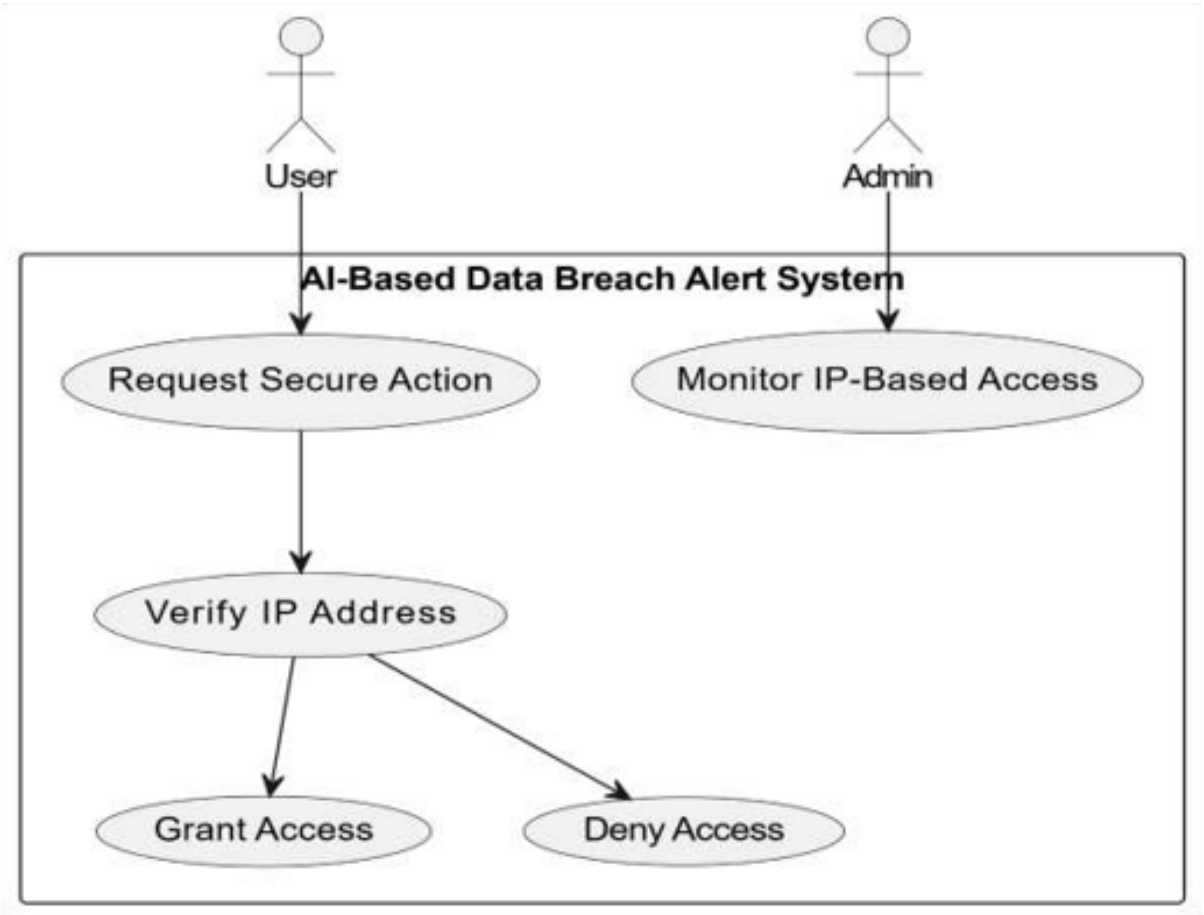

Figure 3.11: Use-Case Diagram 10: IP Based Access Control

Chapter 4

Design

This chapter focuses on the design and architecture of the proposed AI-Based Data Breach Alert System. The system architecture, graphical user interface (GUI) design, design constraints, system modeling, and structural layout are all covered in detail. In order to provide real-time monitoring, content classification, IP-based control, and secure data access in a telecommunications environment, the chapter describes how system components such as user portals, admin dashboards, AI modules, and database interaction collaborate.

4.1 System Architecture

4.1.1 Frontend

- **React.js (v19):** React is a modern JavaScript library for building dynamic and reusable user interface components. Modular development is made possible by its component-based architecture. React makes rendering efficient and helps in managing UI state seamlessly.
- **JavaScript (ES6+) + JSX:** JavaScript drives the application's frontend logic and interactive features. Modern syntax and features for cleaner, more effective code are provided by ES6+. JSX makes it possible to write HTML-like syntax inside JavaScript, which facilitates component rendering.
- **Vite (v6):** Vite is a fast and modern build tool that replaces traditional bundlers like Webpack or Create React App. It offers hot module replacement for quick updates during development. Vite also provides efficient bundling and optimized builds for production.
- **Tailwind CSS (v4):** Tailwind CSS is a utility-first framework for designing responsive and modern UI components. It allows developers to apply pre-defined classes without writing custom CSS. This results in faster development and consistent design across the application.

4.1.2 Backend

- **Python:** This project uses Python, a flexible programming language, to create backend logic and integrate AI and ML.
- **Flask:** Flask is a lightweight Python web framework for building RESTful APIs that facilitate front-end and back-end communication.
- **LSTM Model:** The LSTM (Long Short-Term Memory) model is used to detect spam in messages and emails.

- **Whisper Model:** Whisper is an AI model used for transcribing audio messages into text.
- **PDF Processing Service:** This service uses programming to handle and validate PDF files.
- **TensorFlow/Keras:** Libraries used for loading and running the LSTM model
- **Pickle:** Pickle is used to load the tokenizer object required for preprocessing text.
- **NumPy and Scikit-learn:** Libraries used for text preprocessing and evaluation metrics.

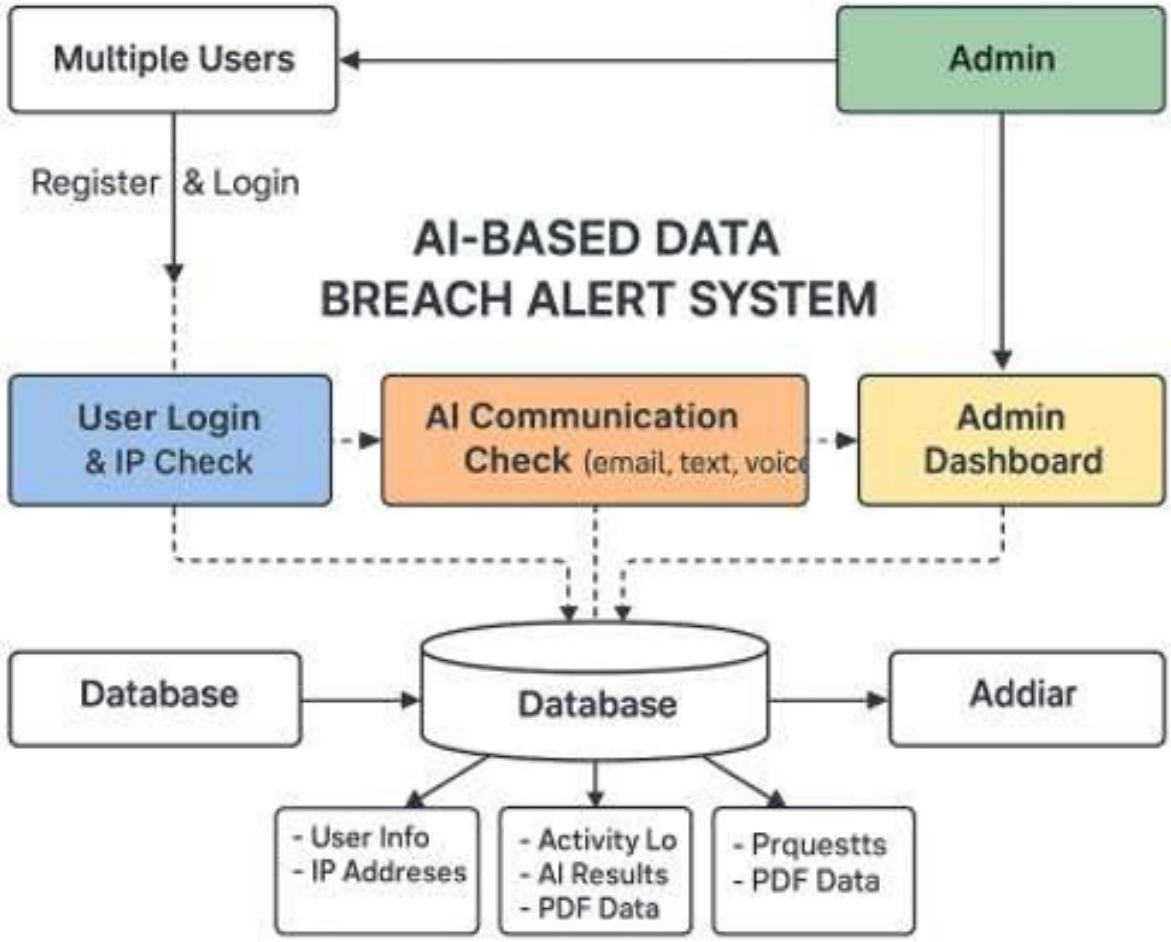

Figure 4.1: System Architecture Diagram

Database:-

- **SQL / SQLite:** SQL (using SQLite) is used to store structured data such as user activity logs and system records.
- **MongoDB:** MongoDB is a NoSQL database used to store flexible or unstructured data like messages, emails, and audio metadata.

4.2 System Modules

4.2.1 Login Activity Monitoring

- Users log in using registered IP addresses.
- The system tracks successful and failed login attempts.

4.2.2 Communication Monitoring

- **Email Sending**
  - Users send emails through the portal.
  - AI module classifies content as “Spam” or “Not Spam”.
  - Admin is notified if content is irrelevant to office work.
- **Text Message Sending**
  - Text messages are sent within the portal and analyzed for relevance.
  - AI classifies and flags inappropriate content to the admin.
- **Voice Message Sending**
  - Users send voice messages.
  - System transcribes voice using Speech-to-Text.
  - AI classifies transcription and alerts admin if required.

4.2.3 PDF File Handling

- Users can upload PDF documents.
- Other users can download files only after:
  - Providing a valid reason.
  - Logging in from a registered IP.
  - Receiving admin approval.

4.2.4 Admin Module

The admin has a central monitoring and control interface.

- **Activity Tracking**
  - Monitor all login/logout attempts.
  - View IP addresses associated with user activity.
  - Review successful and failed login attempts.
- **AI-Based Classification Monitoring**
  - View spam and non-spam classification results for:
    - Emails
    - Text messages
    - Transcribed voice messages
- **Document Access Control**

  - Approve or deny PDF download requests.
  - View uploaded documents.

- **System Analytics and Dashboard**
  - Generate and view pie charts and metrics for:
    * Login activity
    * Spam vs. Not Spam messages
    * File access requests
    * User distribution by IP/location

4.2.5 Feedback Module

- **User Feedback Collection**
  - Users rate the system (1 to 5).
  - Submit comments about their experience.

- **Admin Feedback Dashboard**
  - View and analyze user feedback.
  - Display overall system rating using visual charts.
  - Access history of past user comments for performance review.

4.3 Database Design

4.3.1 Table: admin_credentials

- **Purpose:** Stores credentials and identity information of the admin user.
- **Key Columns:**
  - admin_id (Primary Key)
  - username
  - password_hash
  - email
- **Function:** Used for secure admin login and session validation.

4.3.2 Table: user_data

- **Purpose:** Logs and maintains all user activities and communication logs.
- **Key Columns:**
  - user_id (Primary Key)
  - username
  - registered_ip
  - login_timestamp
  - email_sent
  - text_message

  - voice_transcription
  - content_type (e.g., Email, Text, Voice)
  - classification (Spam / Not Spam)

- **Function:** Tracks user actions, communications, and AI classification results. Important for audit and breach detection.

4.3.3 Table: document_access_log

- **Purpose:** Tracks all document (PDF) uploads and download requests.
- **Key Columns:**
  - request_id (Primary Key)
  - uploader_id
  - file_name
  - download_requester_id
  - request_reason
  - ip_address
  - admin_approval (Approved / Rejected)
- **Function:** Ensures that sensitive documents are accessed under secure and verified conditions.

4.3.4 Table: user_feedback

- **Purpose:** Stores system feedback from users for system evaluation and future improvement.
- **Key Columns:**
  - feedback_id (Primary Key)
  - user_id
  - rating (1–5)
  - comments
  - timestamp
- **Function:** Allows the admin team to review user satisfaction and identify improvement areas.

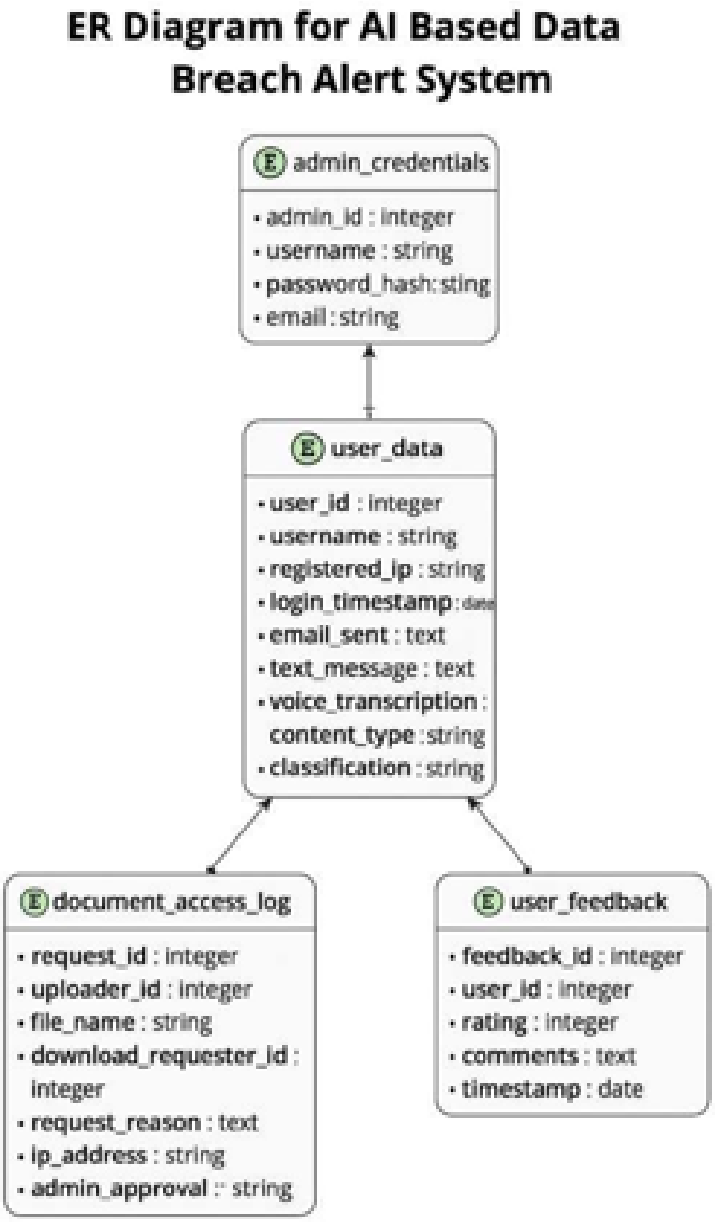

Figure 4.2: Entity Relationship Diagram

4.4 Class Diagram

The class diagram illustrates how various components of the AI-Based Data Breach Alert System interact with each other during system execution.

The process begins when a user logs into the system. The user's IP address is obtained and stored by the system upon login, and it is subsequently utilized to ascertain the location (city, state, or nation). Every login attempt, whether successful or unsuccessful, is monitored and documented in the system.

After that, users can use the secure portal to carry out several tasks:

- Send text messages, voice messages, and emails.
- Upload PDF documents, which are stored securely along with logged metadata.

- If another user wishes to download a PDF:
  - They must be on a registered office IP.
  - They must provide a valid reason for downloading.
  - A download request is sent to the admin for approval.

The Admin Dashboard monitors:

- All login/logout activities.
- Failed login attempts.
- AI classification results (Spam/Not Spam).
- File upload/download history.
- Voice-to-text transcription logs.

All this information is logged and stored in the database, ensuring transparency, accountability, and proactive breach detection.

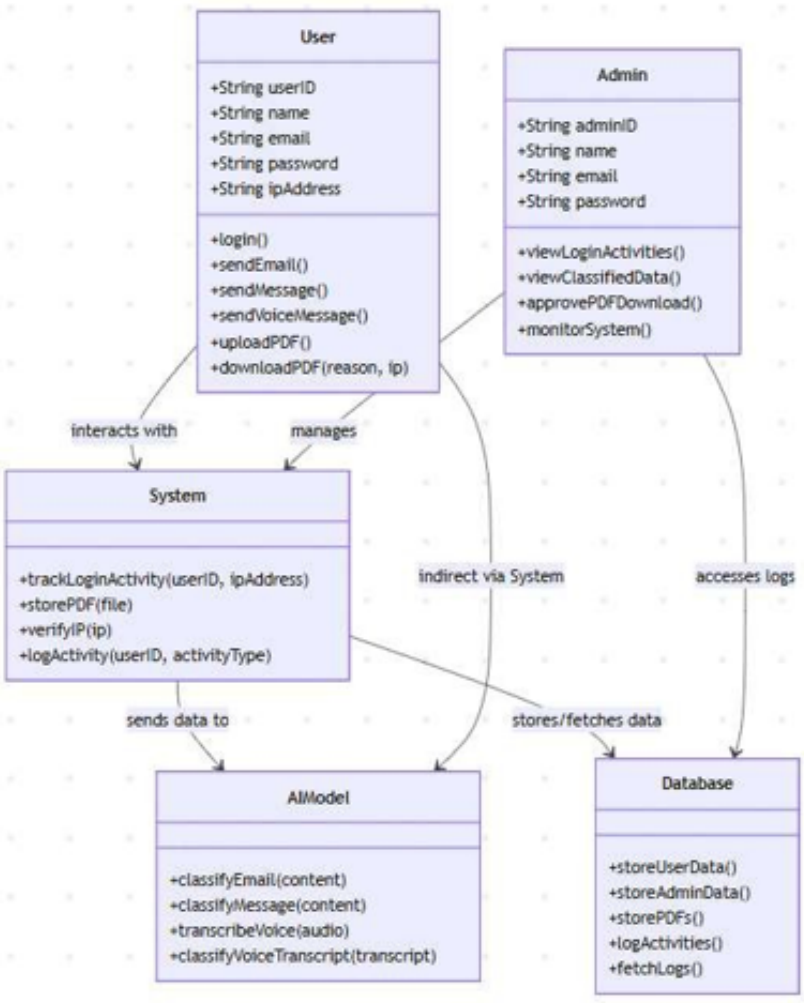

Figure 4.3: Class Diagram

4.5 Event Table

Table 4.1: Event Table

EVENT	TRIGGER	SOURCE	USE CASE	RESPONSE	DESTINATION
User wants to send an email	Send Email	User	Classify Email	Classified as Spam or Not Spam	Admin Dashboard
User wants to send text message	Send Text Message	User	Classify Text	Classified as Spam or Not Spam	Admin Dashboard
User sends voice message	Send Voice	User	Transcribe & Classify Voice	Transcription & Classification Result	Admin Dashboard
User uploads a PDF	Upload PDF	User	Upload PDF	Stored & Visible to Admin	Admin Panel
User requests to download PDF	Download Request	User	Validate & Forward Request	Approved/ Rejected by Admin	User/Admin
Admin logs in	Admin Login	Admin	Monitor Activities	Show Logs and Analytics	Admin Dashboard
System checks IP	Secure Action	System	IP Verification	Access Granted or Denied	System

4.6 High Level Design

How various actions and interfaces will interact with one another in detail is depicted at a high-level design.

4.7 Level-0 Data Flow Diagram

The Business Logic Layer handles and processes user input when a user interacts with the software via the provided interface. in which a controller function is carried out.

The logic layer analyses user input and executes the necessary steps. For instance, a user may provide necessary information. This is handled by the Presentation Layer, which sends the user input to the Logic Layer, which then sends the specified query to the NLP Model. The NLP Parser/Analysis Layer sends the analysis to the user.

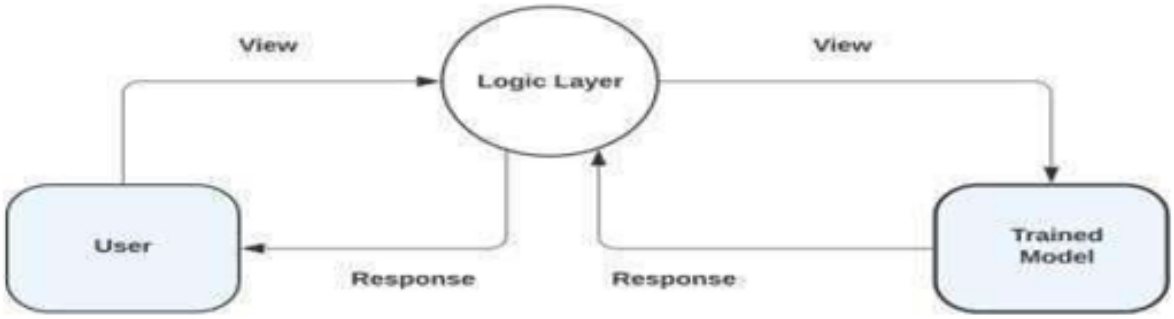

Figure 4.4: Context Diagram

4.8 System Flow

The working of the system is proposed in following figure. The flowchart illustrates the UX design of the application. The user begins by authenticating through the sign-up or login process. Upon successful login, the user is assigned a specific system IP address, which is used for tracking their activity within the office network. After providing all necessary credentials, the user is redirected to the main dashboard where information about the system's features and functionalities is available. From there, the user can perform activities such as sending emails, text messages, voice messages, or uploading PDF files. Each activity is monitored by the system. The AI model analyzes emails, messages, and transcribes voice data to classify whether the content is related to official work. If any content is flagged as non-official or suspicious, it is immediately reported to the admin dashboard as "Spam." The admin has full visibility of login activities, failed login attempts, file access requests, and AI-based classifications, enabling real-time breach detection and security enforcement.

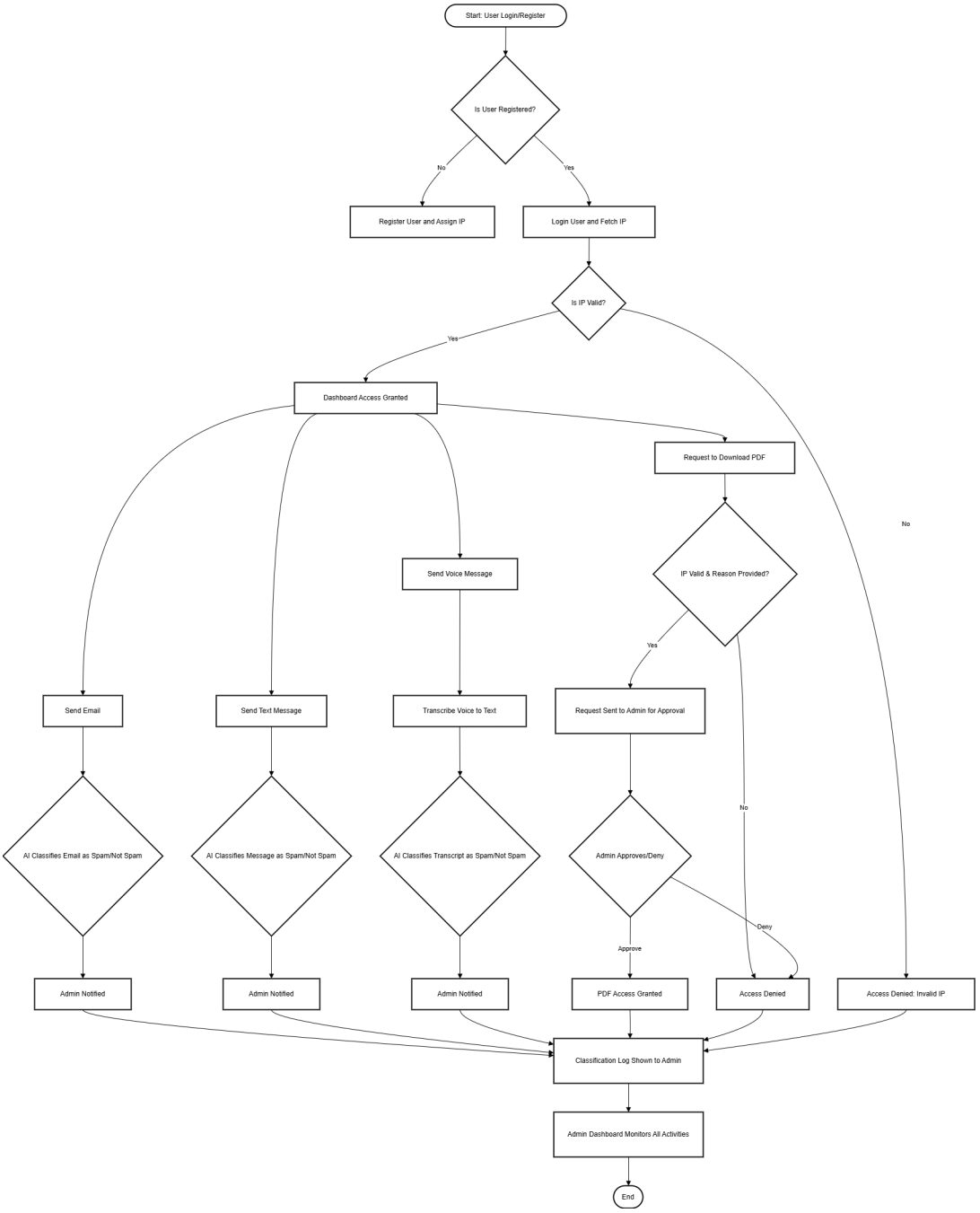

Figure 4.5: System Flow Diagram

4.9 Data Flow Diagram

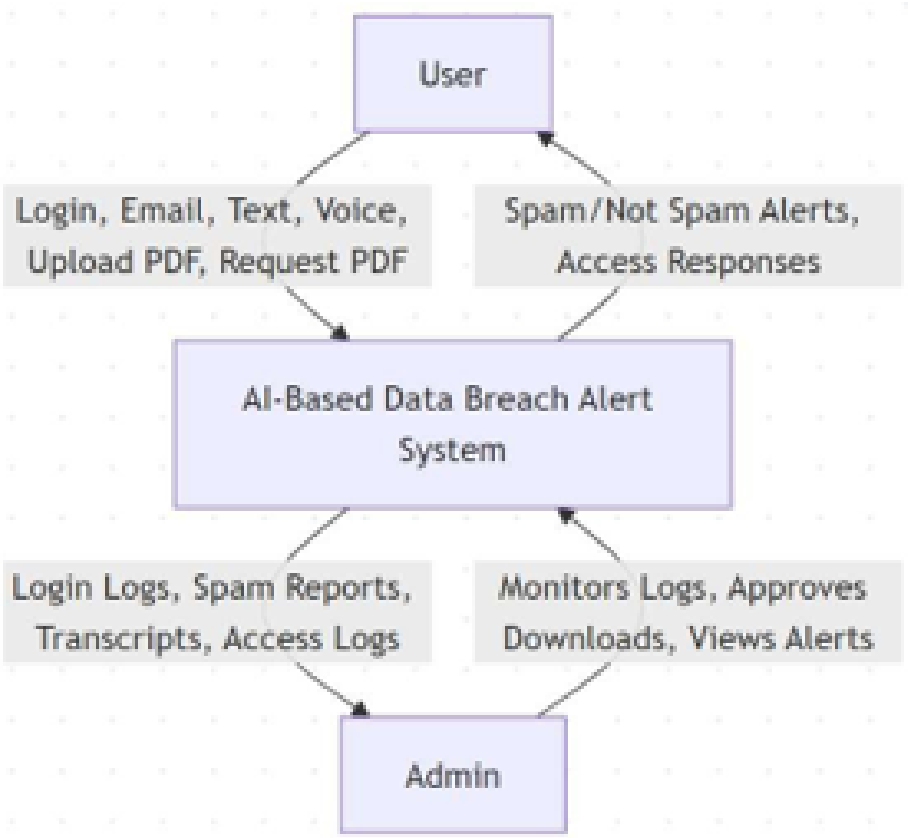

Figure 4.6: Data Flow Diagram (Non Technical)

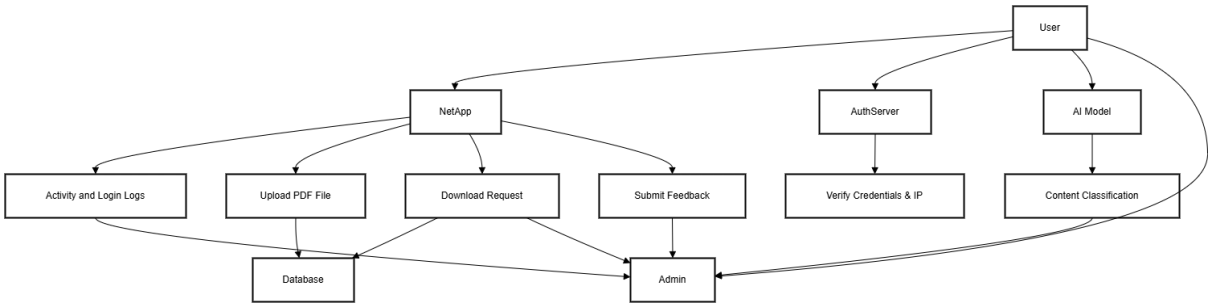

Figure 4.7: Data Flow Diagram (Technical)

4.10 Sequence Diagram

The system is composed of multiple objects that communicate with each other. The interaction begins when the user attempts to log into the web portal. The system checks the user's credentials and verifies the IP address from the database. If the credentials and IP match, the user is granted access to the portal.

After logging in, the user can upload voice messages, send emails, and send texts. The AI module intercepts each of these actions and determines whether the content is "Spam" or "Not Spam." The admin dashboard receives a notification if the content is marked as spam.

Additionally, the user may upload a PDF file or request to download a PDF with a reason. The system checks the IP and validates the reason before granting access. All activities performed by users are logged and viewable by the admin on their dashboard in real time.

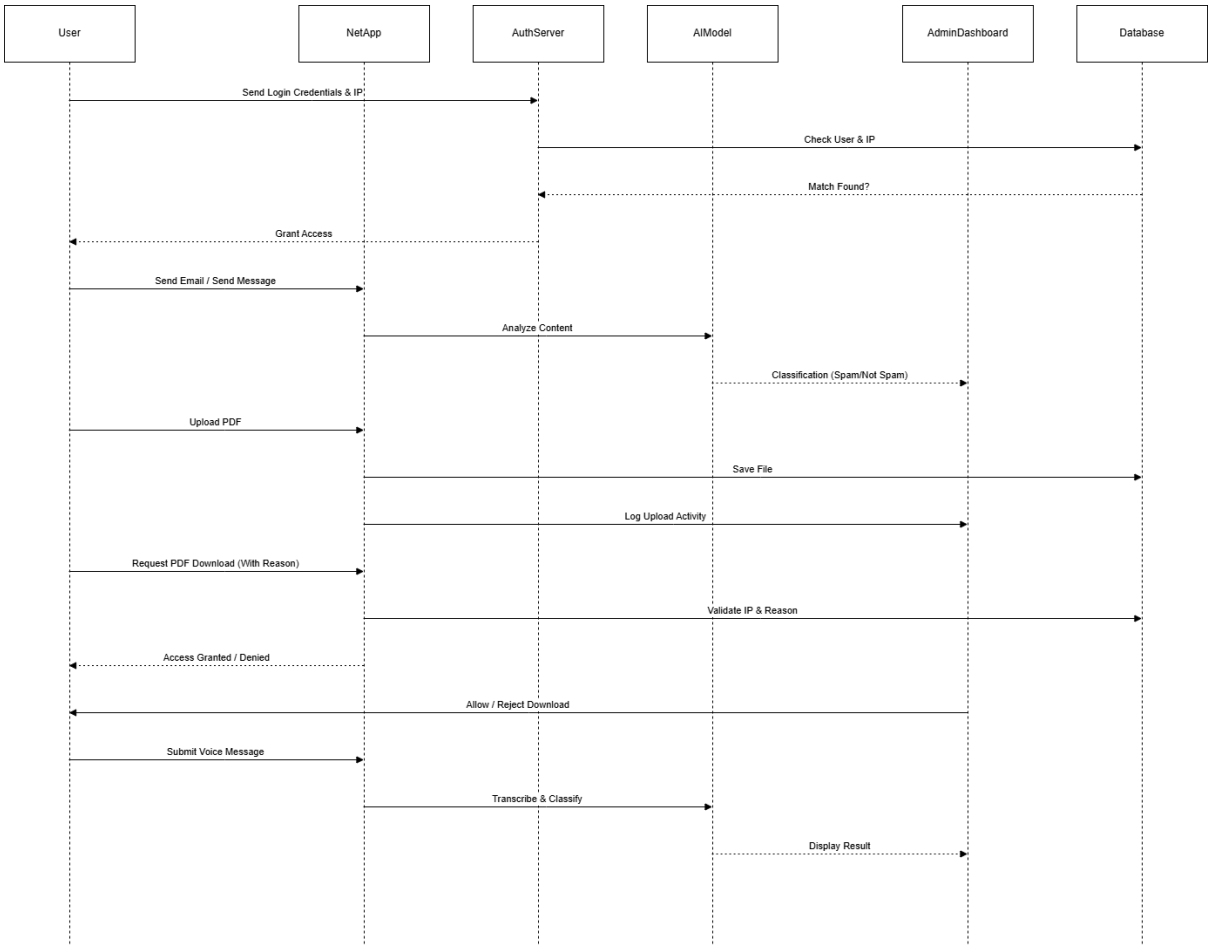

Figure 4.8: Sequence Diagram

4.10.1 Login Sequence Diagram

The user begins by entering their credentials on the login page. The system validates the email, password, and associated IP address. If all inputs are valid and the IP is recognized from the database, the login is successful, and the user is redirected to the main dashboard. If any of the inputs are incorrect (email/password or IP), the system denies access and logs the failed attempt. The admin is also notified of the failed login for monitoring.

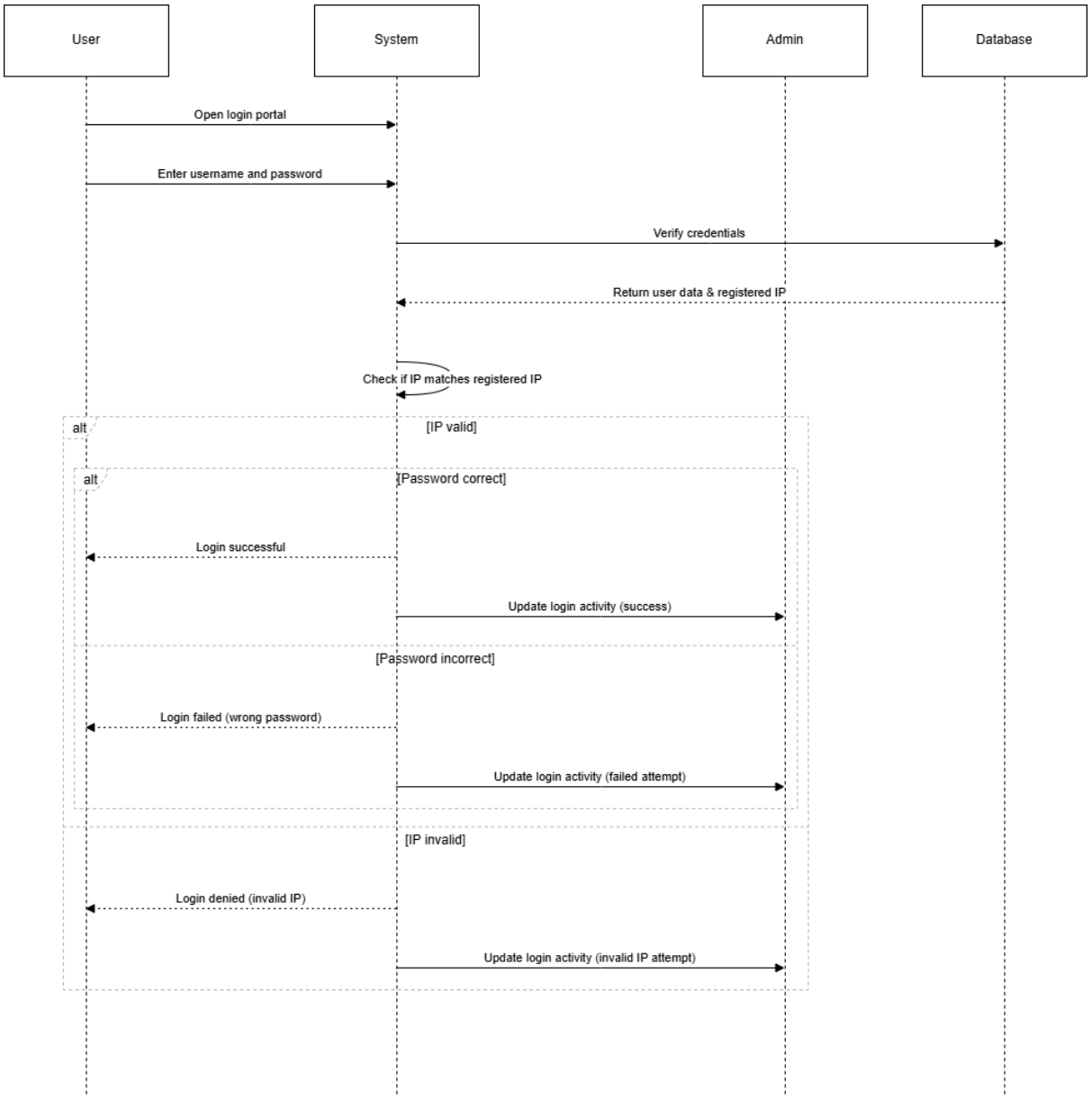

Figure 4.9: Login Sequence Diagram

4.11 GUI Design

The graphical user interface (GUI) of the AI-Based Data Breach Alert System is designed to ensure ease of use and smooth interaction for both admin and users. The system's navigation flow is kept intuitive, allowing users to access key functionalities such as login, message/email sending, and PDF uploads/downloads without requiring technical expertise. Since the interface serves as the primary point of interaction for end users, the GUI design plays a critical role in ensuring efficiency, usability, and trustworthiness. A clean and responsive design enhances the performance and acceptance of the proposed system within the telecommunication environment.

4.11.1 Admin Side Interfaces

4.11.1.1 Admin Login Screen

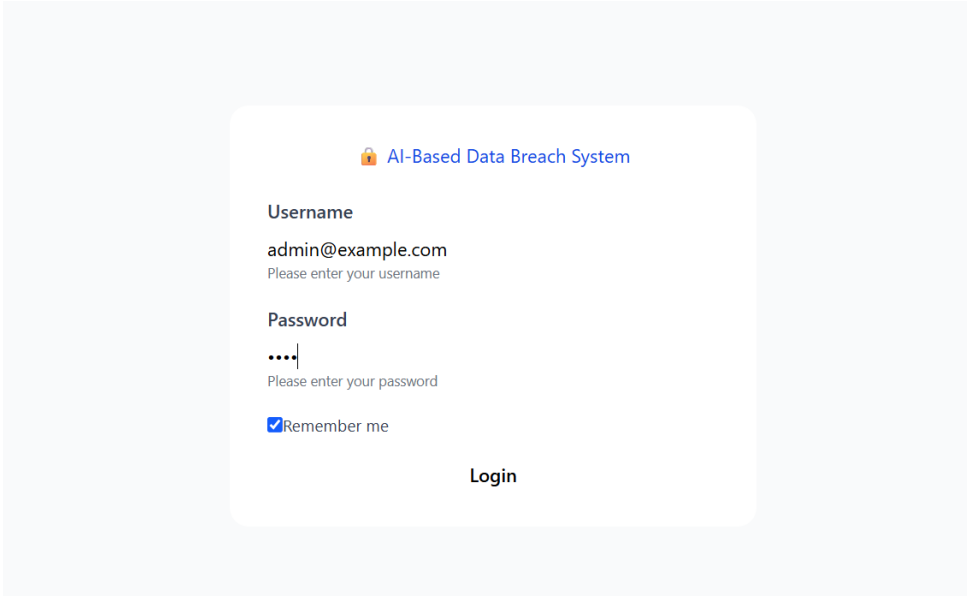

Figure 4.10: Admin Login Screen

4.11.1.2 Admin Dashboard Screen

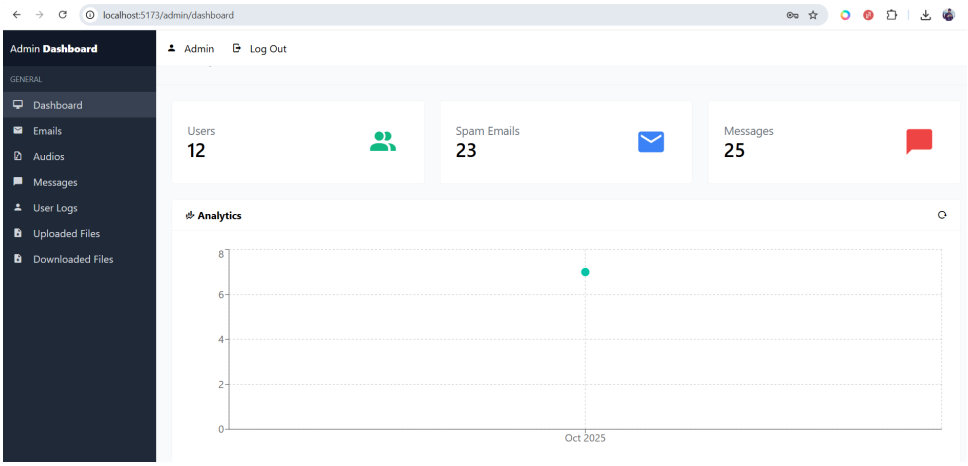

Figure 4.11: Admin Dashboard Screen

4.11.1.3 Admin Logs Screen

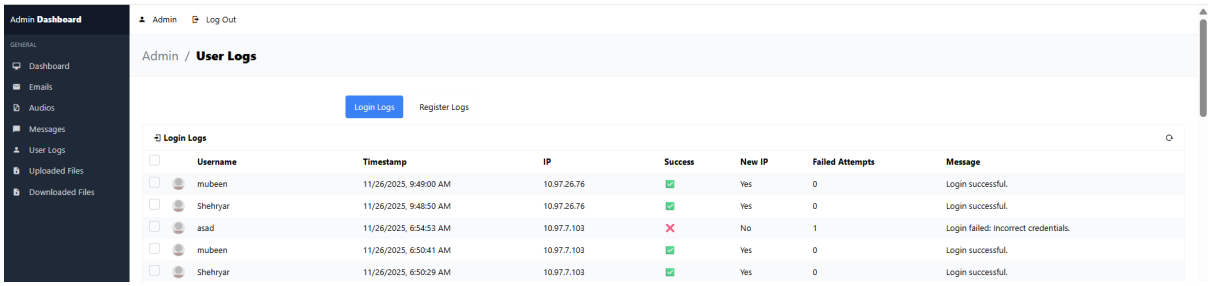

Figure 4.12: Admin Logs Screen

4.11.1.4 Admin Uploaded Files Screen

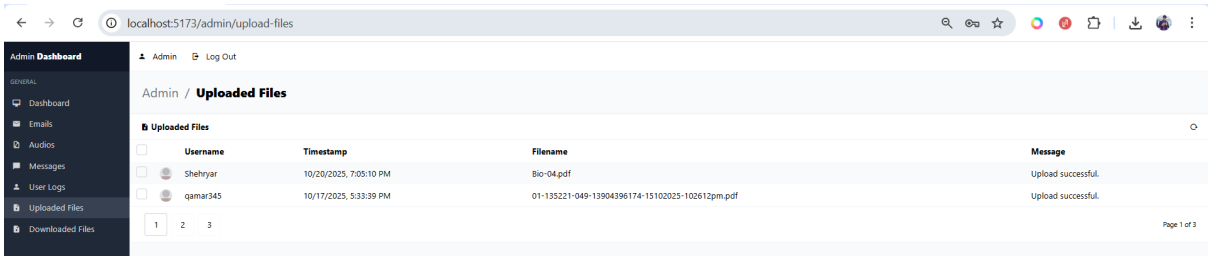

Figure 4.13: Admin Uploaded Screen

4.11.1.5 Admin Downloaded Files Screen

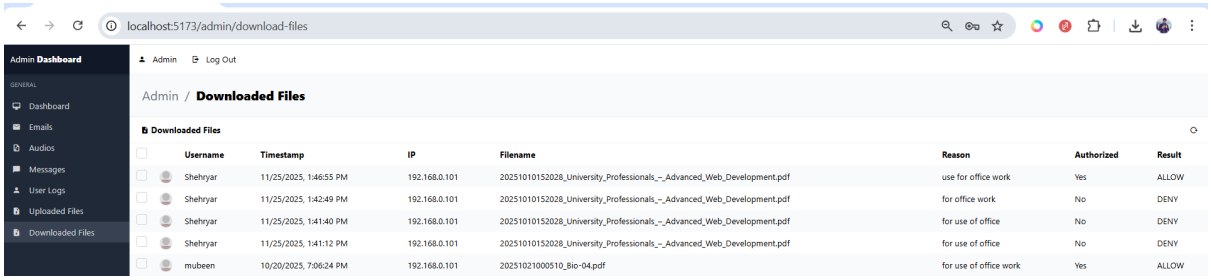

Figure 4.14: Admin Downloaded Files Screen

4.11.1.6 Admin Emails Screen

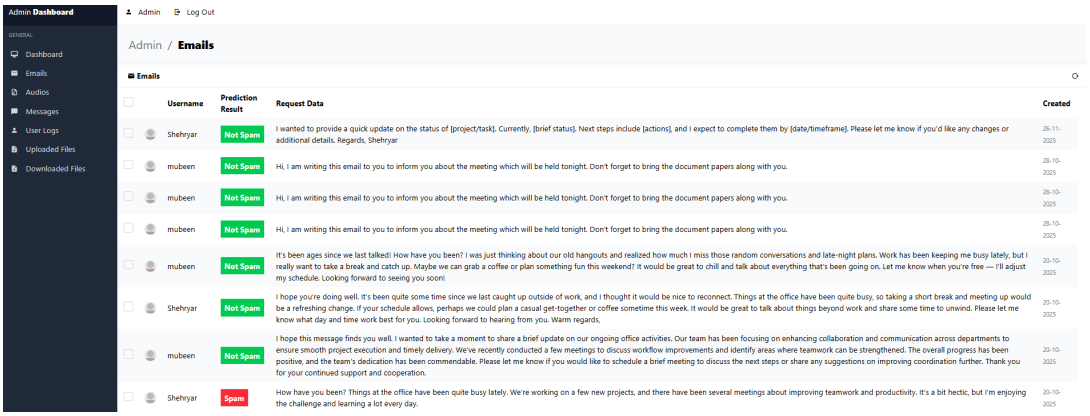

Figure 4.15: Admin Emails Screen

4.11.1.7 Admin Text Messages Screen

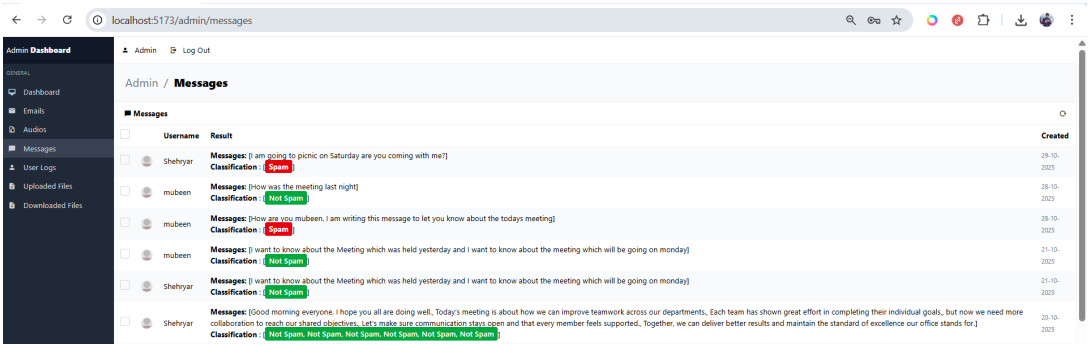

Figure 4.16: Admin Text Messages Screen

4.11.1.8 Admin Voice Messages Screen

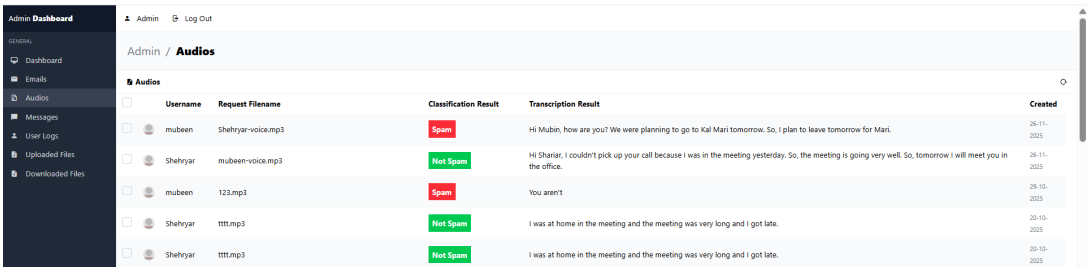

Figure 4.17: Admin Voice Messages Screen

4.11.2 User Side Interfaces

4.11.2.1 User Login Screen

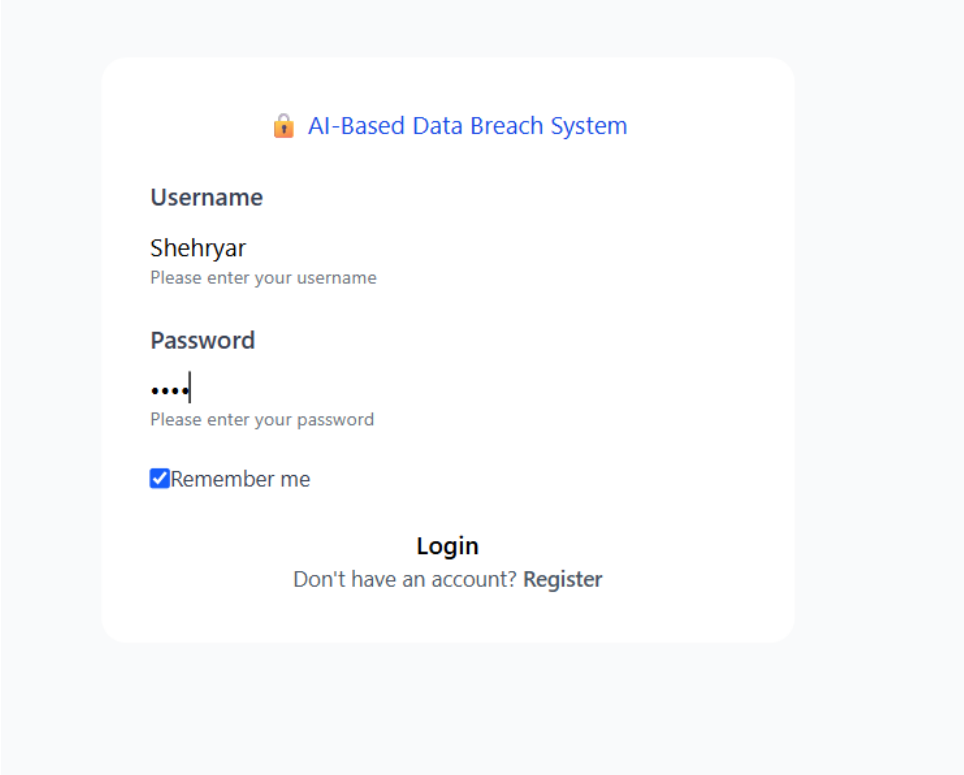

Figure 4.18: User Login Screen

4.11.2.2 User Uploaded Files Screen

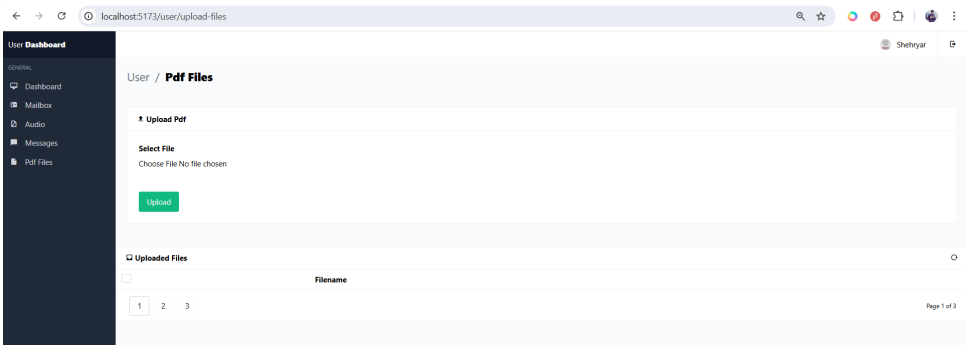

Figure 4.19: User Uploaded Files Screen

4.11.2.3 User Emails Screen

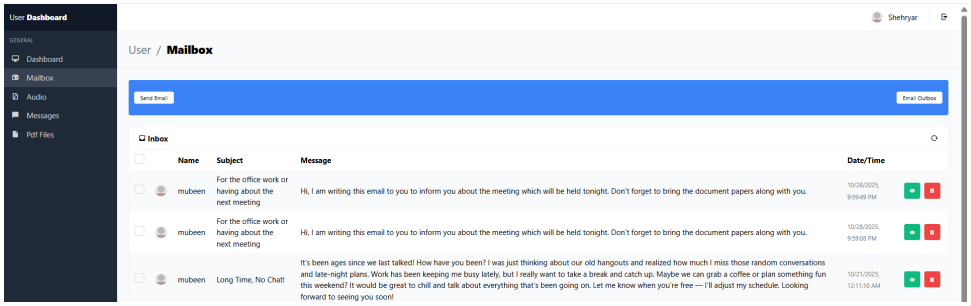

Figure 4.20: User Emails Screen

4.11.2.4 User Text Messages Screen

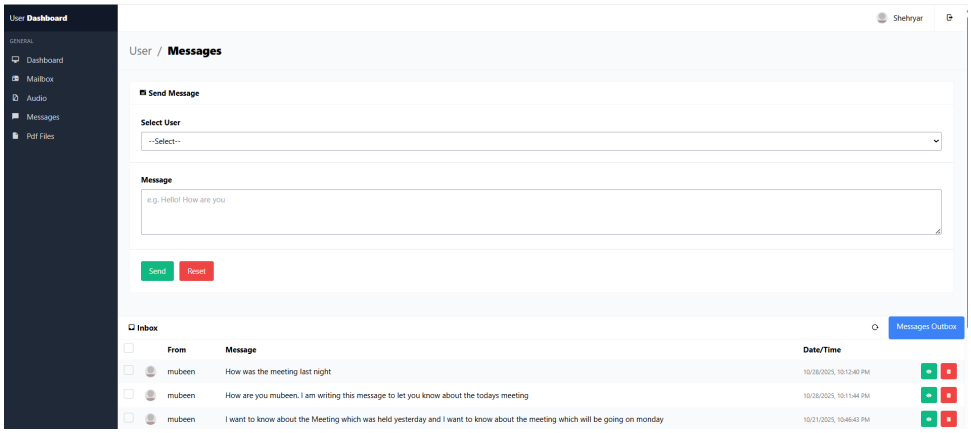

Figure 4.21: User Text Messages Screen

4.11.2.5 User Voice Messages Screen

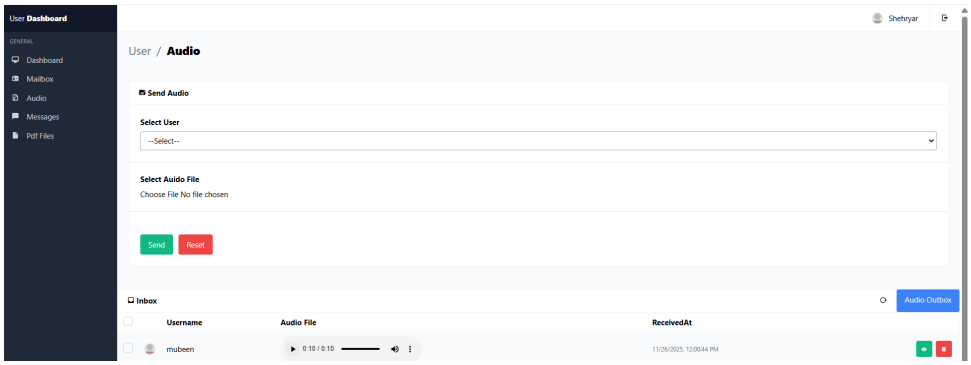

Figure 4.22: User voice Messages Screen

Chapter 5

System Implementation

5.1 System Architecture

The system architecture of the AI-based Data Breach Alert System is designed to provide efficient monitoring and alerting functionalities to safeguard sensitive information within a telecommunication company. The architecture consists of several internal components that interact with each other to ensure smooth operation and security.

5.1.1 System Internal Components

The Admin Panel serves as the central hub where the administrator monitors and manages the activities of all users. It displays login activities, incorrect password attempts, email communications (classified as "Spam" or "Not Spam"), voice messages, and PDF download requests. Each user has a portal to send emails, messages, and voice messages. This portal communicates with the backend system to allow for AI-based spam detection and activity tracking. The AI Spam Detection Engine analyzes the content of emails, messages, and voice messages to classify them as "Spam" or "Not Spam" based on predefined office work parameters.

5.1.2 Functionality of the Components

The Admin Panel provides the admin with a real-time dashboard that tracks user login activities, email exchanges, message content, and PDF download activities. It also displays alerts for any unauthorized activities or breach attempts. The AI Spam Detection Engine leverages machine learning algorithms to predict the relevance of user communications, flagging suspicious messages or emails as potential security risks. MongoDB is used to store and retrieve user data, activity logs, and system alerts. It ensures fast and reliable storage of large volumes of data while also providing scalability.

5.1.3 Communication Between Components

User Portal communicates with the AI Spam Detection Engine to analyze emails and messages. It also interacts with MongoDB to store user details and activity logs. The Admin Panel pulls data from MongoDB to generate real-time reports and alerts. Each user's action (login, email sending, message writing, etc.) is captured and sent to MongoDB. If a user attempts to send a message, the AI engine processes the content before storing the outcome in the database and notifying the admin.

5.1.4 Tools and Technology Used

- **Languages Used:** The system is developed using a combination of Python (for AI modeling and backend logic), JavaScript (for frontend user interface), and Node.js (for

server-side communication).

- **AI Tools:** TensorFlow or scikit-learn are used for building and training the AI model for spam detection.
- **Database:** MongoDB, a NoSQL database, is used to store user data, logs, and system configurations.
- **Frameworks:** The backend is implemented using Express.js (for building APIs), while the frontend uses React.js for creating a dynamic and responsive user interface.

5.2 Application Access Security

The security of the application is a critical aspect, particularly since the system handles sensitive user data and activities. Several measures have been implemented to ensure data integrity and prevent unauthorized access. The system is deployed within a secure network that is protected by firewalls. These firewalls ensure that only authorized IP addresses can access the internal systems, preventing external attacks from unknown sources. Data is transmitted between the client (user or admin) and the server is encrypted using SSL/TLS protocols to protect against interception during transmission.

- **Password Encryption:** User passwords and admin credentials are securely stored in the database using bcrypt or a similar hashing algorithm to ensure that plaintext passwords are not exposed in the system.
- **Authentication:** The system employs username and password authentication for both users, and the admin. Password complexity rules are enforced, such as requiring a minimum length and the inclusion of special characters, to prevent weak passwords. Additionally, IP-based authentication ensures that users are only allowed to perform certain actions, such as downloading PDFs, if they log in from their registered IP addresses. This adds an extra layer of security by preventing unauthorized access.
- **Authorization:** The system defines roles for users (e.g., standard user and admin). Each role has specific permissions, with the admin having full control over the system, while regular users can only perform actions related to their accounts. The admin can monitor all activities, change user settings, and access all logs, while users can only access their own data and perform limited actions.
- **Auditing/Access Logging:** All user activities (login attempts, email and message communications, PDF download requests, etc.) are logged into the database. These logs are regularly reviewed by the admin to ensure there are no unauthorized or suspicious activities. The system also generates alerts for abnormal activities, such as multiple failed login attempts, and notifies the admin in real time
- **Safe Data Storage:** MongoDB ensures that sensitive user data, logs, and communication records are stored securely. Data access is restricted based on roles, and backup measures are implemented to prevent data loss.

5.3 Database Security

Since MongoDB stores sensitive data, security measures are crucial to protect against unauthorized access and ensure data integrity. Remote access to the database is restricted to authorized personnel only, using encrypted communication channels. MongoDB is configured to prevent unauthorized connections from unknown IP addresses. Authentication is required for every user or service attempting to interact with the database. The system enforces role-based access control (RBAC) to limit the actions each user can perform, such as reading, writing, and updating operations.

5.3.1 Auditing/Logging

MongoDB provides auditing capabilities, and the system logs all database operations. These logs are stored securely and can be reviewed by the admin to detect any suspicious or unauthorized actions.

5.3.2 Anonymous and Group Users

The system restricts access to the database to only authenticated users. There are no anonymous or group users with blanket access to sensitive data. This ensures that only users with proper authentication and authorization can interact with the database, minimizing the risk of unauthorized data access or manipulation.

Chapter 6

System Testing and Evaluation

System testing is a vital phase in the software development cycle, where the system is evaluated in its entirety to ensure it meets the defined requirements. For the AIbased Data Breach Alert System designed for telecommunication companies, system testing ensures the application functions as expected, adheres to security and usability standards, and performs efficiently under various conditions.

Merely building the system and documenting its design and features is insufficient. The system must be thoroughly evaluated both qualitatively and quantitatively. This evaluation process includes assessing functionality, user experience, and system performance, and comparing it to existing tools or systems. Understanding both the system's strengths and limitations is essential for producing a comprehensive assessment.

While no system is perfect, recognizing its shortcomings and addressing them is critical. This chapter delves into the various testing methods employed to evaluate the AI-based Data Breach Alert System, including:

6.1 Graphical User Interface (GUI) Testing

GUI testing ensures the user interface is functional, intuitive, and meets the expected design standards. For the AI-based Data Breach Alert System, GUI testing focused on evaluating both the admin dashboard and user portal. The primary goal was to verify that all visual elements, such as buttons, input fields, and navigation menus, work as intended. Additionally, the layout was tested to ensure the interface was clear, user-friendly, and responsive across different devices and screen sizes.

6.2 Usability Testing

Usability testing focuses on evaluating how easy and efficient the system is to use for both administrators and regular users. Testing involved observing real users interacting with the system to identify usability challenges, such as confusing workflows or hard-to-find features. For example, the usability of the admin dashboard for monitoring user activity, tracking login events, and generating alerts was examined. Similarly, the user portal for actions like sending emails, messages, and voice communications was tested. Feedback from real users helped identify areas for improvement, such as making the navigation more intuitive and reducing unnecessary steps.

6.3 Software Performance Testing

Performance testing assesses how the system operates under both normal and peak conditions. The AI-based Data Breach Alert System was tested to ensure it can handle a high volume of user logins, process large amounts of data, and generate alerts promptly without performance degradation.

Key performance aspects tested include:

1. **Response Time:** The system's responsiveness to user inputs such as login attempts, message sending, and alert generation was measured.
2. **Throughput:** The ability of the system to process large volumes of messages, emails, and login attempts was evaluated.
3. **System Stability:** The system's ability to maintain stability under continuous use and avoid crashes or slowdowns was monitored.

6.4 Compatibility Testing

Compatibility testing ensures that the system works seamlessly across various devices, operating systems, and browsers. This testing ensures that all users can access the AI-based Data Breach Alert System regardless of their configuration.

The system was tested on different browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge, to ensure functionality across popular platforms. Additionally, compatibility was verified for desktop computers, laptops, and mobile devices. The system was also tested on different operating systems, such as Windows and Mac OS, to ensure cross-platform compatibility.

6.5 Exception Handling

Testing exception handling verifies that the system can appropriately deal with unexpected situations or errors. This includes testing how the system reacts to incorrect user inputs, unexpected system failures, or other anomalies.

Tests focused on ensuring that:

1. **Error Messages:** The system provides clear and helpful error messages when users enter incorrect data or perform unsupported actions.
2. **System Recovery:** The system can recover from failures, such as invalid login attempts, without crashing.
3. **Graceful Failures:** When an error occurs, the system should not crash, and the user should receive helpful notification or guidance.

6.6 Load Testing

Load testing evaluates the system's ability to handle high traffic or large volumes of data. The system was tested with simulated users performing activities like logging in, sending emails, and requesting PDF downloads. This helped ensure the system could maintain performance during peak usage and handle multiple simultaneous users without significant delays.

6.7 Security Testing

Security is a major concern for systems handling sensitive user data, especially in telecommunications. The AI-based Data Breach Alert System underwent extensive security testing to ensure that it can protect against unauthorized access and data breaches.

Key security testing areas included:

1. **Authentication:** Ensuring that only authorized users could access the system by verifying credentials and IP addresses.
2. **Data Encryption:** Ensuring that data transmitted between the client and the server is securely encrypted using SSL/TLS protocols.
3. **Access Control:** Verifying that the system restricts access based on user roles, ensuring that admins and users can only perform actions within their authorized permissions.

6.8 Installation Testing

Installation testing ensures the system can be correctly installed on new machines and servers. It verifies that all necessary dependencies are included, and the system is installed smoothly without errors. The installation process was tested for both local and remote setups, ensuring the system could be deployed in different environments.

Installation errors, such as missing files or incorrect configurations, were also tested, and the system was designed to provide users with clear error messages and instructions on how to resolve issues.

6.9 Test Cases

Table 6.1: Test Cases

S. No	Test Case	Test Steps	Expected Result	Actual Result	Status
6.1	GUI Testing	Open admin dashboard and user portal; verify buttons, input fields, navigation menus, and layout on different devices	All GUI elements function correctly; interface is user-friendly and responsive	GUI elements working as expected	Pass
6.2	Usability Testing	Observe real users interacting with admin dashboard and user portal; test workflows like monitoring activity, sending emails/messages/ voice	Users can efficiently perform tasks; navigation is intuitive; system feedback is clear	Users performed tasks smoothly	Pass
6.3	Software Performance Testing	Simulate normal and peak usage; measure response time for login, messages, alerts; monitor throughput and system stability	System remains responsive; processes large volumes; maintains stability under load	System stable and responsive	Pass
6.4	Compatibility Testing	Test system on different browsers (Chrome, Firefox, Edge), devices (desktop, laptop, mobile), and OS (Windows, Mac)	System works correctly across browsers, devices, and OS	Compatible across tested platforms	Pass
6.5	Exception Handling	Input incorrect data, perform unsupported actions, and simulate system failures	System provides clear error messages, recovers from failures, and does not crash	Errors handled correctly; system recovered gracefully	Pass

Continued on next page

Table 6.1 – continued from previous page

S. No	Test Case	Test Steps	Expected Result	Actual Result	Status
6.6	Load Testing	Simulate high traffic with multiple users logging in, sending messages, and downloading PDFs	System handles multiple simultaneous users without significant delays	System maintained performance under load	Pass
6.7	Security Testing	Test authentication, role-based access, and data encryption	Only authorized users can access system; data is encrypted; access control enforced	Security measures working correctly	Pass
6.8	Installation Testing	Install system on new machines/servers, check dependencies, verify setup process	System installs correctly on local and remote setups; installation errors are properly handled	Installation completed successfully	Pass

Chapter 7

Conclusion

The AI-based Data Breach Alert System project offered valuable insights into designing, developing, and deploying comprehensive security solutions specifically for telecommunication companies. Throughout this project, we learned a great deal about the challenges of cybersecurity, integrating AI, and managing real-time monitoring systems. One of the standout achievements was incorporating Artificial Intelligence (AI) to detect spam and alert for potential breaches. By using machine learning models to analyze and classify user communications including emails, messages, and voice data as either "Spam" or "Not Spam," the system became much more effective at identifying suspicious behavior and preventing potential misuse of resources in real time.

Real-time monitoring proved to be critical for early detection of security threats. The system tracks login events, reviews message contents, and monitors PDF download activity, allowing administrators to quickly spot any unusual or suspicious activity. Security protocols like IP-based authentication, data encryption, and role-based access controls further ensure that sensitive information remains safe. Performance testing also showed that the system can handle large volumes of data and multiple users simultaneously without any noticeable slowdowns, maintaining both efficiency and security.

Another key focus was making the system easy to use. Both administrators and users can interact with it smoothly, even without technical expertise. Usability testing confirmed that intuitive workflows and clear navigation make daily tasks simpler and more efficient.

Looking ahead, there are several opportunities to improve the system. The AI model's accuracy in detecting spam could be enhanced with larger and more diverse datasets, reducing false positives and better understanding the nuances of everyday office communications. The system could also be optimized for greater scalability, allowing it to manage more users and larger datasets without affecting performance. Future versions may offer detailed user activity reports, capturing logins, messages, and file access to provide deeper insights and support better auditing and compliance.

User Manual

This User Manual provides guidance for using the AI-based Data Breach Alert System, covering installation, navigation, and troubleshooting.

7.1 Introduction

The AI-based Data Breach Alert System is designed to monitor user activities, classify communications as Spam or Not Spam, and provide real-time alerts to administrators. It helps secure sensitive data within telecommunication companies by analyzing emails, messages, and file access.

7.2 System Requirements

The system requires Windows 10 or Mac OS X 10.14+ as the operating system. It is compatible with browsers such as Google Chrome, Mozilla Firefox, or Microsoft Edge. The system uses MongoDB for data storage, and it requires at least 4GB of RAM (8GB recommended). A stable internet connection is necessary for real-time monitoring and updates.

7.3 Installation Instructions

To install the system, download the installer and run the setup. Follow the on-screen instructions to install MongoDB and the system. During the first launch, configure the admin credentials (username and password). After installation, start the system and log in using the admin credentials you configured.

7.4 Admin Dashboard

The Admin Dashboard allows administrators to monitor user activities. It provides information on login activity, including successful and failed logins along with IP addresses. The dashboard also allows administrators to monitor emails and messages sent by users, classifying them as Spam or Not Spam. Additionally, administrators can view and approve or reject PDF access requests, which are based on the user's IP address and provided reason. The dashboard sends real-time alerts for any suspicious activities detected.

7.5 User Portal

The User Portal is the interface for regular users. Users can log in using their registered credentials and from their authorized IP addresses. Through the portal, users can send emails and messages, which are automatically analyzed and classified by the system. Users can also send voice messages, which are transcribed and analyzed. If users need to download PDFs, they must provide a valid reason, and the request will be sent to the admin for approval.

7.6 Troubleshooting

If the system does not start, ensure MongoDB is running properly. For login issues, verify that the correct credentials are entered or reset them if necessary. If performance problems occur, check the network connectivity and ensure that the system is not overloaded. If AI classification errors happen, it may be necessary to retrain the model with updated data. In case a PDF access request is denied, verify that the user is logged in from a registered IP address.

References

[1] T. R. Weil and E. J. Coyne, "Abac and rbac: Scalable, flexible, and auditable access management," *IEEE IT Professional*, May 2013. Accessed from: https://csrc.nist.rip/groups/SNS/rbac/documents.

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, Dec. 2022.

[3] R. Oruç and E. Senyürek, "Spam detection with naïve bayes and tf-idf," *IOSR Journal of Computer Engineering*, vol. 27, no. 4, pp. 47–51, 2024.

[4] A. Dasgupta and S. Y. Mehr, "Enhanced mnb method for spam e-mail/sms text detection using tf-idf vectorizer," *American Journal of Mathematical and Computer Modelling*, vol. 9, pp. 1–9, Apr. 2024.

[5] H. Al-Khateeb and M. Al-Hamed, "Insider threat mitigation: Systematic literature review," *Journal of King Saud University – Computer and Information Sciences*, 2024.

[6] A. J. Hall, N. Pitropakis, W. J. Buchanan, and N. Moradpoor, "Predicting malicious insider threat scenarios using organizational data and a heterogeneous stack classifier," *arXiv preprint arXiv:1907.10272*, July 2019.

[7] CERT Division, Software Engineering Institute (SEI), Carnegie Mellon University, "Insider threat test dataset." SEI Library, 2015.

[8] MongoDB Inc., "Security checklist for self-managed deployments." MongoDB Manual Documentation, 2025.

[9] MongoDB Inc., "Role-based access control / authorization in mongodb." MongoDB Manual Documentation, 2025.