**Email Scanning System**

**Table of Contents:**

* Objective
* System Requirements
* System Architecture
* Features
* Technical Specifications
* Database Design
* User Interface
* Installation and Setup
* Future Enhancements
* Execution Screenshots
* Conclusion

**1. Objective**

Develop an Email Scanning System that efficiently scans, manages, and filters emails to detect and block spam or phishing attempts. The system will utilize advanced features and technologies to process emails, store and manage email data, and provide a user-friendly interface for administrators to monitor and configure scanning rules.

**2. System Requirements**

### **Hardware Requirements**

* Standard PC or laptop

### **Software Requirements**

* Java Development Kit (JDK) 8 or higher
* Eclipse IDE
* MySQL Database
* JDBC Driver for MySQL

**3. System Architecture**

### **Overview**

The Email Scanning System follows a desktop application architecture with Swing for the user interface and JDBC for database connectivity.

### **Components**

* Frontend: Java Swing
* Backend: JDBC for database interaction
* Database: MySQL

### **Data Flow**

* User interacts with the Swing frontend.
* Requests are processed by the Java backend.
* Backend communicates with MySQL database for CRUD operations.
* The frontend updates the User Interface based on the response from the backend.

## 

## **4. Features Employee Management**

* Add, view, update, delete employee records

**5. Technical Specifications**

### **Frontend (Java Swing)**

* Swing components for UI design
* Event-driven programming model

### **Backend (Java JDBC)**

* JDBC API for database connectivity

Prepared statements for SQL queries

### **Database (MySQL)**

* **Tables:**
  + email\_logs
  + email\_scans
  + logs
  + rules

**6. Database Design**

### **Tables**

* **email\_logs**
  + id (Primary Key)
  + log\_entry
  + log\_time
* **email\_scans**
  + id (Primary Key)
  + email\_content
  + suspicious\_percentage
  + scan\_timestamp
* **logs**
  + id (Primary Key)
  + log\_entry
  + timestamp
* **rules**
  + id (Primary Key)
  + rule

**7. User Interface**

### **Screens**

* Main Dashboard: Buttons for Scanning Email Content, Manage rules, View Logs, Export Logs, Generate Reports.
* Manage Rules Frame: Field to enter rules.
* View Logs: Field to view logs.
* Export: Field to export logs in desired format.
* Generate Reports: Chart displaying suspicious percentage by emails.

**8. Installation and Setup**

### **Steps**

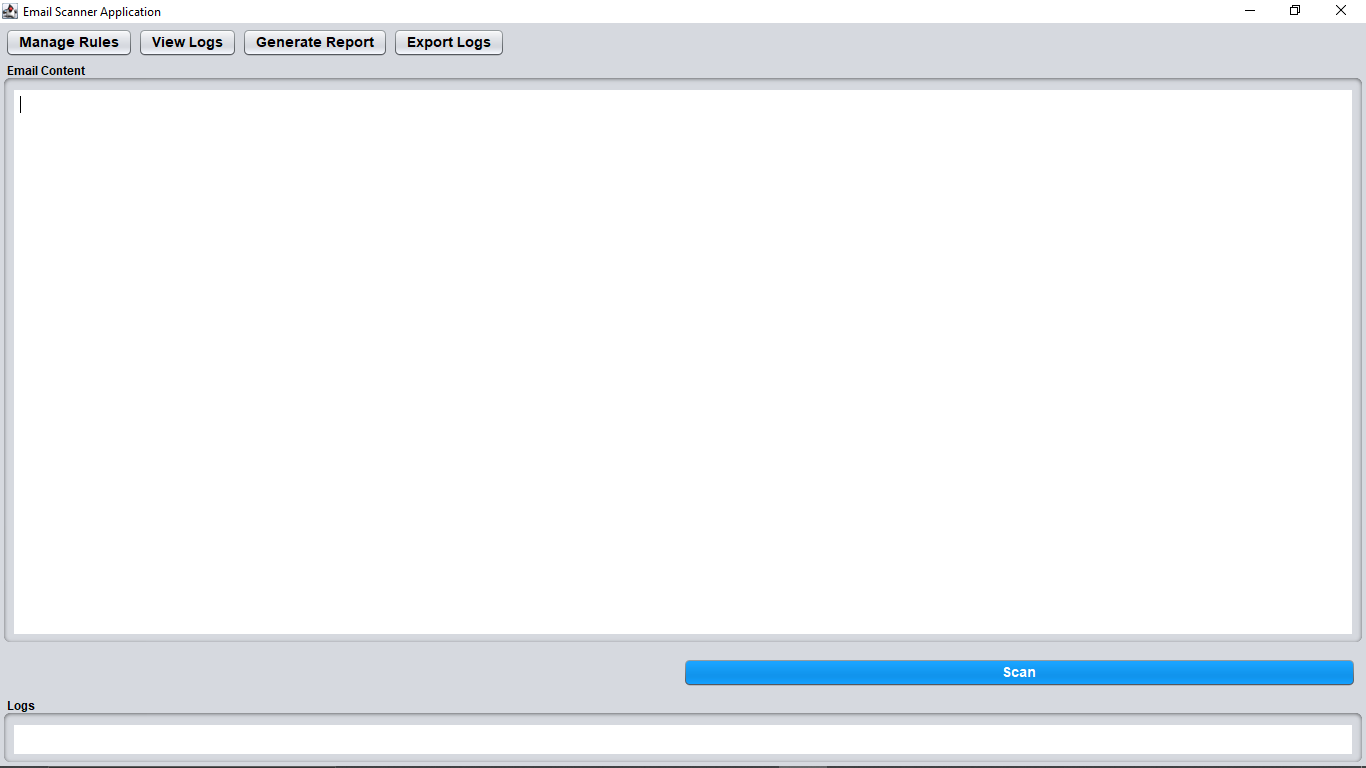
* Download and install Eclipse IDE.
* Clone the Java project repository.
* Import the project into Eclipse.
* Set up MySQL database and import the schema.
* Update database credentials in the Java code.
* Compile and run the EmailScannerApp.java file in Eclipse.

**9. Future Enhancements**

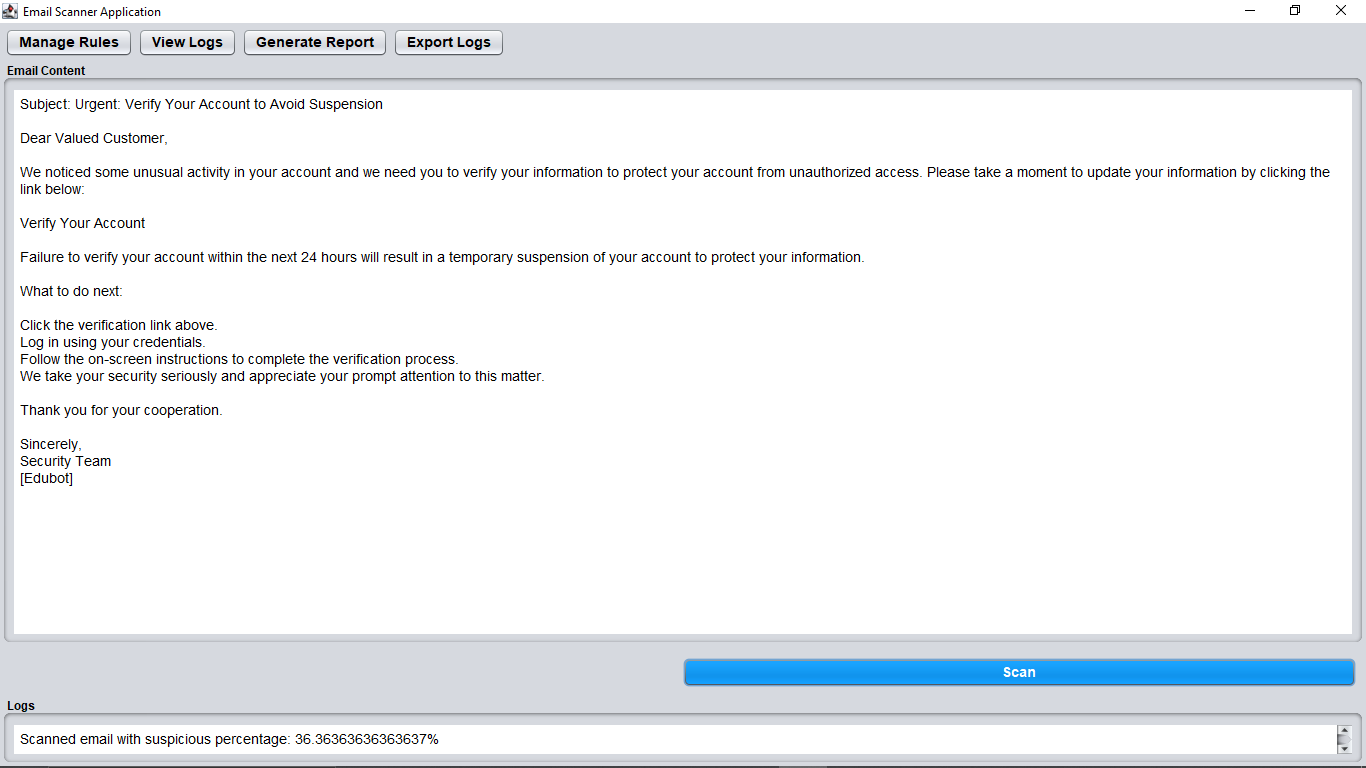
* Implement more sophisticated scanning algorithms to detect spam, phishing, and malware.
* Optimize database queries and indexing for faster data retrieval.
* Develop a more user-friendly web interface with real-time updates and interactive dashboards.
* Integrate with external services like email providers (e.g., Gmail, Outlook) for seamless scanning.
* Use AI to provide predictive analytics and threat detection.

**10. Execution Screenshots**

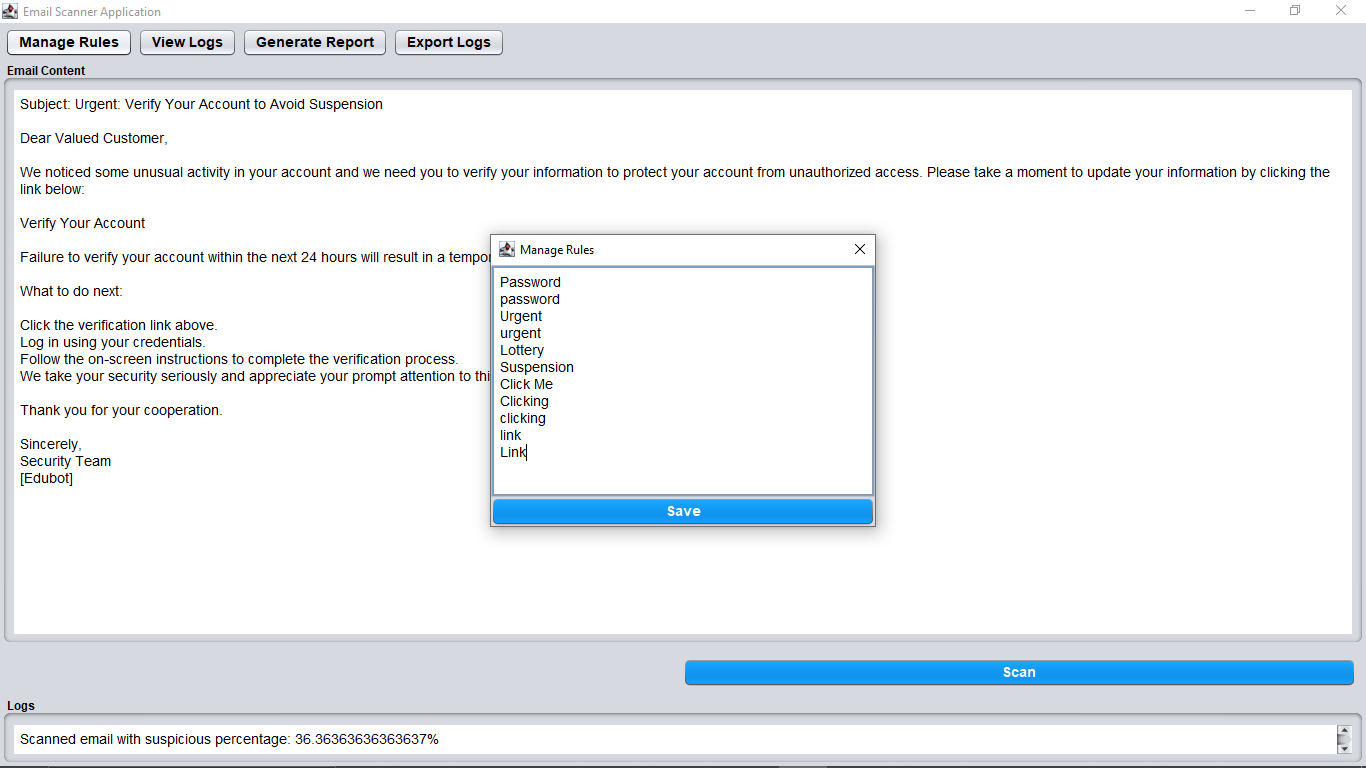
**Main Dashboard:**



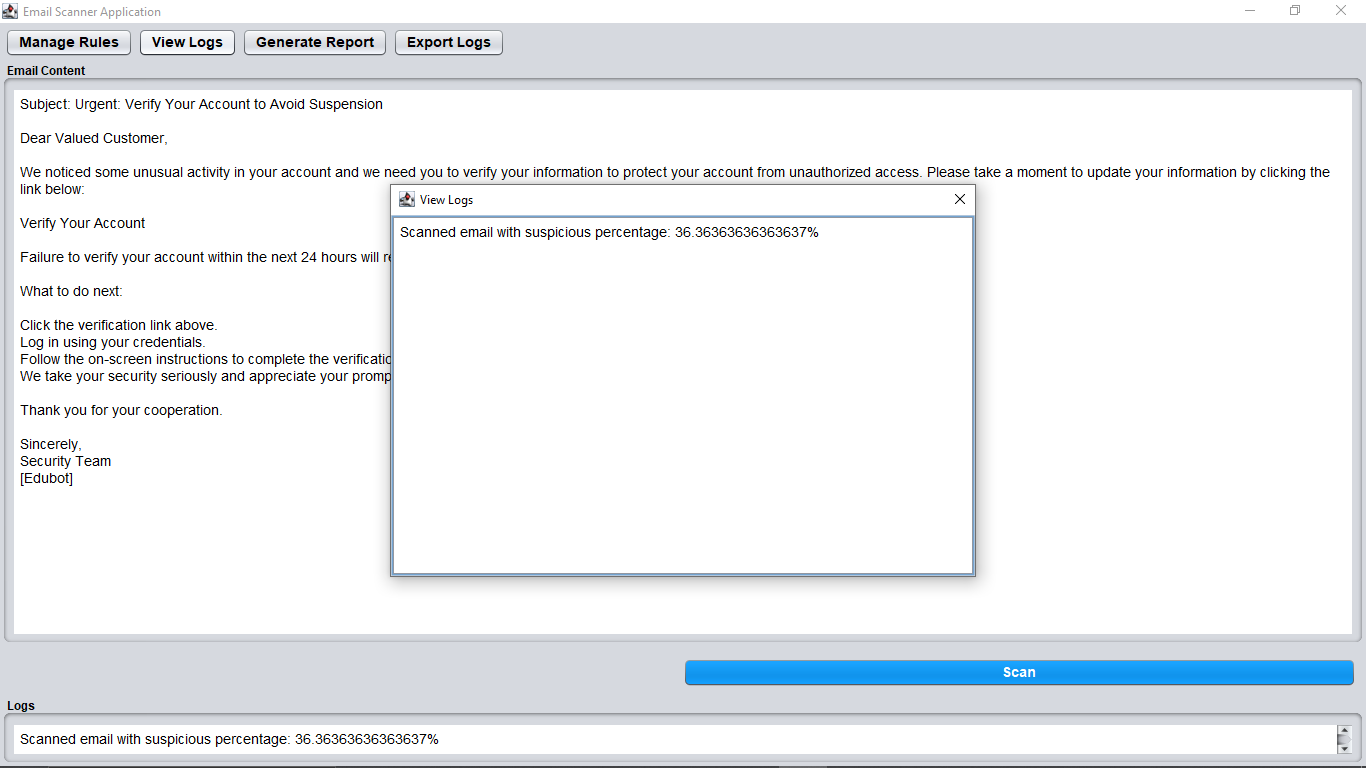
**Email Content:**



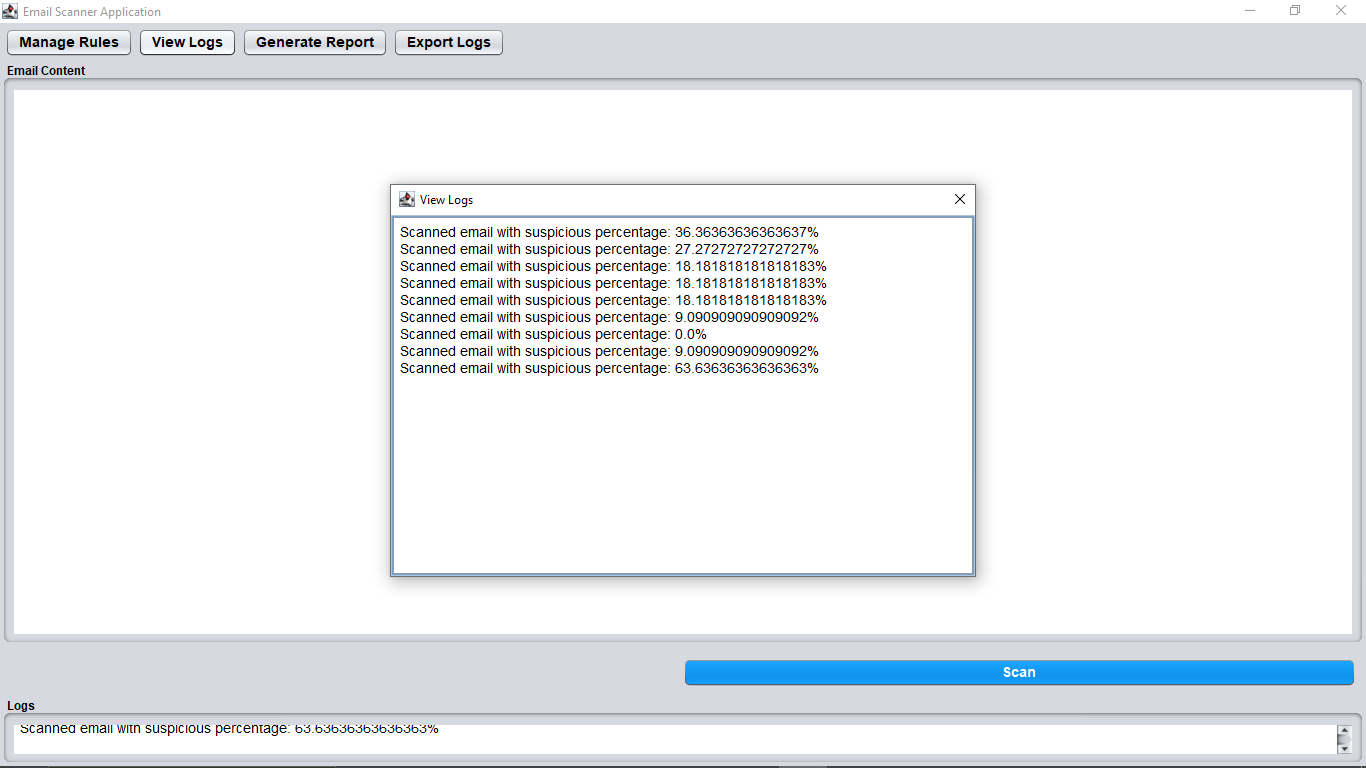
**Manage Rules:**



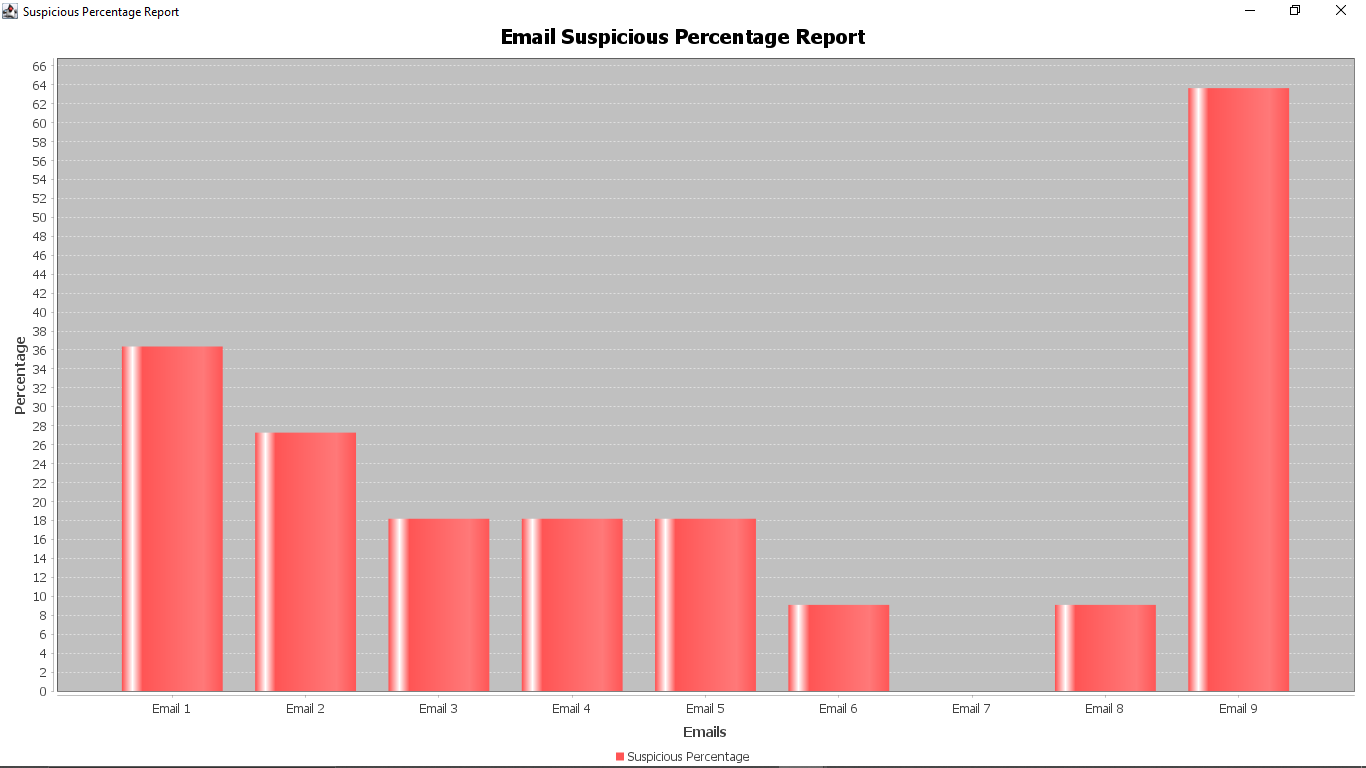
**Log of email content:**



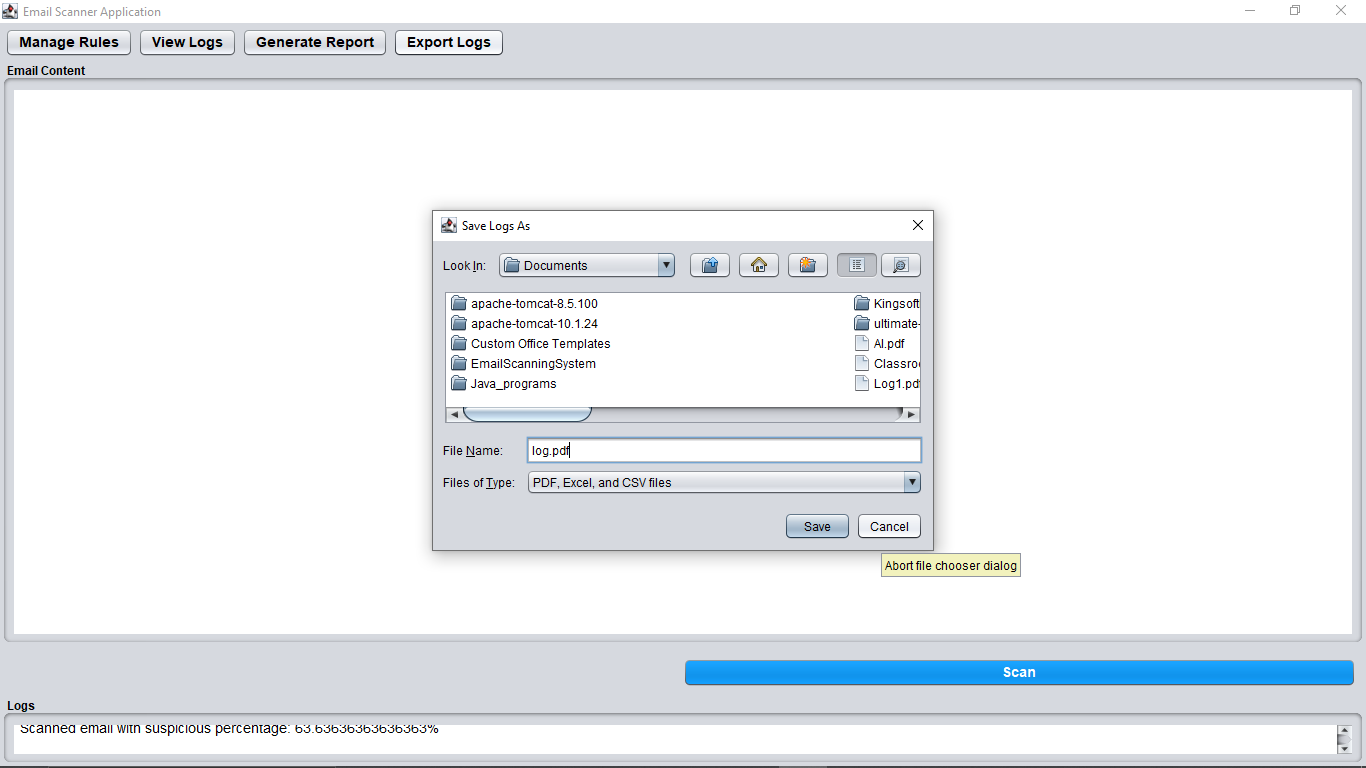
**View Logs:**

****

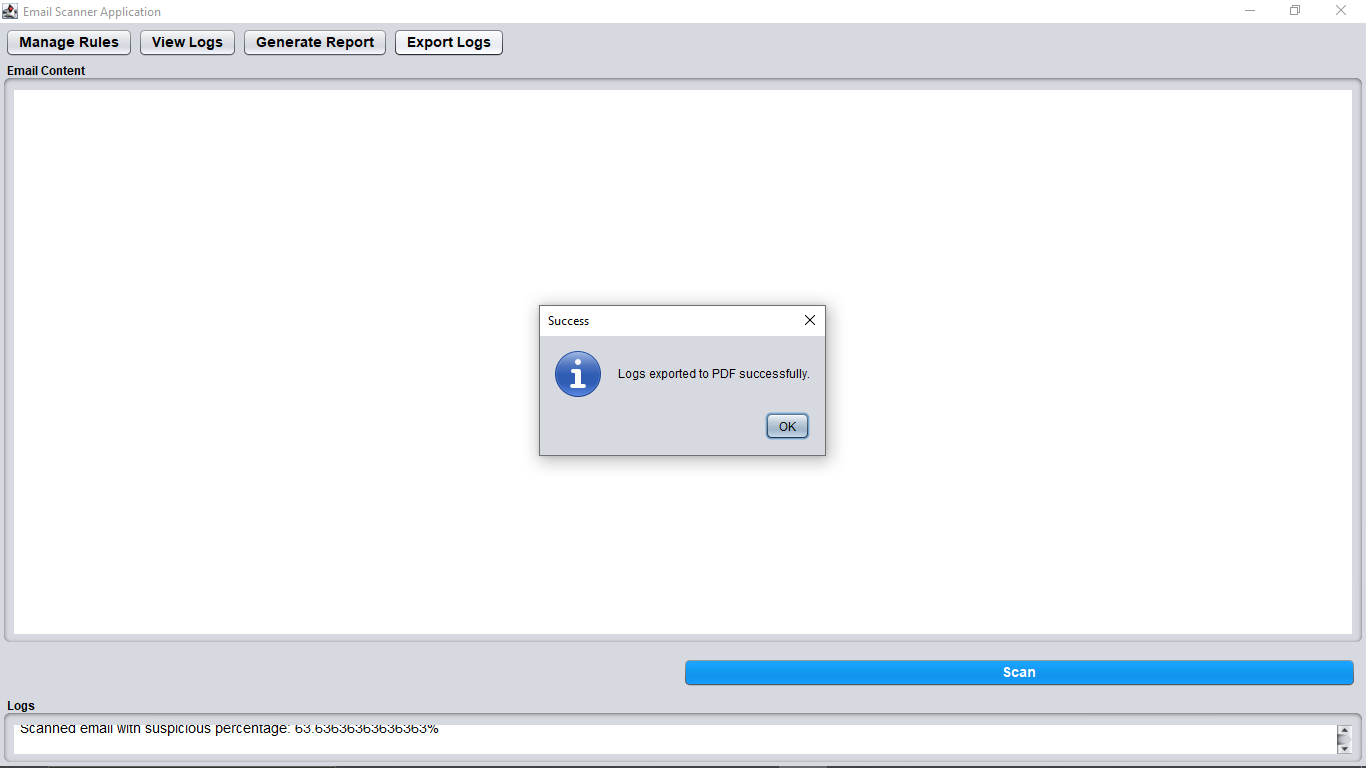
**Email Suspicious Percentage Report:**

****

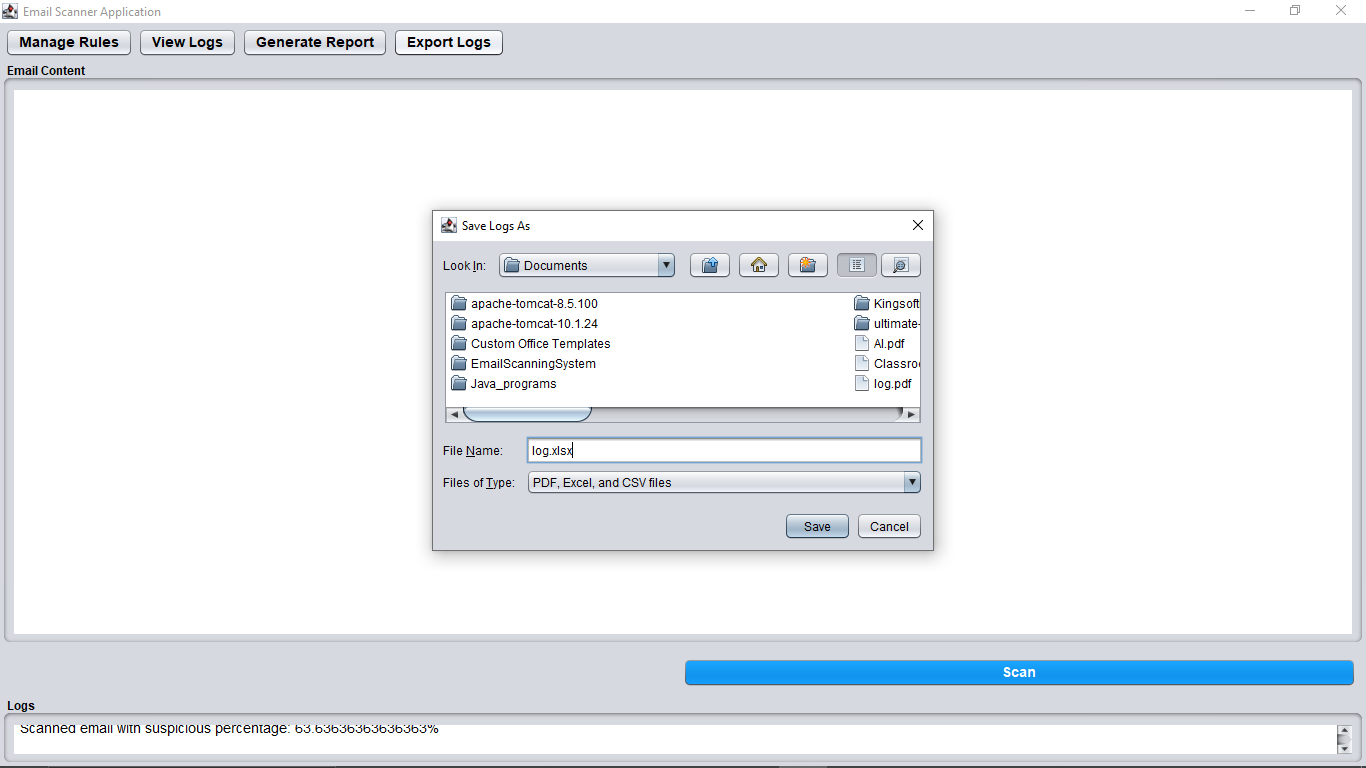
**Export as PDF:**

****

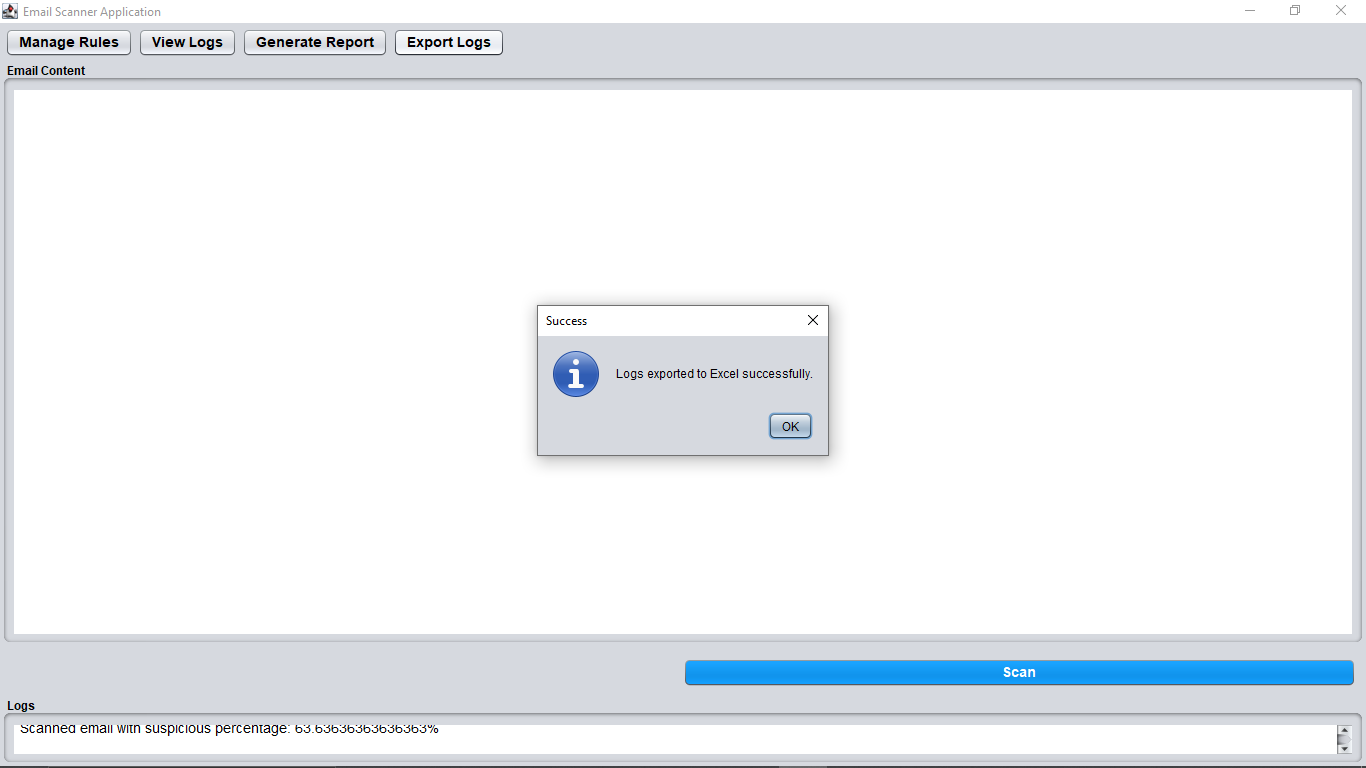
**Export as PDF Success:**

****

**Export as Excel:**



**Export as Excel Success:**



**11. Conclusion**

The Email Scanning System offers a comprehensive solution for managing and filtering email communications using Java, Swing GUI and JDBC. It ensures efficient email sorting, spam detection, and threat prevention. Future enhancements will focus on improving accuracy, expanding feature sets, and integrating with various email clients to provide a more secure and user-friendly experience.

.