PROJECT REPORT

CY4001 Secure Software Design

Shah Muhammad Qureshi 21K-3557
Abdul Moueed 21K-3594
Lachman Das 21K-3632

FAST NUCES, Karachi

Department of Computer Science

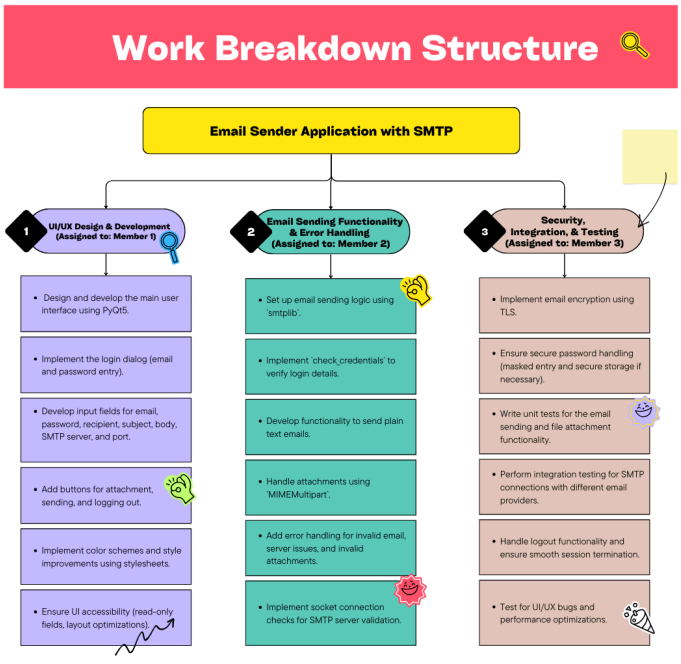
Project Report: Secure Email Sender Application with Advanced Security Features

1. Project Overview

Objective

This project's main goal is to produce an Email Sender Application that is not only safe but also reliable by using the Simple Mail Transfer Protocol (SMTP). This application is equipped with appliances that are preventing a myriad of cyber-attacks which secure everything from the safe transmission of emails to the security of the user's data. The solution has a secure authentication feature, encrypted communication support, attachment and a user-friendly interface as key components.

2. Work Breakdown Structure



3. Tools and Technologies

- **Programming Language**: Python
- **Libraries**:
  - `smtplib`: Secure email sending
  - `email.mime`: Structured email composition
  - `PyQt5`: Graphical User Interface (GUI) development
  - `ssl`: Encrypted communication
  - `unittest`: Comprehensive testing framework
- **Development Environment**: Python 3.9+
- **Version Control**: GitHub (optional)

4. Security Features

4.1 Encryption and Authentication

- **TLS Encryption**: Ensures end-to-end secure communication between client and server.
- **Secure Authentication**: Protects login credentials with masked inputs and secure handling.

4.2 Attachment Restrictions

- Filters attachments to prevent execution of malicious scripts or code.
- Enforces size limits to mitigate the risk of Denial-of-Service (DoS) attacks.

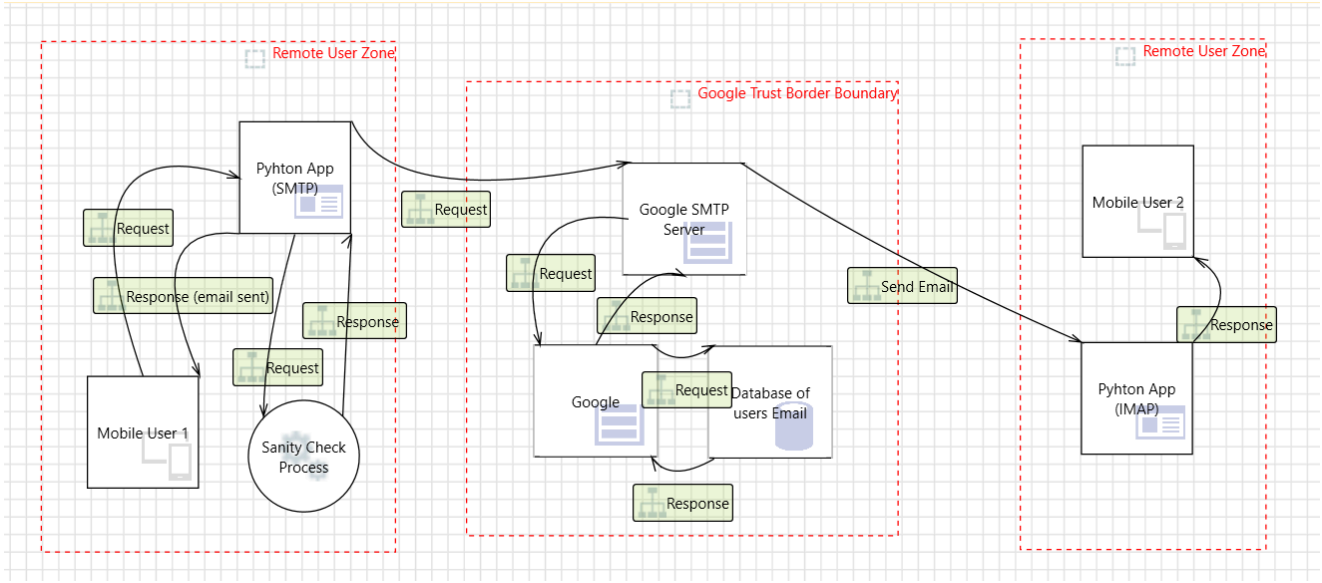
4.3 Session and Credential Management

- Implements session timeout mechanisms to protect against unauthorized access.
- Encrypts stored credentials using industry-standard hashing algorithms.

4.4 Error and Threat Mitigation

- Provides descriptive error messages to guide users without revealing sensitive application logic.
- Includes automated checks for valid SMTP configurations and recipient email addresses.

5. Threat Modeling



Threat List

ID	Diagram	Changed By	Last Modified	State	Title	STRIDE Categ	Description	Justification	Interaction	Possible Mitig	Severity	SDL Phase
15	**Diagram 1**		**Generated**	**Not Started**	**An adversary m**	**Elevation of Priv**	**An adversary m**		**Request**	**Implement impl**	**High**	**Design**
16	Diagram 1	LAPTOP-EL7RN2	12/13/2024 8:01:	Not Started	An adversary car	Information Disc	An adversary car		Request	Implement Certifi	High	Implementation
17	**Diagram 1**		**Generated**	**Not Started**	**An adversary ca**	**Information Dis**	**If application sa**		**Request**	**Encrypt sensitiv**	**High**	**Implementatior**
18	Diagram 1		Generated	Not Started	An adversary car	Tampering	An adversary car		Request	Obfuscate gener	High	Design
19	**Diagram 1**		**Generated**	**Not Started**	**An adversary ca**	**Elevation of Priv**	**If there is no re:**		**Request**	**Configure a Wir**	**High**	**Implementatior**
20	Diagram 1		Generated	Not Started	An adversary car	Elevation of Privi	Database access		Request	Ensure that least	High	Implementation
21	**Diagram 1**		**Generated**	**Not Started**	**An adversary ca**	**Information Dis**	**Additional cont**		**Request**	**Use strong encr**	**High**	**Implementatior**
22	Diagram 1		Generated	Not Started	An adversary car	Information Disc	SQL injection is a		Request	Ensure that login	High	Implementation
23	**Diagram 1**		**Generated**	**Not Started**	**An adversary ca**	**Repudiation**	**Proper logging**		**Request**	**Ensure that logi**	**Medium**	**Implementatior**
24	Diagram 1		Generated	Not Started	An adversary car	Tampering	An adversary car		Request	Add digital signa	High	Design
25	Diagram 1		Generated	Not Started	An adversary ma	Tampering	An adversary ma		Request	Enable Threat de	High	Design

Export Csv 11 Threats Displayed, 11 Total

6. Code Functionality

6.1 Authentication

- Validates user credentials with secure SMTP connections.
- Encrypts password inputs and prevents plaintext storage or transmission.

6.2 Secure Email Transmission

- Composes emails with or without attachments while ensuring compliance with security policies.
- Implements secure MIME handling to mitigate risks associated with attachment misuse.

6.3 Error Handling

- Detects and mitigates potential issues such as invalid addresses, large attachments, and connectivity failures.
- Logs errors securely for debugging without compromising user privacy.

6.4 User Interface

- Provides an accessible and visually secure interface designed with PyQt5.
- Integrates real-time feedback for user actions, enhancing usability.

7. Testing and Results

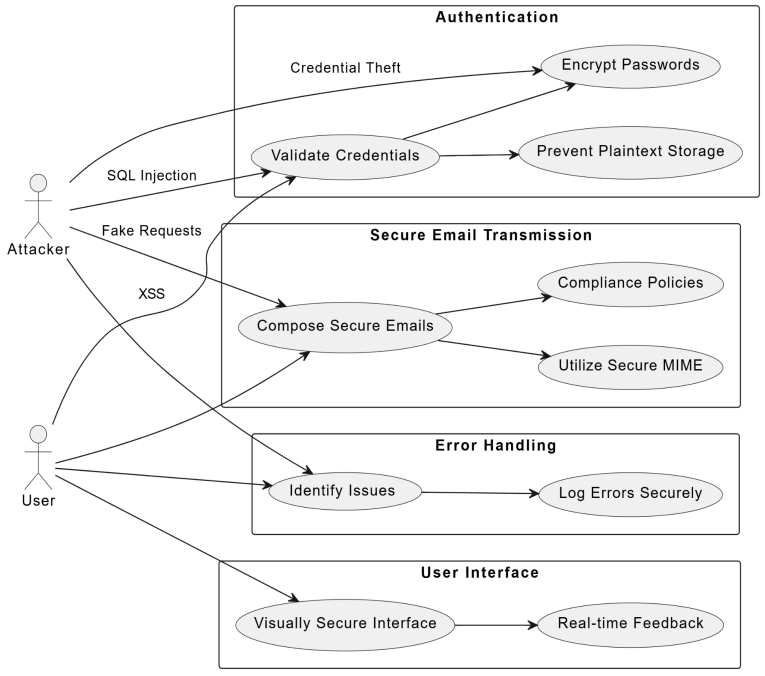
7.1 Testing Framework

- **Unit Testing**: Verified the integrity of individual modules, including authentication and email composition.
- **Integration Testing**: Evaluated the seamless interaction between UI, backend logic, and SMTP servers.
- **Penetration Testing**: Simulated attack scenarios to ensure resistance against brute-force attempts, malformed inputs, and email spoofing.
- **Compatibility Testing**: Ensured reliable operation across Gmail, Yahoo, and Outlook SMTP configurations.

7.2 Results

- Successfully authenticated users with encrypted credentials.
- Transmitted emails securely, including attachments, without breeches.

8. Misuse Case Diagram



9. Challenges and Solutions

9.1 Security Challenges

- **Credential Protection**:
    - **Issue**: Risks associated with plaintext password storage.
    - **Solution**: Adopted encryption protocols for secure handling and storage.
- **Malicious Attachments**:
    - **Issue**: Potential exploitation through harmful file uploads.
    - **Solution**: Implemented filters and validation to restrict executable files.
- **Server Compatibility**:
    - **Issue**: Variations in SMTP configurations among providers.
    - **Solution**: Integrated dynamic server and port input fields to enhance flexibility.

9.2 Lessons Learned

- Security-first design principles significantly enhance application reliability.
- Rigorous testing uncovers vulnerabilities that could otherwise compromise the system.

10. Conclusion

The Secure Email Sender Application portrays a solid incorporation of advanced security traits into a working and customer-friendly email client. Encryption, error handling, and rigid attachment policies are the three main features of this application that are focused on, hence it provides reliable and secure communication. The project forges a pathway for email technology to improve as it focuses more on the security of the application's design.