SECURE MESSAGE APPLICATION

HIGHER DIPLOMA IN SOFTWARE ENGINEERING

SOFTWARE SECURITY

PROJECT PROPOSAL

Submitted By:

GAHDSE22.1F-022 M.B.D. SEWWANDI

GAHDSE22.1F-018 R.L DINAL RANDIKA

TABLE OF CONTENT

INTRODUCTION

The primary goal of this project is to establish a secure and confidential messaging platform. This project aims to develop a secure messaging system using Digital Signature Algorithm (DSA) and the SHA-1 hashing algorithm.

Digital signatures and key exchange are the two main applications for the widely used asymmetric cryptographic method known as DSA. Federal Information Processing Standard (FIPS PUB 186) was created by the National Institute of Standards and Technology (NIST) in the United States and released in 1994. DSA is closely related to the Diffie-Hellman key exchange technique and is based on the mathematical concepts of modular exponentiation and discrete logarithms.

SHA-1 or Secure Hash Algorithm 1 is a cryptographic algorithm which takes an input and produces a 160-bit (20-byte) hash value. This hash value is known as a message digest. This message digest is usually then rendered as a hexadecimal number which is 40 digits long. It is a U.S. Federal Information Processing Standard and was designed by the United States National Security Agency. SHA-1 is now considered insecure since 2005.

This system incorporates role-based access control to manage a wide range of communication situations. It empowers users to securely exchange messages, offering both sending and receiving capabilities. The project entails creating a user-friendly Python application with an intuitive interface designed for the secure transmission and reception of messages.

OBJECTIVES

1. Secure Messaging

The Secure Messaging System seeks to give users access to a private and secure communication environment. Digital Signature Algorithm (DSA) is used to encrypt messages, SHA-1 hashing algorithm is used to store passwords, and input validation is used during user registration. The system protects user data while ensuring confidentiality, authentication, and authorization.

2. Authentication and Authorization

Users are required to log in with their own username and password, which are determined by their user role, to access the system. Only authenticated users in this system will be able to see decrypted communications. Additionally, the system will impose role-based access control, enabling message senders to define which users have access to their communications. In this case mainly Develop a user authentication system requiring valid username and password input, implement role-based access control to authorize users based on their roles, ensure that only authorized users can view decrypted messages.

3. User Registration

A robust user registration process is essential for onboarding users securely. The system will incorporate input validation to guarantee that usernames adhere to the Correct format and that passwords meet minimum security requirements. In this case SHA-1 Hashing algorithm is used to provide protection to password before insert to the database.

4. Password Hashing

Passwords are a critical element of user security. The project will include password hashing using the SHA-1 algorithm. This hashing algorithm is used in user registration process and user sign in process to provide security to user password.

5. Input Validation

Perform input validation during user registration to ensure usernames are in correct format and enforce strong password requirements. Uther password validate using minimum character size, Uppercase, Lowercase, numbers, symbols.

6. Use Secure Database

Establish a secure database to store user details, including usernames, roles, and encrypted passwords**.**

7. Exception Handling

This system handles the errors occur in system by using the exceptions handling process.

IMPLEMENTATION

1. **FRONTEND**:

- Develop a user-friendly application using a Python based framework.
- Implement a responsive and intuitive user interface for message exchange and user interaction.
- Create user interfaces for user registration, login, message management with encryption and key management.
- Ensure the frontend communicates securely with the backend for data exchange.

2. **BACKEND**:

- Build a backend server using a Python web framework Flask for this application.
- Implement DSA-based message encryption and decryption logic to ensure secure communication.
- Develop user authentication mechanisms using SHA-1 Hashing Algorithm.
- Design process for user registration, user login message exchange, key management, and role assignment.
- Handle database operations securely and efficiently (store and retrieve).

3. **DATABASE**:

- Utilize a secure and reliable database system such as MySQL to store user details and messages.
- Design a database schema that includes tables for user accounts, user roles, public keys.
- Message should be stored in a text file.
- Implement indexing and normalization to ensure data integrity and optimize query performance.

ASYMMETRIC ALGORITHM

✞ DSA (Digital Signature Algorithm)

DSA (Digital Signature Algorithm) is a widely-used asymmetric cryptographic algorithm that is primarily used for digital signatures and key exchange. It was developed by the National Institute of Standards and Technology (NIST) in the United States and published as a Federal Information Processing Standard (FIPS PUB 186) in 1994. DSA is based on the mathematical principles of modular exponentiation and discrete logarithms and is closely related to the Diffie-Hellman key exchange algorithm.

ADVANTAGES

- Security: When used appropriately and with the right key sizes, DSA offers robust protection against a variety of threats, such as forging and impersonation. It is based on the computationally expensive and mathematically challenging discrete logarithm problems.

- Efficiency: When it comes to creating and verifying signatures, DSA is typically more effective than some other digital signature algorithms, such as RSA. This qualifies it for situations with limited resources, such as smart cards and embedded systems.

- Deterministic Signatures: For a given message and private key, DSA generates deterministic signatures. In some applications where predictability is sought, this means that the same message and private key will always result in the same signature, which can be significant.

HASHING ALGORITHM

✞ SHA-1 or Secure Hash Algorithm

SHA-1 or Secure Hash Algorithm 1 is a cryptographic algorithm which takes an input and produces a 160-bit (20-byte) hash value. This hash value is known as a message digest. This message digest is usually then rendered as a hexadecimal number which is 40 digits long. It is a U.S. Federal Information Processing Standard and was designed by the United States National Security Agency. SHA-1 is now considered insecure since 2005.

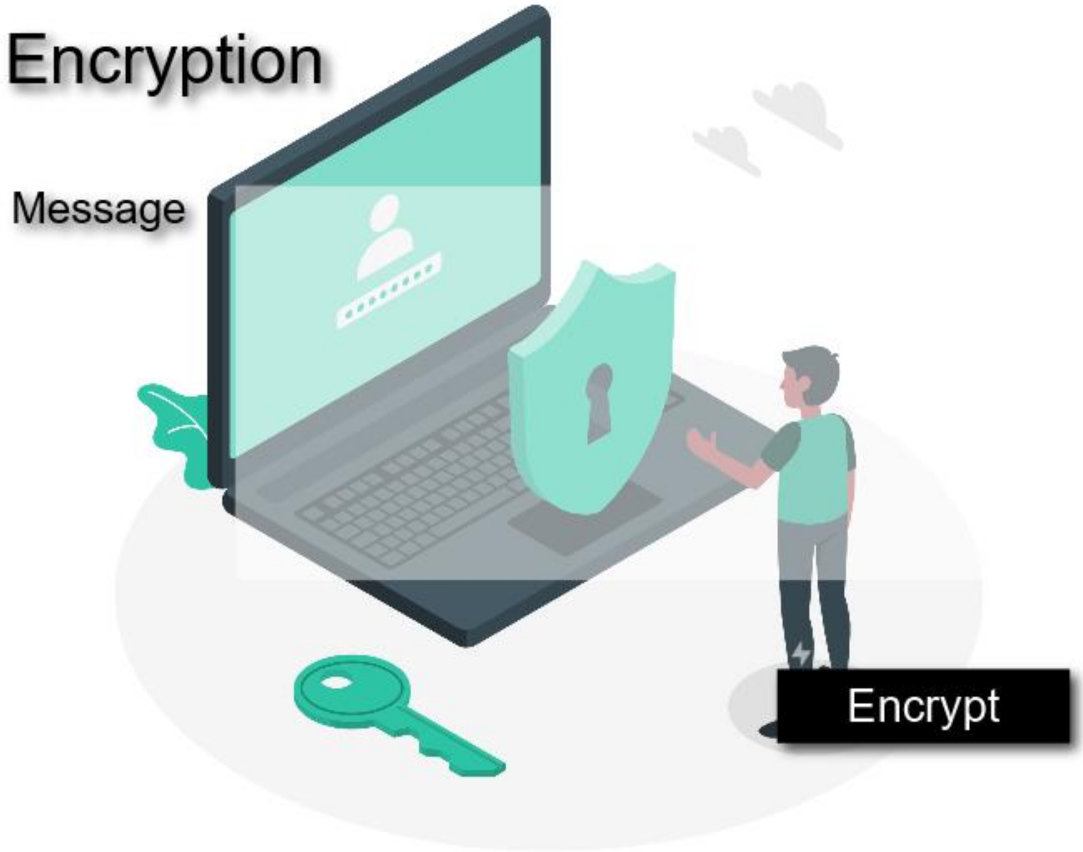
ADVANTAGES:

- Speed: In terms of computation, SHA-1 is fairly quick. It is suitable for applications where performance is important because it can hash data quickly.
- Widespread Support: At one time, SHA-1 was widely supported by a number of platforms and programming languages, making it simpler to implement in a variety of systems and applications.

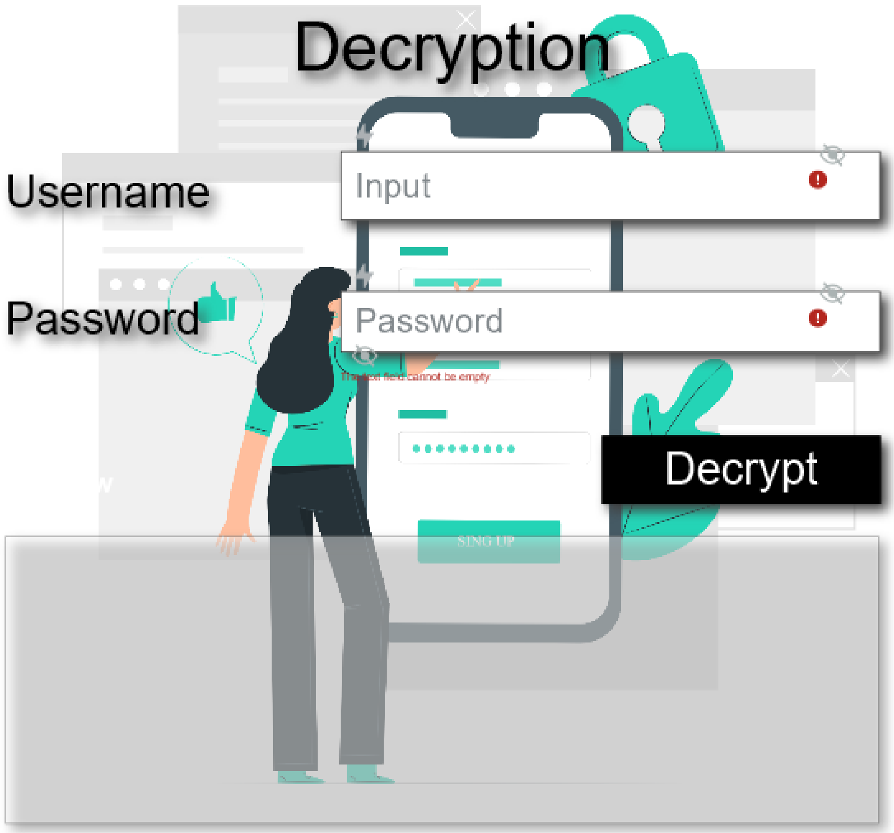
- Deterministic: SHA-1 will always generate the same hash value from the same input. In some circumstances, such as checksum checking, this determinism may be advantageous.

INTERFACE DESIGN

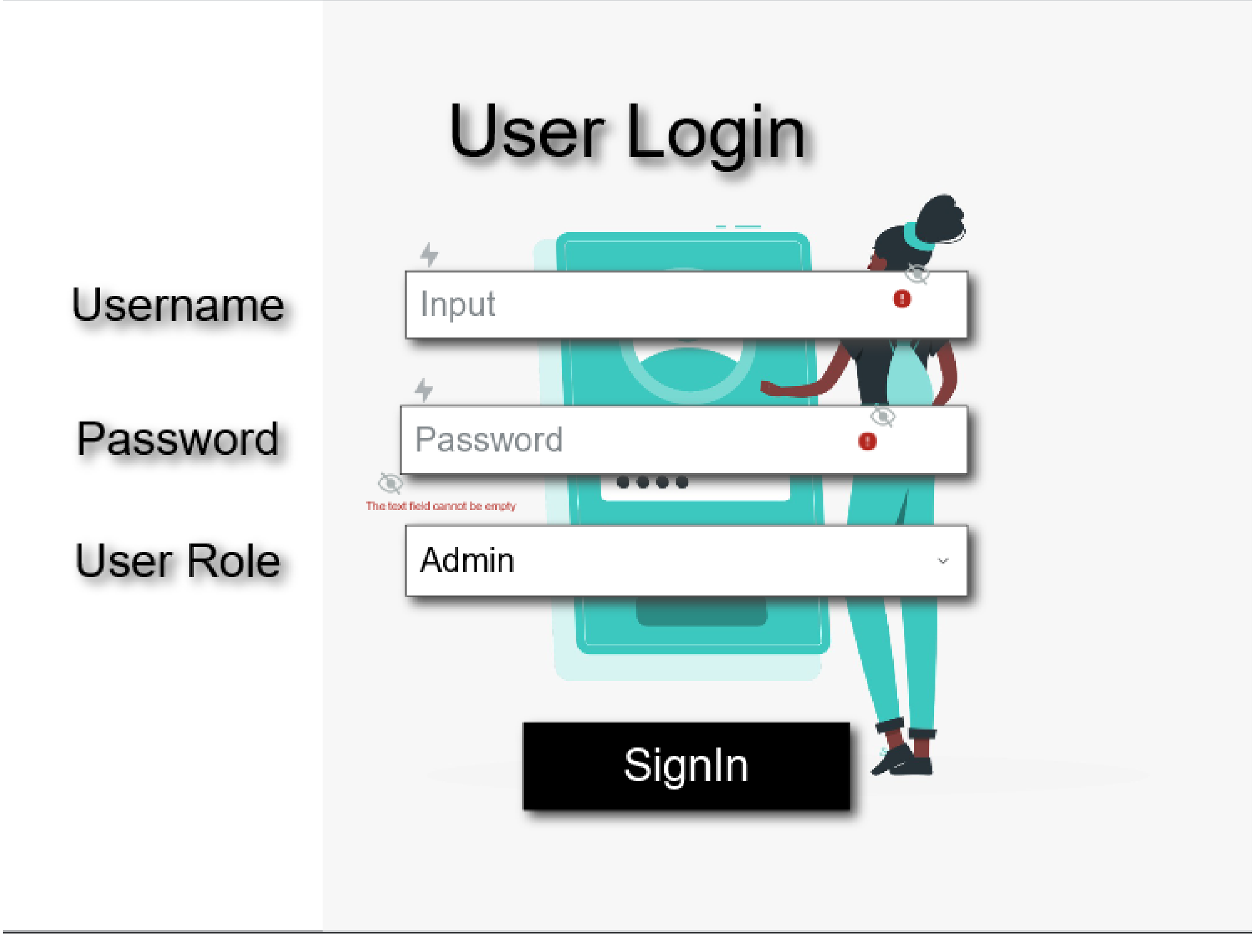
1. SEND MESSAGE PAGE



2. VIEW MESSAGE PAGE



3. USER REGISTRATION PAGE



User Login

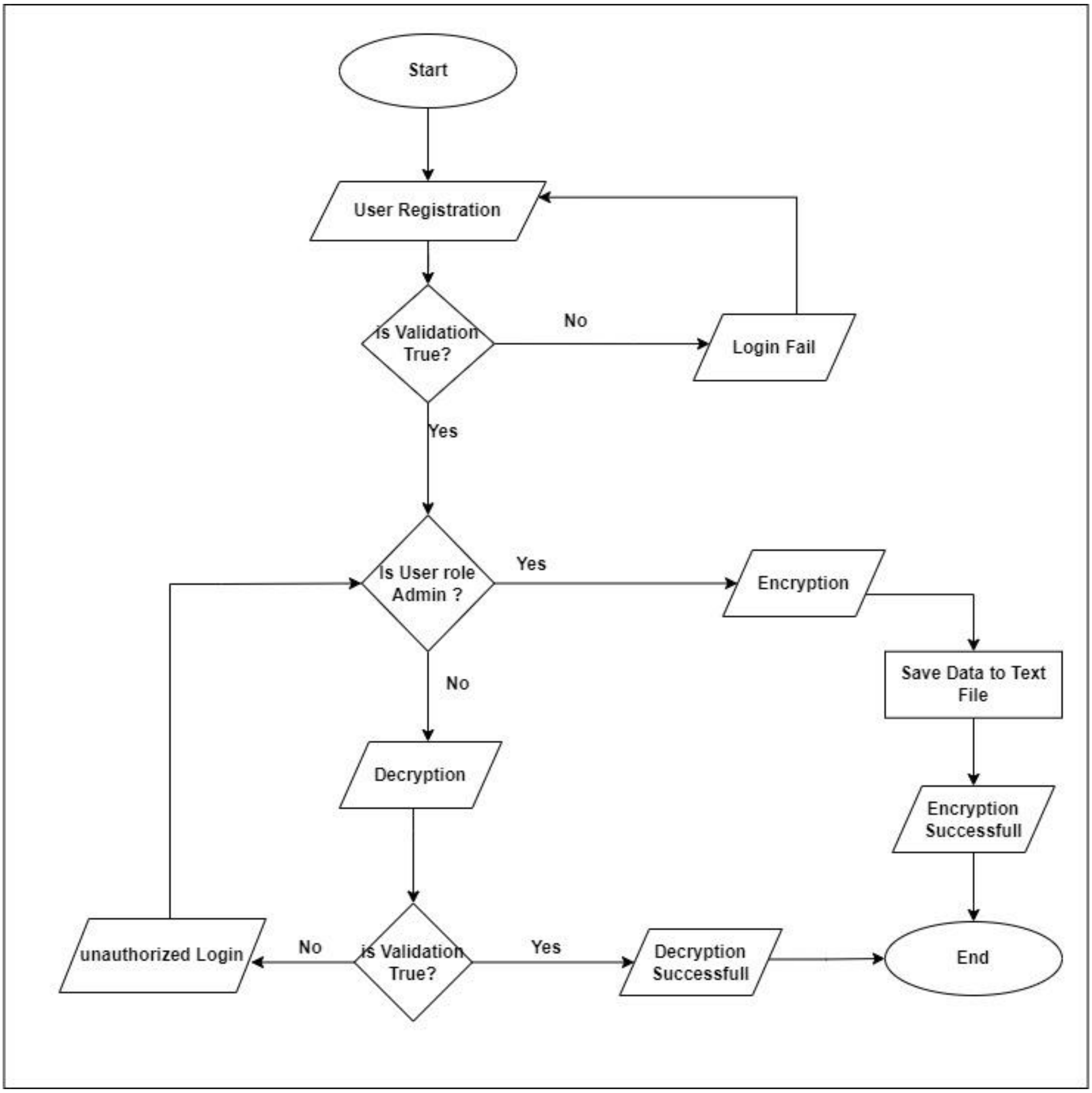
Username: Input

Password: Password

The text field cannot be empty

User Role: Admin

FLOW CHART DIAGRAM



***