PASSWORD MANAGER

4th Year Project

JASON DARLEY
C00246523

Contents

Introduction

Overview

The application to be developed is a java executable. It will allow users to generate passwords and store them within a database after they are encrypted with aes using cbc mode and decrypted when retrieving them while also using aes with cbc mode to decrypt the data.

When the user creates an account a salt and pepper will be generated so the user's password will be securely hashed and when the user tries to login the salt and hash will be retrieved and added to the new password input and then hash the input and compare it to what was stored and if they match the user will be granted access otherwise a message will display saying "incorrect password". A random key will also be generated for each user so they can encrypt and decrypt their data securely, the salt, pepper and key will all be stored in separate database tables so when the user tries to retrieve them it will match the username and retrieve the corresponding salt, pepper, and key for the user, this information will be stored in separate tables so if one table gets breached say the table with the salt then the hash will stay secure as the hacker wont have access to the pepper and therefore cant try to find the original text of the hash.

General Description

System Functions

MySQL/ SqLite functions

- Store the user account information like usernames and hashed passwords
- Store a generated key for encryption and decryption for each user
- Store a generated salt and pepper for each user
- Allow users to retrieve their data

- Allow users to securely login using sha 256 to compare the stored and newly inputted hashed password
- Allow users to change passwords in case they forget their password and need to regain access
- Allow users to create a new account

Java Application Functions

- Will contain the code that will provide all the gui for each section
- Will contain the code to generate strong passwords and will also provide a strength tester as part of the generator
- Contains the encryption and decryption code for aes using cbc mode
- Will prevent the attempts of brute force attacks on logins by limiting the number of failed login attempts and when that limit is reached either a timeout will be given or the app will close
- Provides the user a way to securely login and logout
- Provides the user an account creation feature
- Function to provide users with training on certain topics and allows the user to take quizzes on those topics to test their knowledge

User Characteristics and Objectives

Target Market

This application will be targeted towards tech companies as the password manager would be useful for the employees to keep track of all the passwords used for different sites and applications used within the company and the password manager also provides free cyber security training to the users, so companies won't have to pay for training separately.

Objectives

Desirables of the application

- Automation of the setup of SQLite/MySQL to make the use of the application easier for the user
- Password generator compatible with majority of sites by including certain criteria such as upper and lowercase characters, numbers, and special characters along with the password length does not exceed certain sites since some only allow a max of 12 characters
- Provide a user-friendly interface that is very easy to navigate
- Easy setup of the application after its downloaded
- Secured with modern encryption and hashing methods like aes and sha 256
- Secured against common vulnerabilities

Operational Scenarios

Initial Setup

- The user downloads the application software
- The user installs the program on their device
- The user creates an account
- The user has an auto generated key, salt, and pepper created for them in order to use the application fully
- The user logs into the application using the newly made credentials

General Usage

New Password

- The user logs into their account
- The user clicks the password generator button on the menu bar
- The user clicks on the create button to generate a new password
- The user copies the newly generated password displayed to them

- The user clicks the store new password button
- The user pastes the password and enters the site it is used for then clicks the save button and is prompted for their password before saving

View passwords

- The user logs into their account
- The user clicks the view passwords button from the menu bar
- The user is prompted for their password to decrypt their passwords
- The passwords are decrypted and shown to the user if the password is entered correctly

Training

- The user logs into their account
- The user clicks the Training dropdown button from the menu bar revealing the different training topics
- The user clicks on whichever topic they want to learn about
- The user is brought to the training for the topic they clicked on and presented with the information provided for that topic

Quizzes

- The user logs into their account
- The user clicks the quizzes dropdown button from the menu bar revealing the different quiz topics
- The user clicks on whichever topic they want to learn about
- The user is brought to the quiz for the topic they clicked on and presented with questions based on what the user learns from the training of the same topic

Functional Requirements

Functional

#	Function	Description	Criticality	Tech Issues	Dependencies
1	Create Account	Allows a new user to create an account	High	None	MySQL/SQLite
2	Login	Allows a user to login to their account	High	None	Function #1 MySQL/SQLite
3	Generate Key for each user	When account is made a key will be made for each user	High	None	MySQL/SQLite
4	Generate a salt and pepper for each user	When account is made a salt and pepper will be made for each user	High	None	MySQL/SQLite
5	Encrypt Data	User's data will be encrypted using aes and stored in the database	High	None	Bouncy Castle MySQL/SQLite

6	Decrypt Data	User's data will be decrypted using aes and displayed to the user	High	None	MySQL/SQLite Bouncy Castle
7	Store Data	The users data will be sent and retrieved to/from the database	High	None	MySQL/SQLite
8	Generate Strong Passwords	The User can use this feature to generate strong passwords	High	None	Randomness Requires upper, lowercase characters, number, special characters.
9	Logout	Securely logs out the user	High	None	User is logged in

10	Reset Password	The user could reset their password if they forgot it so they can regain access to their data and account	High	None	MySQL/SQLite
11	Training				
12	quizzes				

Usability

- The system must be easy to use
- The system must work for windows
- The generated passwords must be compatible with majority of sites policies

Reliability

- The system must be secure
- The information on the training must be up to date

Performance

- The system must be able to encrypt and decrypt quickly
- The system must be able to send and retrieve the data from the database quickly

Supportability

- A navigation guide must be available for the users so they can become familiar with the application and where everything is
- Installation of the application must be fast and easy

Difference from Existing Products

Most other password managers will only use a hash with no salting or peppering to make the hash more secure which can be a big vulnerability to the product while uses a unique salt and pepper for each user when they create an account using my program.

My password manager will also include cyber awareness training in order to educate users on cyber attacks. I chose to include training as people are the weakest part of any security system so by including training users can ensure their passwords will stay safe as they'll know not to send personal information/credentials to people, and they will learn methods on how to keep their password secure.

System Architecture

Requirements

MySql/SqLite Database

The database will consist of multiple tables in order to store information like the passwords to be secured, the users account credentials, and the users personal salt and pepper that gets generated when they make their account.

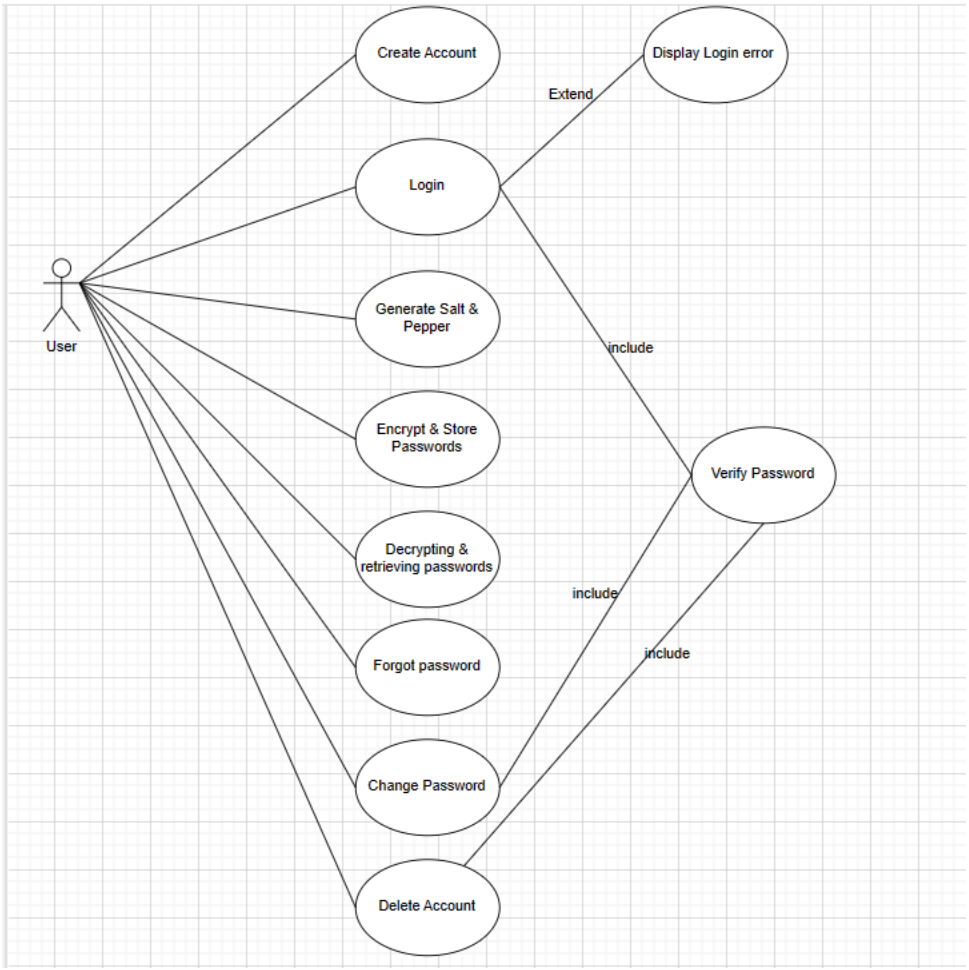
Java

Java is required as it is the language, I will be using to code the project, I chose to use java as it has less vulnerabilities than web page languages and I have more experience with java than languages like python.

Bouncy Castle

I will be using bouncy Castle in order to perform the necessary cryptographic algorithms as bouncy castle is a java library used to perform cryptographic functions and gives access to more algorithms for java.

High-Level-Design-Use-Case-Diagram



Create Account

Primary Actor

User

Preconditions

None

Success Guarantee

The user can create an account and their username and password is saved to the database and is used to authenticate later. A salt and pepper will also be created for when a user makes an account and stored in the database.

Main Success Scenario

1. The user opens the application
2. The user clicks the create account button
3. The user enters a username and password they want to use for their account
4. If the username is available, the users account is created, and a salt and pepper get generated, and all this information gets stored in the database

Alternative Flow

The username is not available, and the user is prompted to try a different username.

Login

Primary Actor

User

Preconditions

The user must have made an account on the application prior to attempting a login

Success Guarantee

The user authenticates with the application and the user gains access

Main Success Scenario

1. The user opens the application
2. The user enters their username and password

3. The salt and pepper stored for that user if the user exists is retrieved and added to the newly entered password then hashed and compared to the stored password.
4. If the passwords match the user gains access to the account and then application
5. The user is brought to the home screen and has full access to the application and its features

Alternative Flow

The user enters the wrong username or password and is prompted to try again

Generate Salt and Pepper

Primary Actor

User

Preconditions

The user creates an account

Success Guarantee

When the user creates an account, a generator is used to create a random salt and pepper for each user which is then used to hash the password of the account, and these are then stored in the database.

Main Success Scenario

1. The user creates an account on the application
2. A generator function is performed to create a random string for the salt and a random string for the pepper

3. The salt and pepper then get added to the user's plaintext password before being hashed
4. The hashed password and the plaintext salt and pepper get stored in the database to be retrieved for the next login for that user

Alternative Flow

None

Encrypt and Store Passwords

Primary Actor

User

Preconditions

User must have a password inputted to be encrypted

Success Guarantee

The password inputted by the user is transformed using the AES algorithm then the encrypted passwords get sent to the users database

Main success Scenario

1. The user enters the data they wish to encrypt then click encrypt
2. The AES algorithm is called which transforms the user's data into ciphertext
3. The ciphertext then gets transferred into the users table in the database so only they can access it

Alternative Flow

1. The user doesn't input anything to be encrypted

Decrypting and Retrieving Passwords

Primary Actor

User

Preconditions

User must have a password stored in the tables for their database

Success Guarantee

The password(s) retrieved by the user is decrypted using the AES algorithm then the decrypted passwords are displayed to the user

Main success Scenario

1. The user clicks the view passwords option
2. The user's stored passwords are located and retrieved
3. The AES algorithm is called which transforms the user's data into plaintext
4. The plaintext data is then displayed to the user

Alternative Flow

1. The user doesn't have any passwords/data stored

Forgot Password

Primary Actor

User

Preconditions

User must have an account made for the application

Success Guarantee

The user must be able to reset their password if they forgot their password in order to regain access to their account

Main success Scenario

1. The user clicks the forgot password option on the login page
2. The user is sent an MFA confirmation to allow a password change or may need to answer a security question
3. The user is then able to change their password for their account

Alternative Flow

1. User enters the password that already exists for their account
2. User doesn't get the security question correct or doesn't confirm the MFA token

Change password

Primary Actor

User

Preconditions

User must have an account made for the application

Success Guarantee

The user can change their password by inputting their old password as confirmation of the user owning the account

Main success Scenario

1. The user clicks the change password option
2. The user is prompted for their old password and the new password they want to use
3. The user is asked to confirm the new password by typing it twice
4. If the new passwords enter match and the user clicks confirm the password will be changed for the account

Alternative Flow

1. The user doesn't know the old password
2. The user doesn't type the new password correctly, so they don't match as their asked for a new password to be typed twice

Delete Account

Primary Actor

User

Preconditions

User must have an account for the application

Success Guarantee

The user can delete their account and all the information to do with the account

Main success Scenario

1. The user clicks the delete account option
2. The user is prompted for their accounts password and are asked if their sure they want to delete the account
3. The account and all the information stored with the account is deleted

Alternative Flow

1. The user doesn't know their password to delete the account
2. The user chooses not to delete the account

Project Plan

#	Plan	Due Date	Deliverable
1	*Complete research manual*	25/11/2022	Word doc of all the research necessary to complete project
2	Presentation	16/12/2022	Presentation to discuss what ill be doing to complete the project
3	Functional spec	16/12/2022	Word doc of what functions the application will perform

4	Password Generator	10/1/2023	Code to randomly create strong passwords for the user to use
5	Main menu	15/1/2023	Code for the main menu that will be easy for users to navigate
6	Login/create account	30/1/2023	Code that will allow users to make an account and login
7	*Store/receive data*	15/2/2023	Code that will allow users to encrypt/decrypt and store/retrieve code from a database
8	Change/forgot password	30/2/2023	Code that will allow a user to change/reset their password
9	Delete Account	15/3/2023	Code that will allow a user to delete their accounts
10	Training/quizzes	30/3/2023	Code that will allow users to perform training on topics and take tests based on the training