import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import java.util.Base64;

import java.util.HashMap;

import java.util.Map;

import java.util.Random;

import java.util.Scanner;

public class PasswordManager {

private static final String ALGORITHM = "AES";

private static final int KEY\_SIZE = 128;

private SecretKey secretKey;

private Map<String, String> passwordStore;

private Scanner scanner;

public PasswordManager() throws Exception {

this.secretKey = generateSecretKey();

this.passwordStore = new HashMap<>();

this.scanner = new Scanner(System.in);

}

private SecretKey generateSecretKey() throws Exception {

KeyGenerator keyGenerator = KeyGenerator.getInstance(ALGORITHM);

keyGenerator.init(KEY\_SIZE);

return keyGenerator.generateKey();

}

private String encrypt(String plainText) throws Exception {

Cipher cipher = Cipher.getInstance(ALGORITHM);

cipher.init(Cipher.ENCRYPT\_MODE, secretKey);

byte[] encryptedBytes = cipher.doFinal(plainText.getBytes());

return Base64.getEncoder().encodeToString(encryptedBytes);

}

private String decrypt(String cipherText) throws Exception {

Cipher cipher = Cipher.getInstance(ALGORITHM);

cipher.init(Cipher.DECRYPT\_MODE, secretKey);

byte[] decodedBytes = Base64.getDecoder().decode(cipherText);

byte[] decryptedBytes = cipher.doFinal(decodedBytes);

return new String(decryptedBytes);

}

private String generateRandomPassword(int length) {

String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&\*()\_-+=<>?";

Random random = new Random();

StringBuilder password = new StringBuilder(length);

for (int i = 0; i < length; i++) {

password.append(characters.charAt(random.nextInt(characters.length())));

}

return password.toString();

}

public void addPassword() throws Exception {

System.out.print("Enter the account name: ");

String account = scanner.nextLine();

System.out.print("Enter the desired password length: ");

int length = scanner.nextInt();

scanner.nextLine(); // Consume newline

String password = generateRandomPassword(length);

String encryptedPassword = encrypt(password);

passwordStore.put(account, encryptedPassword);

System.out.println("Password generated and stored successfully.");

System.out.println("Generated Password: " + password);

}

public void retrievePassword() throws Exception {

System.out.print("Enter the account name: ");

String account = scanner.nextLine();

if (passwordStore.containsKey(account)) {

String encryptedPassword = passwordStore.get(account);

String decryptedPassword = decrypt(encryptedPassword);

System.out.println("Retrieved Password: " + decryptedPassword);

} else {

System.out.println("Account not found.");

}

}

public void displayMenu() {

System.out.println("Password Manager");

System.out.println("1. Add a password");

System.out.println("2. Retrieve a password");

System.out.println("3. Exit");

}

public void run() throws Exception {

int choice;

do {

displayMenu();

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

addPassword();

break;

case 2:

retrievePassword();

break;

case 3:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Please try again.");

}

System.out.println();

} while (choice != 3);

}

public static void main(String[] args) {

try {

PasswordManager passwordManager = new PasswordManager();

passwordManager.run();

} catch (Exception e) {

e.printStackTrace();

}

}

}