

Password Management Application

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

Yash Sawant Roll No. 40
Manas Raut Roll No. 36
Vignesh Pillutla Roll No.27
Mihir Singh Roll No.49

Supervisor:

Prof. Suchetadevi Gaikwad.



Department of Information Technology

Atharva College of Engineering

Year: 2020-21



ATHARVA COLLEGE OF ENGINEERING

MALAD (W), MUMBAI 400 095

YEAR: 2020-2021

CERTIFICATE

This is to certify that

Manas Avinash Raut (Roll No. 36)

*have submitted the Mini Project report for the requirements of the Bachelor of
Engineering in Information Technology satisfactorily*

on

“Password Management Application”

*As prescribed by the **University of Mumbai** Under the guidance of*

PROJECT GUIDE

H.O.D.

PRINCIPAL

INTERNAL EXAMINER

COLLEGE SEAL

EXTERNAL EXAMINER

Approval for S. E. Mini Project Report

This project report entitled Password Management Application by **Manas Avinash Raut** is approved for the degree of ***Bachelor of Engineering in Information Technology***

Examiners:

1. -----

(Internal Examiner Name & Sign)

2. -----

(External Examiner Name & Sign)

Date: 08/12/2020

Place: Mumbai

Contents

Abstract

1. Introduction

- 1.1 Introduction
- 1.2 Motivation
- 1.3 Problem Statement(2 to 3 lines)
- 1.4 Objectives(1 or 2 lines)

2. Literature Survey

- 2.1 Survey of Existing system
- 2.2 Limitation Existing system or research gap
- 2.3 Mini Project contribution

3. Proposed System

- 3.1 Architecture/Framework
- 3.2 Algorithm and Process Design
- 3.3 Software Requirement Specification
- 3.4 Implementation Code(write classes as heading)
- 3.5 Output Screenshots(with title of each output)

4. Conclusion(5 to 6 lines)

5. Future work

References (include IEEE Paper names. No Wikipedia.com, google.com)

Acknowledgement

Password Management Application

1. Introduction:

1.1 Motivation:

World is currently going through a internet revolution. More and more people are coming online for simple social media reasons to online payments. Thus one problem always arrives which is data protection. A simple password is used to protect all this data which itself is vulnerable. Thus finding a method to protect this passwords motivated us to choose this topic.

1.2 Problem statement:

The current scenario on internet is to login on different websites to with different usernames with different passwords which had to be remembered for login. Users may forgot their passwords which result wastage of their time in trying to reset their passwords. To avoid this people set similar passwords for most of their accounts which compromises their security.

The password management application saves the passwords of different websites in one single application and uses encryption methods to securely store them. So user can access all saved passwords by just login into only one application. The application also suggests strong passwords for websites to increase security of current password. Application also aims to make storing the passwords as simple as possible but without compromising on security.

1.3 Objectives:

1. To store passwords in a single application to make them easily accessible.
2. To secure them by applying encryption.

Password Management Application

2. Literature Survey:

2.1) Survey of existing system:

In the Era of Online World, people regularly login on different websites. They tend to forgot their passwords, which in result have to waste time on trying to login, calling for help desk, proving their identity,etc. Each problem incident may consume 20-30 minutes of user. This can create great havoc for users who are just to login for their simple tasks.

2.2 Limitation of Existing system or Research gap:

Many people use similar passwords on most of their websites so they do not forget them. But this makes them insecure as a person only needs to know one password in order to get access of all the accounts. People also write down their passwords in certain applications where they are very vulnerable.

2.3 Mini project contribution:

The application has contribution of saving user's time by storing websites usernames and passwords in a single application while passwords are in encrypted form and can be only accessed by particular user through application's username and password. Application also provides double layer of security by using privacy pin as means to view or edit stored passwords.

Password Management Application

3. Proposed system:

Architecture and framework:

All windows are independent and access methods of two common classes. First is UserDBUtilities class. It contains all database methods like insert , delete, update and retrieve. All these functions are applied on UserDetails database. Another class named PrivateDB is used to perform database operations on PrivateDB database to store private keys.

All the passwords are stored in encrypted form to secure data. These passwords are encrypted using another common class named Encryption. These class contains two methods named decrypt and encrypt which are used by all other classes. Encryption is of RSA standard.

RSA is a asymmetric encryption system meaning it uses two keys known as public and private key which are different. The public key is known and is used for encryption whereas the private key is hidden and is used for decryption. The message cannot be decrypted without the private key. We use two random large prime numbers in this process which are used in key generation process for encryption and decryption. This encryption is based on the fact that finding factors of large composite numbers are really difficult and takes a long time. Finding prime factors of a composite number is known as prime factorization.

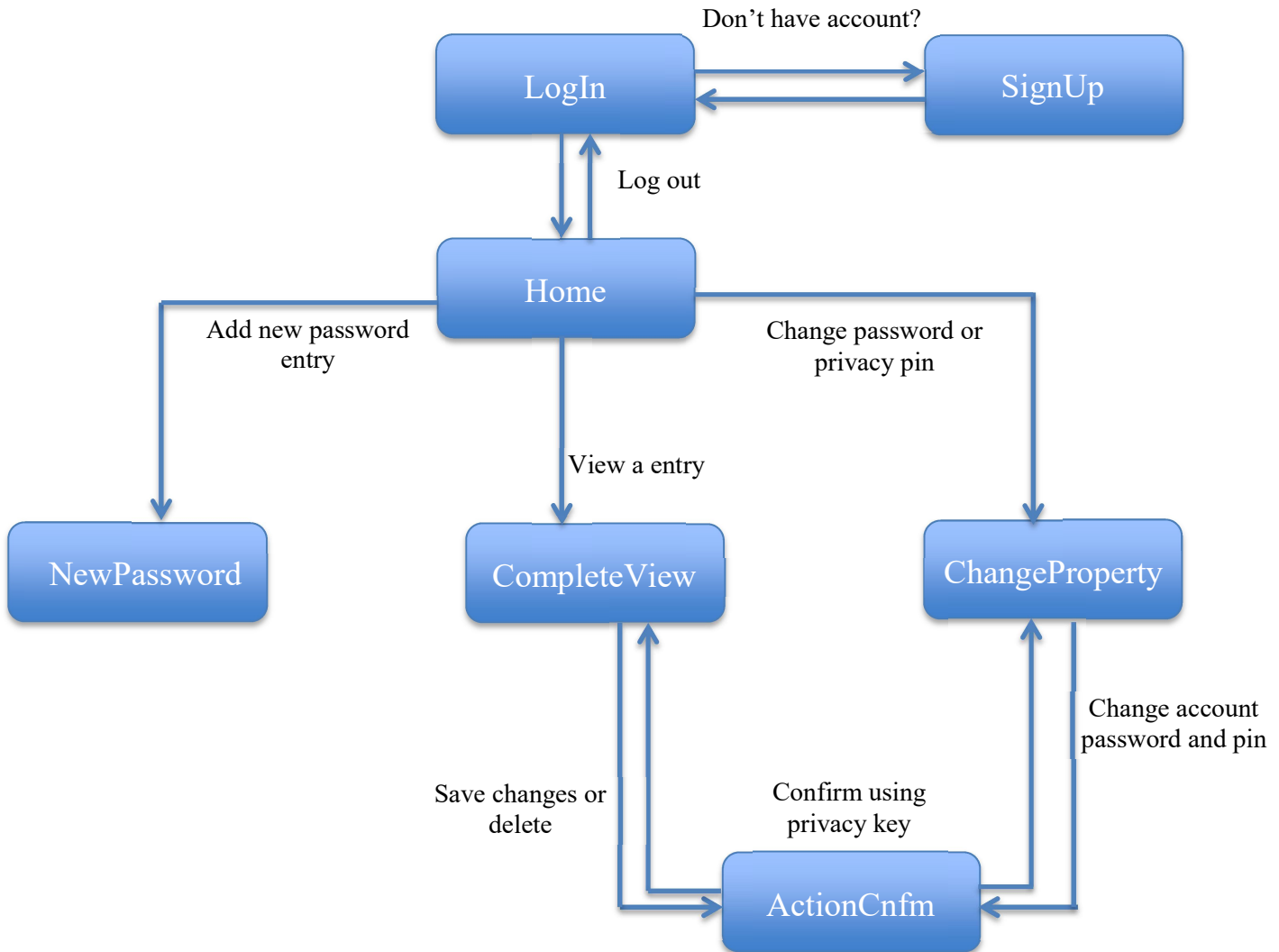
Software Requirement specification:

1. Requires latest Java JDK which includes java AWT, java Swing.
2. IDE (like IntelliJ IDEA, VSCode , Eclipse)
3. Windows OS

Password Management Application

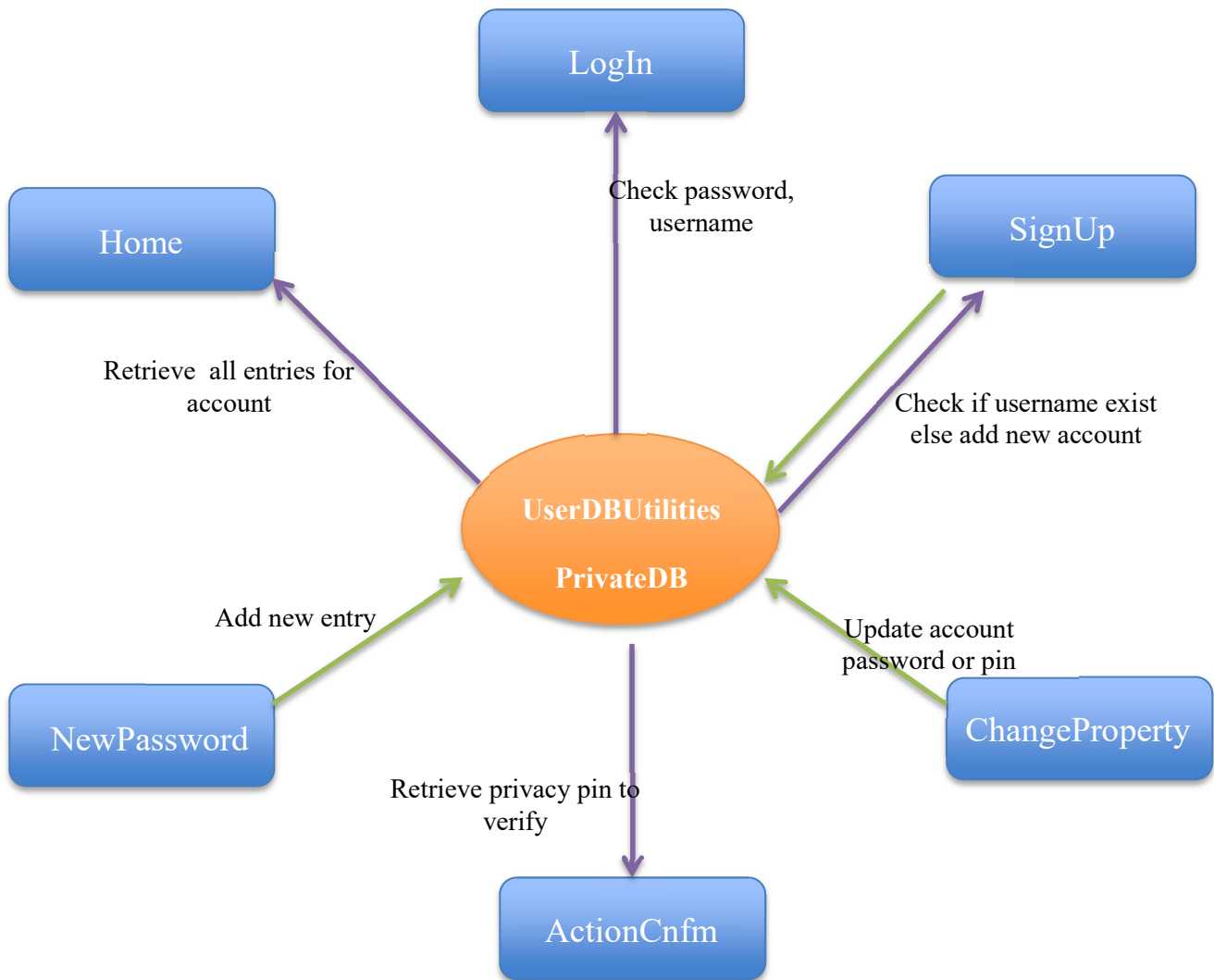
Algorithm and process design:

For UI:



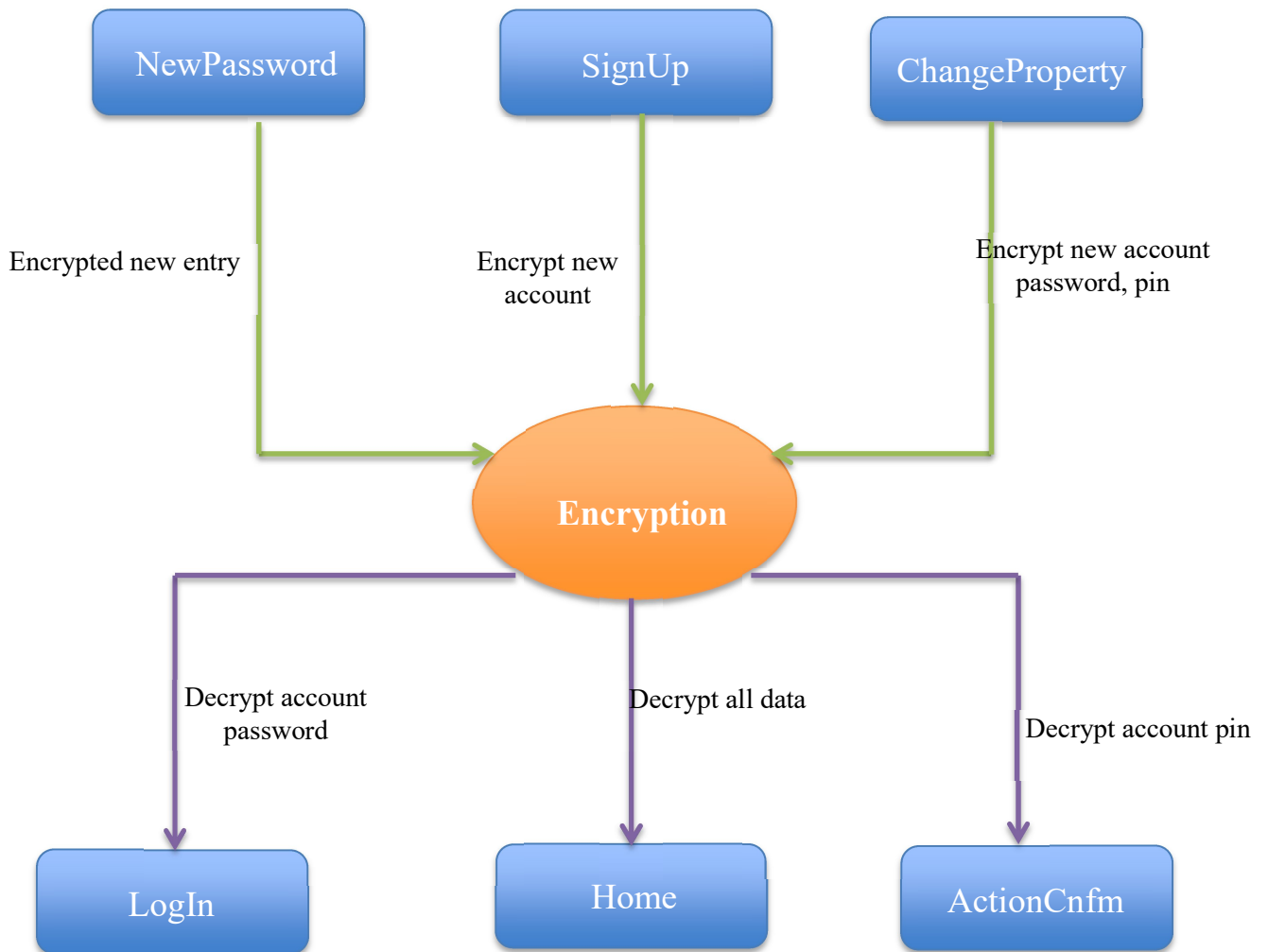
Password Management Application

For Database:



Password Management Application

For Encryption:



Password Management Application

Implementation Code:

LogIn class:

```
package java_miniproject;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class LogIn {
    JFrame frame;
    JTextField userField;
    JPasswordField pswordField;

    LogIn() {
        frame = new JFrame("PassWarden");
        frame.setSize(750, 450);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BoxLayout(frame.getContentPane(), BoxLayout.X_AXIS));
        initUi();
        frame.setVisible(true);
    }

    private void initUi() {

        JPanel sidebanner = new JPanel();
        sidebanner.setLayout(new BoxLayout(sidebanner, BoxLayout.Y_AXIS));
        sidebanner.setBackground(Styles.primary_violet);
        sidebanner.setMaximumSize(new Dimension(375, 450));
        JPanel panel_1 = new JPanel(new FlowLayout());
        JPanel panel_2 = new JPanel(new FlowLayout());
        panel_1.setBackground(Styles.primary_violet);
        panel_2.setBackground(Styles.primary_violet);
        ImageIcon appiconImage = new
        ImageIcon("src\\java_miniproject\\assets\\app_logo.png");
        frame.setIconImage(appiconImage.getImage());
        JLabel appIcon = new JLabel(appiconImage);
        JLabel appName = new JLabel("PassWarden");
        JLabel slogan = new JLabel("Never forget another password");
        appName.setFont(new Font("Roboto", Font.BOLD, 26));
        slogan.setFont(new Font("Calibri", Font.ITALIC, 16));
        appName.setForeground(Color.WHITE);
        slogan.setForeground(Color.WHITE);
    }
}
```

Password Management Application

```
JPanel panel_3 = new JPanel(new FlowLayout());
panel_3.setBackground(Styles.primary_violet);
JLabel label_1 = new JLabel("Don't have a account ? ");
label_1.setForeground(Color.WHITE);
JButton signupBtn = new JButton("Sign up");
signupBtn.setBackground(Color.WHITE);
signupBtn.setFocusPainted(false);
signupBtn.setBorder(Styles.black_button_border);

panel_1.add(appIcon);
panel_1.add(appName);
panel_2.add(slogan);
panel_3.add(label_1);
panel_3.add(signupBtn);

sidebanner.add(Box.createVerticalStrut(150));
sidebanner.add(panel_1);
sidebanner.add(panel_2);
sidebanner.add(Box.createVerticalStrut(50));
sidebanner.add(panel_3);

JPanel parent = new JPanel();
parent.setLayout(new BoxLayout(parent, BoxLayout.Y_AXIS));
parent.setBackground(Styles.primary_white);
parent.setMaximumSize(new Dimension(500, 450));

JPanel panel1 = new JPanel(new FlowLayout(FlowLayout.LEFT));
panel1.setBackground(Styles.primary_white);
JLabel label1 = new JLabel("Log In");
label1.setFont(Styles.big_label_font);
label1.setForeground(Styles.primary_violet);

JPanel panel2 = new JPanel(new FlowLayout());
panel2.setBackground(Styles.primary_white);
JLabel label2 = new JLabel("Username: ");
userField = new JTextField(16);
userField.setBackground(Color.WHITE);
userField.setFont(Styles.text_field_font);
userField.setBorder(Styles.text_field_border);

JPanel panel3 = new JPanel(new FlowLayout());
panel3.setBackground(Styles.primary_white);
JLabel label3 = new JLabel("Password");
pswordField = new JPasswordField(16);
pswordField.setEchoChar('*');
pswordField.setBackground(Color.WHITE);
```

Password Management Application

```
pswordField.setFont(Styles.text_field_font);
pswordField.setBorder(Styles.text_field_border);

JPanel panel4 = new JPanel(new FlowLayout());
panel4.setBackground(Styles.primary_white);
JButton loginBtn = new JButton("LogIn");
JButton clearBtn = new JButton("Clear");
loginBtn.setBackground(Styles.primary_button_color);
loginBtn.setForeground(Color.WHITE);
loginBtn.setBorder(Styles.white_button_border);
loginBtn.setFocusPainted(false);
clearBtn.setBackground(Styles.primary_button_color);
clearBtn.setForeground(Color.WHITE);
clearBtn.setBorder(Styles.white_button_border);
clearBtn.setFocusPainted(false);

panel1.add(Box.createHorizontalStrut(150));
panel1.add(label1);
panel2.add(label2);
panel2.add(userField);
panel3.add(label3);
panel3.add(pswordField);
panel4.add(loginBtn);
panel4.add(clearBtn);
parent.add(Box.createVerticalStrut(50));
parent.add(panel1);
parent.add(panel2);
parent.add(panel3);
parent.add(panel4);
frame.add(sidebanner);
frame.add(parent);
frame.getRootPane().setDefaultButton(loginBtn);
clearBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        pswordField.setText("");
        userField.setText("");
        pswordField.setBorder(Styles.text_field_border);
    }
});
loginBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {
            userField.setBorder(Styles.text_field_border);
            UserDBUtilities userDb = new UserDBUtilities();
            PrivateDB privateDb = new PrivateDB();
            String text = userField.getText().strip();
```

Password Management Application

```
        if(!userDb.checkUser(text)){
            userField.setBorder(Styles.red_warning_border);
            return;
        }
        String appPswrd_E = userDb.getAccountPassword(text);
        String appPswrd_pk = privateDb.getAccPasswordKey(text);
        EncryptedData data = new EncryptedData();
        data.encryptedPassword = appPswrd_E;
        data.privateKey = appPswrd_pk;
        userDb.endConnection();
        privateDb.endConnection();
        if(String.valueOf(pswordField.getPassword()).equals(Encryption.decrypt(data)))
    {
        new Home(text);
        frame.setVisible(false);
    } else {
        pswordField.setBorder(Styles.red_warning_border);
    }
    } catch( Exception e) {
        System.out.println(e.getMessage());
        System.out.println("Exception at login");
    }
    }
    });
    signupBtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            frame.setVisible(false);
            new SignUp(frame);
        }
    });
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new LogIn();
        }
    });
}
```

Password Management Application

Encryption class:

```
package java_miniproject;

//import javax.crypto.Cipher;

import javax.crypto.Cipher;
import java.nio.charset.StandardCharsets;
import java.security.*;
import java.security.spec.PKCS8EncodedKeySpec;
import java.util.Base64;

class EncryptedData {
    //encrypted password will go in User data base
    //privateKey will go in private data base
    String encryptedPassword,privateKey;
}

public class Encryption {
    public static EncryptedData encrypt(String password)throws Exception {
        //Creating a Signature object
        Signature sign = Signature.getInstance("SHA256withRSA");

        //Creating KeyPair generator object
        KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("RSA");

        //Initializing the key pair generator
        keyPairGen.initialize(2048);
        //Generate the pair of keys
        KeyPair pair = keyPairGen.generateKeyPair();

        //Getting the public key from the key pair
        PublicKey publicKey = pair.getPublic();
        PrivateKey privateKey = pair.getPrivate();
        //Creating a Cipher object
        Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
        byte[] privateKeyBytes=privateKey.getEncoded();
        String privateKeyStr=Base64.getEncoder().encodeToString(privateKeyBytes);
        cipher.init(Cipher.ENCRYPT_MODE, publicKey);
        byte[] passwordBytes=password.getBytes();
        cipher.update(passwordBytes);
        byte[] cipherText = cipher.doFinal();
        EncryptedData data=new EncryptedData();
```

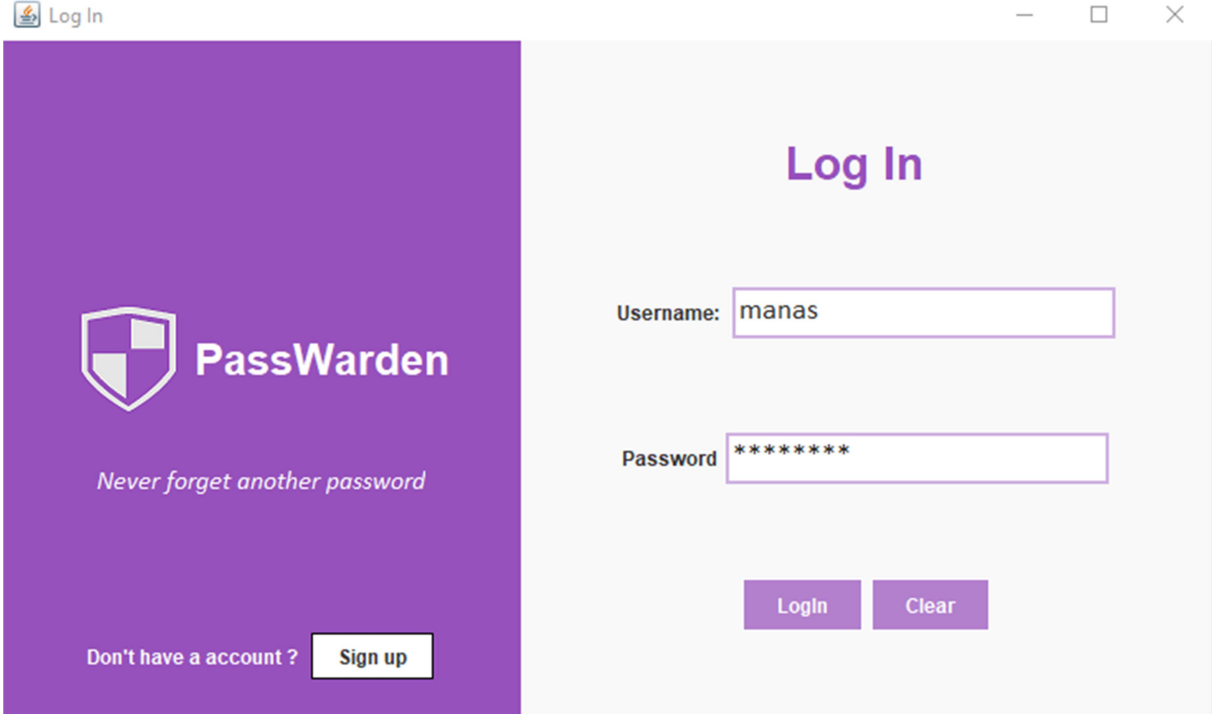
Password Management Application

```
data.encryptedPassword=Base64.getEncoder().encodeToString(cipherText);
data.privateKey=privateKeyStr;
return data;
}
public static String decrypt(EncryptedData data)throws Exception{
    Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
    String encryptedKey=data.privateKey;
    byte[]
privateKeyByte=Base64.getDecoder().decode(encryptedKey.getBytes(StandardCharsets.UTF
_8));
    PKCS8EncodedKeySpec spec=new PKCS8EncodedKeySpec(privateKeyByte);
    KeyFactory keyFact=KeyFactory.getInstance("RSA");
    PrivateKey privateKey=keyFact.generatePrivate(spec);
    cipher.init(Cipher.DECRYPT_MODE, privateKey);
    byte[]
cipherText=Base64.getDecoder().decode(data.encryptedPassword.getBytes(StandardCharsets
.UTF_8));
    //Decrypting the text
    byte[] decipheredText = cipher.doFinal(cipherText);
    return new String(decipheredText);
}
}
```


Password Management Application

Output Screenshots:

1. LogIn form:

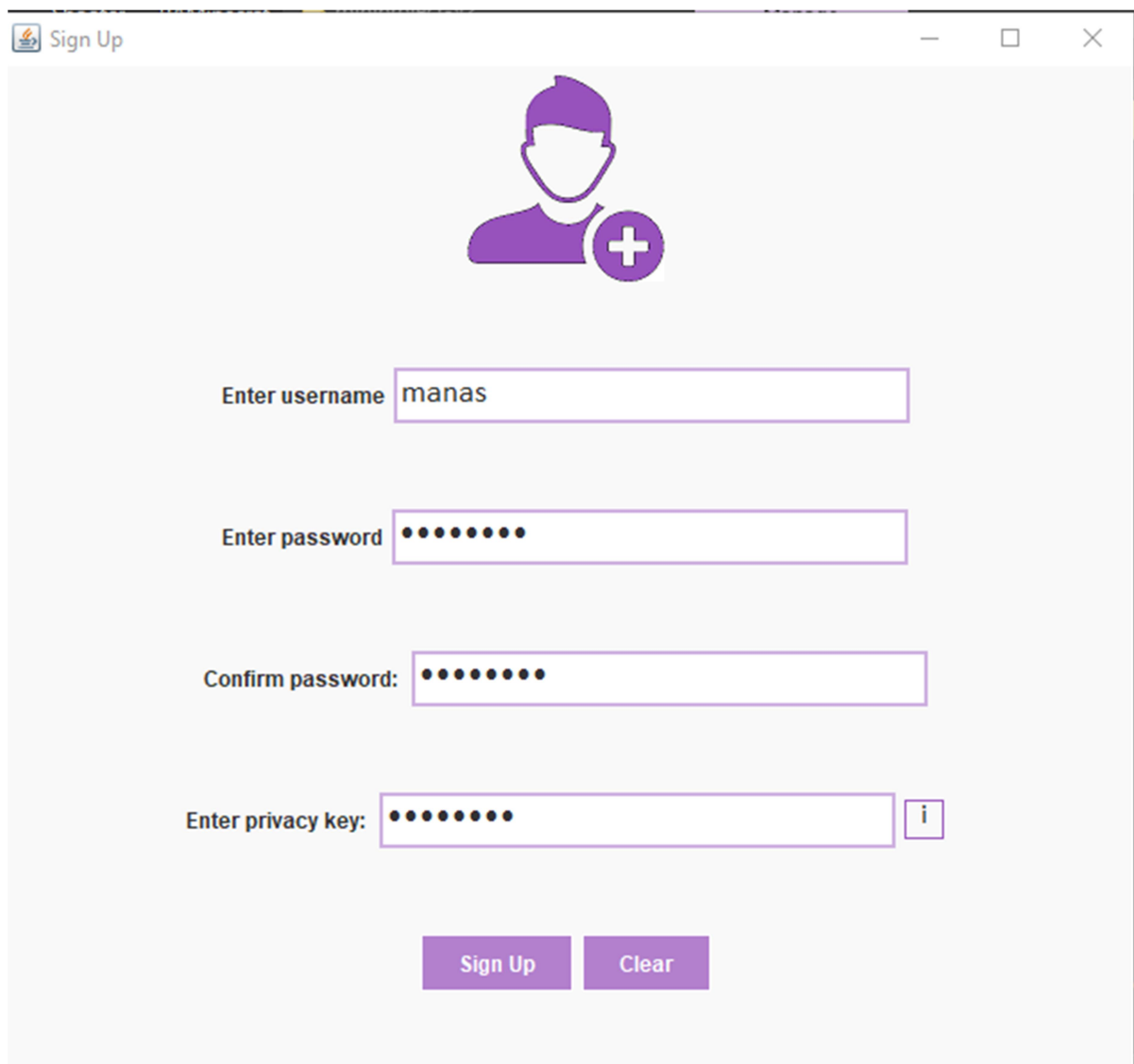


The screenshot shows a web application window titled "Log In". The window is split into two main sections. The left section has a purple background and features the "PassWarden" logo (a shield with a cross) and the text "Never forget another password". At the bottom of this section, there is a link "Don't have a account ?" and a "Sign up" button. The right section has a light gray background and contains the "Log In" title. Below the title, there are two input fields: "Username:" with the value "manas" and "Password:" with the value "*****". At the bottom of the right section, there are two buttons: "LogIn" and "Clear".

It is a log in form to log in the application using application account password. While logging it checks if user exists, if not then highlights username textfield to indicate user is wrong. It also includes button open sign up window.

Password Management Application

SignUp







The screenshot shows a web browser window titled "Sign Up". The page has a light gray background. At the top center is a purple icon of a person's head and shoulders with a white plus sign inside a purple circle. Below this icon are four input fields, each with a label to its left. The first field is labeled "Enter username" and contains the text "manas". The second field is labeled "Enter password" and contains ten black dots. The third field is labeled "Confirm password:" and also contains ten black dots. The fourth field is labeled "Enter privacy key:" and contains ten black dots. To the right of the fourth field is a small purple square button with a white lowercase letter 'i'. At the bottom center of the form are two purple buttons: "Sign Up" and "Clear".

SignUp form allows us to create new accounts for application. It is also used to create privacy pin fro application.

Password Management Application

Home page

 User: manas



User: manas

You have 4 passwords saved.

Sr No..	Website	Username
1.	facebook.com	manasFacebook
2.	Google.com	manasGoogle
3.	instagram.com	manasInstagram
4.	amazon.com	manasAmazon

Add new

Edit password

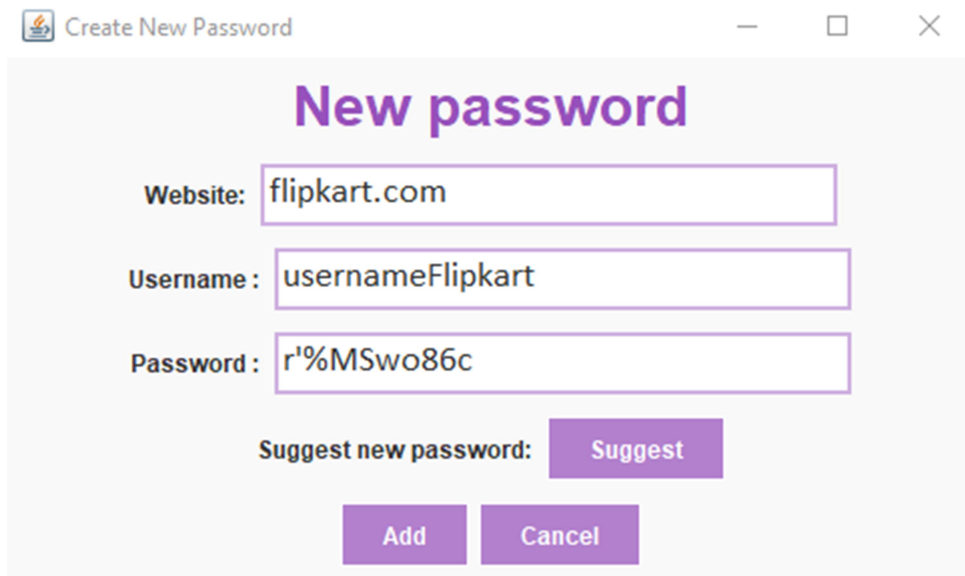
Edit key

Log Out

Home page consists list of all entries in application for the logged in account. It is also used to navigate to other windows.

Password Management Application

NewPassword:



Create New Password

New password

Website: flipkart.com

Username: usernameFlipkart

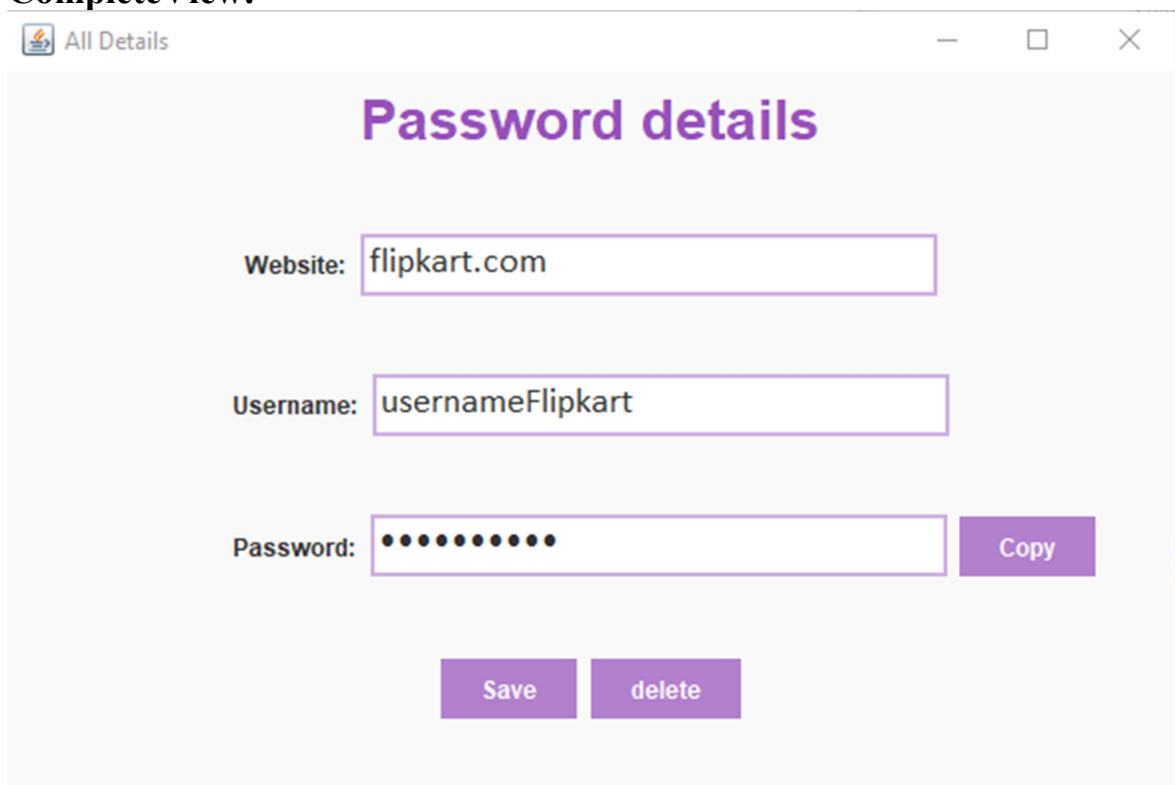
Password: r!%MSwo86c

Suggest new password: Suggest

Add Cancel

This window is used to create add a new password entry in application. It also suggests strong passwords.

CompleteView:



All Details

Password details

Website: flipkart.com

Username: usernameFlipkart

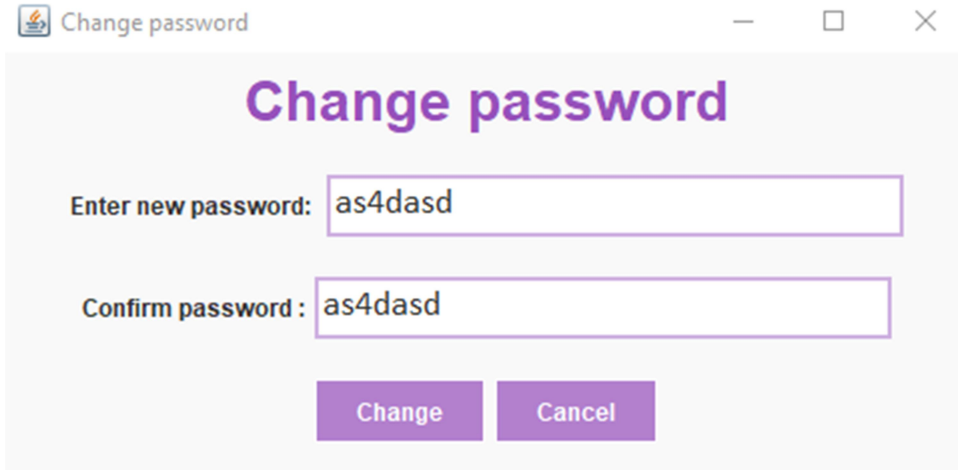
Password: ●●●●●●●● Copy

Save delete

Password Management Application

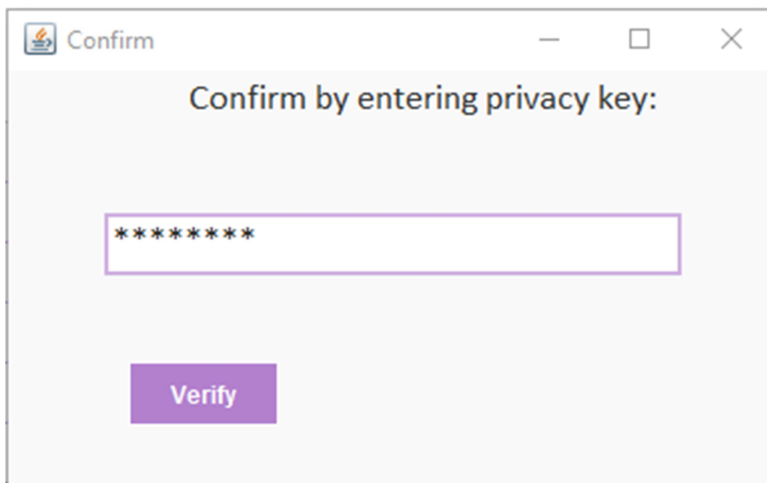
Complete view displays all details of a particular entry and also provides option to copy password on clipboard.

ChangeProperty:

A screenshot of a 'Change password' dialog box. The title bar shows a small icon and the text 'Change password'. The dialog has a light gray background. At the top, the title 'Change password' is displayed in a large, bold, purple font. Below the title, there are two text input fields. The first is labeled 'Enter new password:' and contains the text 'as4dasd'. The second is labeled 'Confirm password:' and also contains 'as4dasd'. At the bottom of the dialog, there are two buttons: 'Change' and 'Cancel', both in a purple color.

To change password or privacy pin of application account.

ActionCnfm:

A screenshot of a 'Confirm' dialog box. The title bar shows a small icon and the text 'Confirm'. The dialog has a light gray background. At the top, the text 'Confirm by entering privacy key:' is displayed in a bold, dark gray font. Below this text is a single text input field containing eight asterisks '*****'. At the bottom of the dialog, there is a single button labeled 'Verify' in a purple color.

A dialog box to enter privacy pin to perform all operations like change password, delete a password, and view all details of a password.

Password Management Application

4.) Conclusion:

1. Thus we conclude that we have a application made in Java to help people to protect their passwords indirectly protecting their invaluable data.
2. Its GUI aims to make application as easy as possible to use.
3. So all these features make the application ready to general use.

6.) Future Work:

1. In future we application can be hosted on a server so it become accessible on any device with internet connection.
2. Also more than just passwords , sensitive information like official documents and notes can be stored in the application.

7. References:

1. RSA algorithm :- Wikipedia.com
2. Java the complete reference :- by Herbert Schildt