LockedMe.com

A personal locker for all your passwords

Here at LockedMe .com, you can save all your passwords along with your usernames for any number of websites in a single location and you don’t have to worry about forgetting them in the future.

2020

Manjunath V

Microsoft

8/23/2020

**Heading Of The Project**

**LockedMe.com** is a prototype of a digital locker where a user can store his/her credentials so in one place so that the user don’t have to remember each and every credential.

**Project Description**

In order to avail the facilities provided by LockedMe.com , the user will have to register to the locker.

Once the registration is complete, the user can login and avail multiple facilities provided by LockedMe.com, which included.

* Adding username and password of multiple websites to the locker.
* Removing username and password of multiple websites from the locker.
* Changing password of a website already added in the locker.
* Displaying all the websites in the user’s locker along with their usernames and passwords.
* User can also delete is LockedMe.com account.
* User can navigate freely in the console and doesn’t have to run the code again to perform another operation.
* Last but not the least, the user can log-out, once he/she is done using the locker.

**Tech Stack**

The code for this project is written in **JAVA** my implementing **Agile-Scrum** methodology. Numerous Java techniques have been implemented in this project’s code. Some of them are

* Encapsulation

Encapsulation has been used to secure user credentials during log in and registration processes and also to identify username during the user processes post log-in.

* Regular Expression

Regular-Expression has been used during the registration process to ensure that the password is Strong enough.

* File Handling Techniques

User data is read and written into particular user files when a new website info is added or when an existing website info is modified or deleted.

User log in credential is added and removed when a new user is registered and an existing account is deleted respectively.

Login credentials of the user are also fetched from the file during the log-in process.

* Exception Handling Techniques

Exception handling is used along with the file handling process in order to deal with any exceptions that may occur.

* Collections (Hash-map)

The username and passwords of Locked Me accounts are fetched from file as a hash map with username as key and password as value. The website details stored by the user are also fetched as a hash Map with website as key and an array containing username and password as the value.

* Recursion

Used recursive Methods so that the user doesn’t have to re-run the code to perform another operation or when the user action fails. She/he will have the option to go back to the main menu or try again or log out.

**Project Stories (Agile - Scrum)**

**Sprint 1:**

* As a developer I want to build a registration page for the user to register by entering username and password
* As a developer I want to build a log-in page for the user.

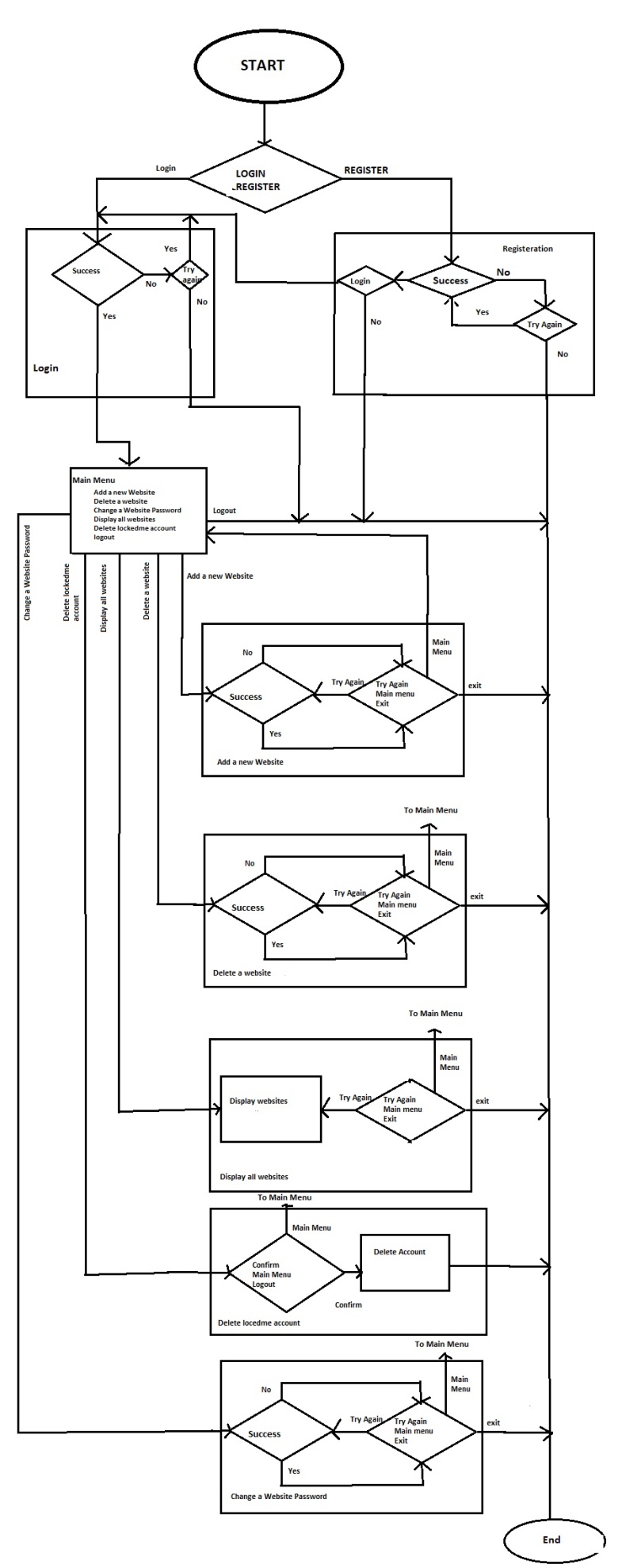
**Sprint 2:**

* As a user I want the password to be more strong
* As a user I need to have the option to add a website along with its username and password into the database.
* As a user I need to have to option to view the websites along with its username and password.
* As a user I need to have an option to change a website password.
* As a user I need to have the option to delete a website from my database

**Sprint 3:**

* As a user I need to have the facility to delete my locker account.
* As a user I need to have the facility to logout of my locker account.
* As a user I need to have an option attempt the task again or get back to the main menu instead of running the code again
* As a user I need to have an option to try log in again if the attempt fails.
* As a user I need to have an option to **log** in once registration completes.

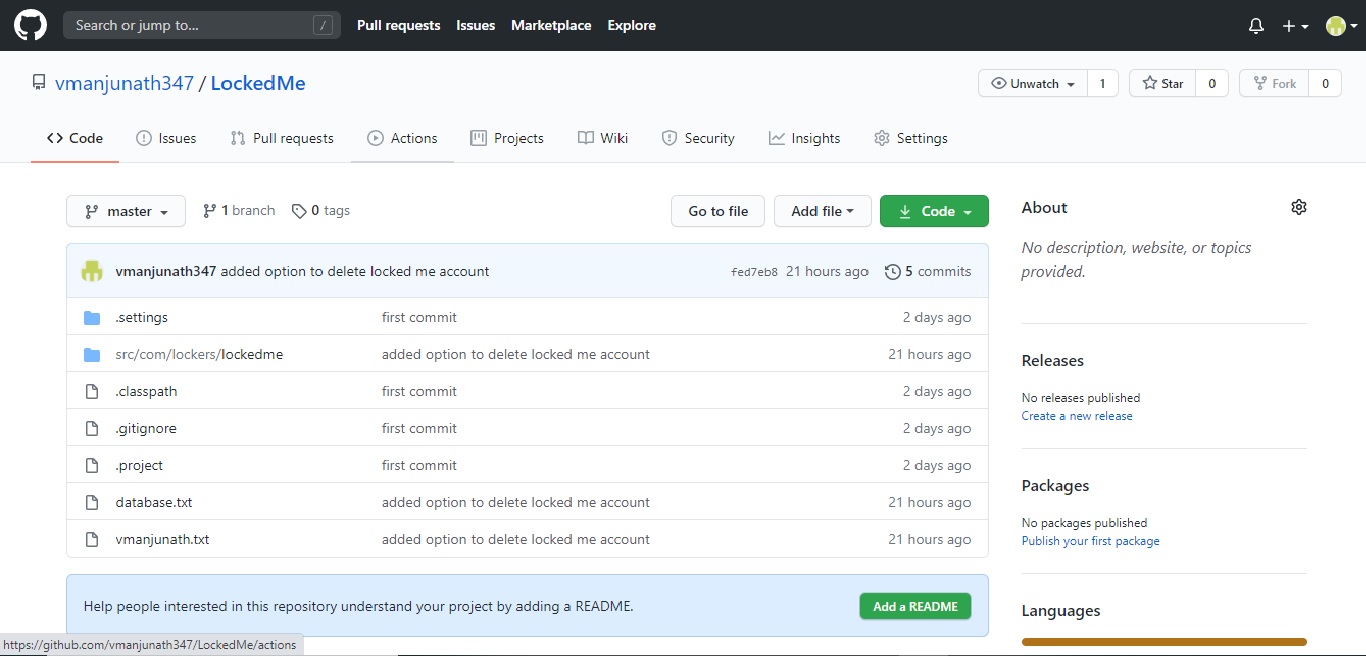
**Program Flow Chart**

****

**Project Git Repositories**

**Link:** <https://github.com/vmanjunath347/LockedMe.git>

**Clone git:** git clone <https://github.com/vmanjunath347/LockedMe.git>



**How To run The Project**

1. Clone Project:

git clone <https://github.com/vmanjunath347/LockedMe.git>

## 2. Open [src](https://github.com/vmanjunath347/LockedMe/tree/master/src) -> [com](https://github.com/vmanjunath347/LockedMe/tree/master/src/com) -> [lockers](https://github.com/vmanjunath347/LockedMe/tree/master/src/com/lockers) -> **lockedme** -> UserInterface.java

## 3. Right click -> run as java application

**Directory Structure**



Note: UserInterface.java contains the main method.

.txt files are sample database files.

**Source Code Files**

## **UserInterface.java**

|  |
| --- |
| package com.lockers.lockedme;  import java.util.HashMap;  import java.util.Scanner;  public class UserInterface {  public static void main(String[] args) {    // displays welcome message  welcomeScreen();  //performs and login and registeration operations according to user input  authentication();      }    // displays welcome message  public static void welcomeScreen() {  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("\* \*");  System.out.println("\* Welcome to LockedMe.com \*");  System.out.println("\* Your personal locker \*");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  }    //performs and login and registeration operations according to user input  public static void authentication() {    System.out.println("\n \nEnter 1 to Login");  System.out.println("Enter 2 to Register");  System.out.println("Enter 3 to Exit");    Scanner input = new Scanner(System.in);    String inputNumber = input.nextLine();//user input    switch(inputNumber) {    case "1":  //login  Login loginObj= new Login();  loginObj.login();// Handles the login operations of a user    UserProccess processObj = new UserProccess();  processObj.userProcesses(loginObj);  break;    case "2":  //register  Registration Registration = new Registration();  Registration.fetchAuthInfo();//handles registeration operation at user interface  authenticationAfterRegistration();//provides user option to log in after registration or exit program  break;  case "3":  System.out.println("exit success");  break;  default:  System.out.println("Invalid Input. \nRun code again to try again");  tryagain(input);//runs autentication again  }      }    //provides user option to log in after registration or exit program  public static void authenticationAfterRegistration() {    System.out.println("\n \nEnter 1 to Login");  System.out.println("Enter 2 to Exit");    Scanner input = new Scanner(System.in);    String inputNumber = input.nextLine();//user input    switch(inputNumber) {    case "1":  //login  Login loginObj= new Login();  loginObj.login();// Handles the login operations of a user    UserProccess processObj = new UserProccess();  processObj.userProcesses(loginObj);  break;    case "2":  System.out.println("exit success");  break;  default:  System.out.println("Invalid Input. \nRun code again to try again");  tryagain(input);//runs autentication again  }      }    //runs autentication again  public static void tryagain(Scanner input) {  System.out.println("\n---------------------");  System.out.println("Enter 1 to try again.");  System.out.println("Enter 2 to exit.");    String userinput= input.nextLine();    switch(userinput) {  case "1":  authentication();  break;  case "2":  System.out.println("exit success");  return;  default:  System.out.println("Invalid Input");  }  }  } |

## **Registration.java**

|  |
| --- |
| package com.lockers.lockedme;import java.io.File;import java.io.FileNotFoundException;import java.io.FileWriter;import java.io.IOException;import java.util.HashMap;import java.util.Scanner;import java.util.regex.Pattern;//for methods asociated with registerationpublic class Registration {private HashMap<String, String> allUserCredentails = new HashMap<String,String>();//to store username and passwords of already existing user.//fetches user name and password from database file and adds them to the hashmapprivate void setAllUserCredentionals() {File fileObj = new File("database.txt");Scanner scannerReader;try {if(fileObj.exists()==false)fileObj.createNewFile();scannerReader = new Scanner(fileObj);int lineCounter=0;String tempUser= new String("");while(scannerReader.hasNextLine()) {if(lineCounter%2==0)tempUser=scannerReader.nextLine();else if(lineCounter%2!=0)allUserCredentails.put(tempUser, scannerReader.nextLine());lineCounter++;}scannerReader.close();}catch (FileNotFoundException e) {e.printStackTrace();} catch (IOException e) {// TODO Auto-generated catch blocke.printStackTrace();}}//handles registration operation at user interfacepublic void fetchAuthInfo() {Scanner input = new Scanner(System.in);System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");System.out.println("\* \*");System.out.println("\*Welcome To Registeration Page\*");System.out.println("\* \*");System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");System.out.println("\n\*\*\*\*\*\*\*\*\*\*Enter User Name\*\*\*\*\*\*\*\*\*\*");String userName= input.nextLine();//returns if entered username is nullif(userName.equals("")) {System.out.println("Username is emply.");tryAgaintoRegister(input);// runs fetchAuthInfo againreturn;}setAllUserCredentionals();//check if username already existsif(userNameExists(userName)==true) {System.out.println("Username already exists.");tryAgaintoRegister(input);// runs fetchAuthInfo againreturn;}System.out.println("\*\*\*\*\*\*\*\*\*\*Enter a password\*\*\*\*\*\*\*\*\*\*");System.out.println("\nNote: Password should be atleast 6 characters long,");System.out.println("should have atleast 1 uppercase character,");System.out.println("should have atleast 1 number value.");String password= input.nextLine();//if entered password is null, returnsif(password.equals("")) {System.out.println("Password is emply.");tryAgaintoRegister(input);// runs fetchAuthInfo againreturn;}//if passes doesnt pass the crieteria, returnsif(paswordCheck(password)==false) {tryAgaintoRegister(input);// runs fetchAuthInfo againreturn;}System.out.println("Confirm password");String password2=input.nextLine();//if both the password matches registeration happensif(password.equals(password2))addnewUser(userName,password);else {System.out.println("Passwords doesn't match");tryAgaintoRegister(input);// runs fetchAuthInfo again}}// runs fetchAuthInfo againprivate void tryAgaintoRegister(Scanner input) {System.out.println("------------");System.out.println("Enter 1 to try again");System.out.println("Enter 2 to Exit");String inputvalue=input.nextLine();switch(inputvalue) {case "1":fetchAuthInfo();break;case "2":System.out.println("Exited");return;//sbreak;default:System.out.println("Invalid Input");tryAgaintoRegister(input);//System.out.println("--------------------");break;}}// checks password criteriaprivate boolean paswordCheck(String psswrd) {if(psswrd.length()<6) {System.out.println("Password should be atleast 6 characters long");return false;}if(Pattern.matches(".\*[0-9].\*", psswrd)==false) {System.out.println("Password should have atleast 1 integer");return false;}if(Pattern.matches(".\*[A-Z].\*", psswrd)==false) {System.out.println("Password should have atleast 1 uppercase character.");return false;}return true;}//checks if username already existsprivate boolean userNameExists(String user) {if(allUserCredentails.containsKey(user))return true;return false;}//adds new user to the database fileprivate void addnewUser(String user,String passwrd) {File fileObj = new File("database.txt");FileWriter fileWriter = null;try {if(fileObj.exists()) {fileWriter = new FileWriter(fileObj,true);Scanner scannerReader = new Scanner(fileObj);if(scannerReader.hasNext())fileWriter.append("\n"+user);elsefileWriter.append(user);fileWriter.append("\n"+passwrd);System.out.println("Successfully Registered");}else {throw new FileNotFoundException("File is not Available with name "+fileObj.getName());}}catch (IOException e) {System.out.println("An Error Occurred");//e.printStackTrace();}try {fileWriter.close();}catch (IOException e) {// TODO Auto-generated catch blocke.printStackTrace();}}} |

## **Login.java**

|  |
| --- |
| package com.lockers.lockedme;import java.io.File;import java.io.FileNotFoundException;import java.io.IOException;import java.util.HashMap;import java.util.Scanner;// Handles the login operations of a userpublic class Login {private String username; //Stores the Confirmed user name;private String password; //Stores the confirmed password;private String inputUserName; //Stores the user name entered by the user;private String inputPassword; //Stores the password entered by the user;private String loginAttempt = new String("Failed"); //Stores the status of the login attempt//getter method for usernamepublic String getUsername() {return username;}//getter method for login attempt statuspublic String getLoginAttempt() {return loginAttempt;}//stores all the web sites as key and user name and password in an array as valueprivate HashMap<String, String> allCredentials = new HashMap<String,String>();//fetches all user credentials from database file to hashmap allCredentialsprivate void setAllUserCredentionals() {File fileObj = new File("database.txt");Scanner scannerReader;try {if(fileObj.exists()==false)fileObj.createNewFile();scannerReader = new Scanner(fileObj);int lineCounter=0;String tempUser= new String("");while(scannerReader.hasNextLine()) {if(lineCounter%2==0)tempUser=scannerReader.nextLine();else if(lineCounter%2!=0)allCredentials.put(tempUser, scannerReader.nextLine());lineCounter++;}scannerReader.close();}catch (FileNotFoundException e) {e.printStackTrace();} catch (IOException e) {// TODO Auto-generated catch blocke.printStackTrace();}}//deals with the login operations at the user endpublic void login() {System.out.println("==========================================");System.out.println("\* \*");System.out.println("\* Welcome to Login Page \*");System.out.println("\* \*");System.out.println("==========================================");System.out.println("Enter your Username:");Scanner input = new Scanner(System.in);inputUserName = input.nextLine();// input user name//if username is blank, returnif(inputUserName.equals("")) {System.out.println("Username is empty");tryAgaintoLogin(input);// runs log in operation againreturn;}setAllUserCredentionals();//fetches all user credentials from database file to hashmap allCredentials//if username already exists in database returnsif(usernameDoesnotExists()==true) {System.out.println("Username Invalid ");tryAgaintoLogin(input);// runs log in operation againreturn;}System.out.println("Enter your Password:");inputPassword = input.nextLine();//inputs password//if input password is empty, returnsif(inputPassword.equals("")) {System.out.println("Password is empty");tryAgaintoLogin(input);return;}//if password matched, loginif(passwordCheck()==true) {System.out.println("Login Success");loginAttempt="Success";// login status is changd to successusername=inputUserName;// added username is given as final usernamepassword=inputPassword;// added password is given as final pasdwordreturn;}else {System.out.println("Incorrect Password");tryAgaintoLogin(input);//runs log in againreturn;}}//gives an option to run login operation again or exitprivate void tryAgaintoLogin(Scanner input) {System.out.println("------------");System.out.println("Enter 1 to try again");System.out.println("Enter 2 to Exit");String inputvalue=input.nextLine();switch(inputvalue) {case "1":login();break;case "2":System.out.println("Exited");return;//sbreak;default:System.out.println("Invalid Input");tryAgaintoLogin(input);//System.out.println("--------------------");break;}}//checks if the entered password matches the password assigned to that username in hasmapprivate boolean passwordCheck() {if(allCredentials.get(inputUserName).equals(inputPassword))return true;return false;}//checks if a key == username exists in hasmapprivate boolean usernameDoesnotExists() {if(allCredentials.containsKey(inputUserName))return false;return true;}} |

## UserProccess.java

|  |
| --- |
| package com.lockers.lockedme;import java.io.File;import java.io.FileNotFoundException;import java.io.FileWriter;import java.io.IOException;import java.util.HashMap;import java.util.Iterator;import java.util.Map;import java.util.Map.Entry;import java.util.Scanner;//deals with the tasks a user can perform once loginpublic class UserProccess {private String username;// stores the username of the login userprivate HashMap<String, String[]> userDatabase = new HashMap <String,String[]>();// stores the website details of the log in user from the user file//stores all the web sites as key and user name and password in an array as valueprivate HashMap<String, String> allCredentials = new HashMap<String,String>();//this method is called outsie class to run any user processpublic void userProcesses(Login loginObj) {String loginAttemptStatus=loginObj.getLoginAttempt();// collect the login attempt status//return if log in is failedif(loginAttemptStatus.equals("Failed"))return;username=loginObj.getUsername();// gets login user name.distplayUserOptions();// displays the option of tasks available to the user.}// displays the option of tasks available to the user.private void distplayUserOptions() {System.out.println("\n-----------------------------------------------");System.out.println("Enter 1 to add a new website credential");System.out.println("Enter 2 to remove an existing website credential");System.out.println("Enter 3 to change an existing website password");System.out.println("Enter 4 to display all website details");System.out.println("Enter 5 to logout");System.out.println("Enter \* to delete your lockedMe account");System.out.println("-----------------------------------------------");operations(); // performs operation based on users input.}// performs operation based on users input.private void operations() {Scanner input=new Scanner(System.in);String inputNumber = input.nextLine();//collects user input//chooses tast based on user inputswitch(inputNumber) {case "1":addNewWebsite(input);// adds a new website to the user's databasebreak;case "2":deleteWebsite(input);// deletes a website from user's databasebreak;case "3":changeWebsitePassword(input); // changes password of a selected website in user's database.break;case "4":displayWebsiteDetails(); // diaplays all the websites along with it's username and password in users databasebreak;case "\*":delteLockedmeAccount(input); //deletes locked me accountbreak;case "5":System.out.println("Log out successfull");return;default:System.out.println("Invalid input");}}//deletes locked me accountprivate void delteLockedmeAccount(Scanner input) {System.out.println("press 1 to confirm account deletion");System.out.println("press 2 to go back to the main menu");System.out.println("press 3 to Logout");String userInput = input.nextLine();switch(userInput) {case "1":deleteUserFile(); //deletes file with user detailsdeleteUserDetailsFromDb();// deletes user details from main databaseSystem.out.println("Account Deleted Successfully");break;case "2":distplayUserOptions();// displays the option of tasks available to the user.break;case "3":System.out.println("Logout Successful");break;default:System.out.println("Invalid input");delteLockedmeAccount(input);}}// diaplays all the websites along with it's username and password in users databaseprivate void displayWebsiteDetails() {// pulls all data of the user from his/her data base to the hashmap "userDatabase"try {getUserDataBase();} catch (FileNotFoundException e) {e.printStackTrace();}// if user has'nt stores anything yet displays "No info available"if(userDatabase.isEmpty())System.out.println("No info avilable");else {System.out.println("----------------------------");displayWebsiteDetailsFromDatabase();//displays all the website in the user databaseSystem.out.println("----------------------------");}tryAgainDisplayWebsites();//gives an option to the user to run displayWebsiteDetails again}// //gives an option to the user to run displayWebsiteDetails againprivate void tryAgainDisplayWebsites(){System.out.println("------------");System.out.println("Enter 1 to display websites");System.out.println("Enter 2 to get back to main menu");System.out.println("Enter 3 to Logout");Scanner input = new Scanner(System.in);String inputvalue=input.nextLine();switch(inputvalue) {case "1":displayWebsiteDetails();break;case "2":distplayUserOptions();break;case "3":System.out.println("Log out success");return;//sbreak;default:System.out.println("Invalid Input");tryAgainDisplayWebsites();//System.out.println("--------------------");break;}}// changes password of a selected website in user's database.private void changeWebsitePassword(Scanner input) {System.out.println("Enter the website name");String website=input.nextLine();//returns if user enters an empty website nameif(website.equals("")) {System.out.println("Website name is empty");tryAgainChangeWebsitePassword(input);// gives an option to the user to run changeWebsitePassword againreturn;}// pulls all data of the user from his/her data base to the hashmap "userDatabase"try {getUserDataBase();} catch (FileNotFoundException e) {e.printStackTrace();}//returns of no website by the given input is foundif(websiteAlreadyExists(website)==false){System.out.println("Website not found");tryAgainChangeWebsitePassword(input);//gives an option to the user to run displayWebsiteDetails againreturn;}System.out.println("Enter wesite's new password");String password1=input.nextLine();// user inputs password the 1st time//returns if password field is emptyif(password1.equals("")) {System.out.println("Password cannot be empty");tryAgainChangeWebsitePassword(input);//gives an option to the user to run displayWebsiteDetails againreturn;}System.out.println("Enter website's new password once again");String password2=input.nextLine();// user inputs the password the second time.//if both the passwords entered matches password of the website is changedif(password1.equals(password2)) {changeWebsitePasswordFromDatbase(website,password2);//assigns new password to the given website.System.out.println("Website's password changed successfully");tryAgainChangeWebsitePassword(input);//gives an option to the user to run displayWebsiteDetails again}else {System.out.println("website's passwords entered doesn't match");tryAgainChangeWebsitePassword(input);//gives an option to the user to run displayWebsiteDetails again}}//gives an option to the user to run displayWebsiteDetails againprivate void tryAgainChangeWebsitePassword(Scanner input){System.out.println("------------");System.out.println("Enter 1 to change password of a website");System.out.println("Enter 2 to get back to main menu");System.out.println("Enter 3 to Logout");String inputvalue=input.nextLine();switch(inputvalue) {case "1":changeWebsitePassword(input);break;case "2":distplayUserOptions();break;case "3":System.out.println("Log out success");return;//sbreak;default:System.out.println("Invalid Input");tryAgainChangeWebsitePassword(input);//System.out.println("--------------------");break;}}// deletes a website from user's databaseprivate void deleteWebsite(Scanner input) {//user enters the webite to be deletedSystem.out.println("Enter the website name");String website=input.nextLine();//if user enters blank website name, returnaif(website.equals("")) {System.out.println("Website name is empty");tryAgainDeleteWebsite(input);// gives an option to the user to run deleteWebsite againreturn;}// pulls all data of the user from his/her data base to the hashmap "userDatabase"try {getUserDataBase();} catch (FileNotFoundException e) {e.printStackTrace();}// returns if the website does'nt exist in the user databaseif(websiteAlreadyExists(website)==false){System.out.println("Website not found");tryAgainDeleteWebsite(input); // gives an option to the user to run deleteWebsite againreturn;}deleteWebsiteFromDatbase(website);// detes the given wesite from the user's databaseSystem.out.println("website details have been removed from database");tryAgainDeleteWebsite(input); // gives an option to the user to run deleteWebsite again}// gives an option to the user to run deleteWebsite againprivate void tryAgainDeleteWebsite(Scanner input){System.out.println("------------");System.out.println("Enter 1 to delete a website");System.out.println("Enter 2 to get back to main menu");System.out.println("Enter 3 to Logout");String inputvalue=input.nextLine();switch(inputvalue) {case "1":deleteWebsite(input);break;case "2":distplayUserOptions();break;case "3":System.out.println("Log out success");return;//sbreak;default:System.out.println("Invalid Input");tryAgainDeleteWebsite(input);//System.out.println("--------------------");break;}}//deals with the operation of adding a new website to the user's databaseprivate void addNewWebsite(Scanner input1) {System.out.println("Enter the website name");//Scanner input1= new Scanner(System.in);String website=input1.nextLine();//returns if website entered is emptyif(website.equals("")) {System.out.println("Website name is empty");tryAgainAddNewWebsite(input1); // gives the user an option to run addNewWebsite againreturn;}//// pulls all data of the user from his/her data base to the hashmap "userDatabase"try {getUserDataBase();} catch (FileNotFoundException e) {e.printStackTrace();}//returns if website doesn'nt existif(websiteAlreadyExists(website)==true){System.out.println("Website already exists in database");tryAgainAddNewWebsite(input1);// gives the user an option to run addNewWebsite againreturn;}//gets websites username from userSystem.out.println("Enter wesite's Username");String username1=input1.nextLine();//returns if username is emptyif(username1.equals("")) {System.out.println("Username cannot be empty");tryAgainAddNewWebsite(input1);// gives the user an option to run addNewWebsite againreturn;}//gets uer entters passwordSystem.out.println("Enter wesite's password");String password1=input1.nextLine();//if user entered password is empty returns.if(password1.equals("")) {System.out.println("Password cannot be empty");tryAgainAddNewWebsite(input1);// gives the user an option to run addNewWebsite againreturn;}//gets password from user for a second timeSystem.out.println("Enter website's password once again");String password2=input1.nextLine();//if both the passwords match, website details are added to user databaseif(password1.equals(password2)) {addwebsiteDetails(website,username1,password2);// adds given website details to users databaseSystem.out.println("Website details entered sussessfully");tryAgainAddNewWebsite(input1); // gives the user an option to run addNewWebsite again}else {System.out.println("website's passwords entered doesn't match");tryAgainAddNewWebsite(input1); // gives the user an option to run addNewWebsite again}}// gives the user an option to run addNewWebsite againprivate void tryAgainAddNewWebsite(Scanner input){System.out.println("------------");System.out.println("Enter 1 to add a new website");System.out.println("Enter 2 to get back to main menu");System.out.println("Enter 3 to Logout");String inputvalue=input.nextLine();switch(inputvalue) {case "1":addNewWebsite(input);break;case "2":distplayUserOptions();break;case "3":System.out.println("Log out success");return;//sbreak;default:System.out.println("Invalid Input");tryAgainAddNewWebsite(input);//System.out.println("--------------------");break;}}// dispalys the websites in user database along with its details if availableprivate void displayWebsiteDetailsFromDatabase() {Iterator<Entry<String, String[]>> iterator = userDatabase.entrySet().iterator();while(iterator.hasNext()) {System.out.println("----------------------------");Map.Entry mapElemnt = (Map.Entry)iterator.next();System.out.println("Website: "+mapElemnt.getKey());String tempArray[]=(String[]) mapElemnt.getValue();System.out.println("Username: "+tempArray[0]);System.out.println("Password: "+tempArray[1]);System.out.println("----------------------------");}}//deletes the given website from user databaseprivate void deleteWebsiteFromDatbase(String website) {userDatabase.remove(website);overRightusersDatabase();}//changes password of the given website from the databaseprivate void changeWebsitePasswordFromDatbase(String website ,String password) {String tempValueArray[]=userDatabase.get(website);tempValueArray[1]=password;userDatabase.replace(website, tempValueArray);overRightusersDatabase();}//checks if the given website exists in dbprivate boolean websiteAlreadyExists(String website) {if(userDatabase.containsKey(website))return true;return false;}//adds the given website details to dbprivate void addwebsiteDetails(String webSite, String tempUsername, String password) {File fileObj = new File(username+".txt");FileWriter fileWriter = null;try {if(fileObj.exists()) {fileWriter = new FileWriter(fileObj,true);Scanner scannerReader = new Scanner(fileObj);if(scannerReader.hasNext())fileWriter.append("\n"+webSite);elsefileWriter.append(webSite);fileWriter.append("\n"+tempUsername);fileWriter.append("\n"+password);}else {throw new FileNotFoundException("File is not Available with name "+fileObj.getName());}}catch (IOException e) {System.out.println("An Error Occurred");//e.printStackTrace();}try {fileWriter.close();}catch (IOException e) {// TODO Auto-generated catch blocke.printStackTrace();}}//updates the user db file with new values in the hashmapprivate void overRightusersDatabase() {File fileObj = new File(username+".txt");FileWriter fileWriter = null;try {if(fileObj.exists()) {fileWriter = new FileWriter(fileObj);Iterator<Entry<String, String[]>> iterator = userDatabase.entrySet().iterator();int iterationCounter=0;while(iterator.hasNext()) {Map.Entry mapElemnt = (Map.Entry)iterator.next();if(iterationCounter!=0)fileWriter.append("\n"+(String) mapElemnt.getKey());elsefileWriter.append((String) mapElemnt.getKey());String tempArray[]=(String[]) mapElemnt.getValue();fileWriter.append("\n"+tempArray[0]);fileWriter.append("\n"+tempArray[1]);iterationCounter++;}}else {throw new FileNotFoundException("File is not Available with name "+fileObj.getName());}}catch (IOException e) {System.out.println("An Error Occurred");//e.printStackTrace();}try {fileWriter.close();}catch (IOException e) {// TODO Auto-generated catch blocke.printStackTrace();}}//gets all the website details from user db file to a hasmapprivate void getUserDataBase() throws FileNotFoundException {File fileObj = new File(username+".txt");try {if(fileObj.exists()==false)fileObj.createNewFile();Scanner scannerReader = new Scanner(fileObj);while(scannerReader.hasNextLine()) {String tempKey=scannerReader.nextLine();String usernameAndPwd[]= new String[2];usernameAndPwd[0]=scannerReader.nextLine();usernameAndPwd[1]=scannerReader.nextLine();userDatabase.put(tempKey, usernameAndPwd);}scannerReader.close();} catch (IOException e) {e.printStackTrace();}}// deletes user details from main databaseprivate void deleteUserDetailsFromDb() {getAllUserCredentionals();//fetches user name and password from database file and adds them to the hashmap//removes user key from hashmapif(allCredentials.containsKey(username))allCredentials.remove(username);setAllUserCredentionals(); //adds updated usermap to db;}//adds updated usermap to db;private void setAllUserCredentionals() {File fileObj = new File("database.txt");FileWriter fileWriter = null;try {if(fileObj.exists()) {fileWriter = new FileWriter(fileObj);Iterator<Entry<String, String>> iterator = allCredentials.entrySet().iterator();int iterationCounter=0;while(iterator.hasNext()) {Map.Entry mapElemnt = (Map.Entry)iterator.next();if(iterationCounter!=0)fileWriter.append("\n"+(String) mapElemnt.getKey());elsefileWriter.append((String) mapElemnt.getKey());fileWriter.append("\n"+(String) mapElemnt.getValue());iterationCounter++;}}else {throw new FileNotFoundException("File is not Available with name "+fileObj.getName());}}catch (IOException e) {System.out.println("An Error Occurred");//e.printStackTrace();}try {fileWriter.close();}catch (IOException e) {// TODO Auto-generated catch blocke.printStackTrace();}}//deletes file with user detailsprivate void deleteUserFile() {File fileObj = new File(username+".txt");if(fileObj.exists()==true)fileObj.delete();}//fetches user name and password from database file and adds them to the hashmapprivate void getAllUserCredentionals() {File fileObj = new File("database.txt");Scanner scannerReader;try {if(fileObj.exists()==false)fileObj.createNewFile();scannerReader = new Scanner(fileObj);int lineCounter=0;String tempUser= new String("");while(scannerReader.hasNextLine()) {if(lineCounter%2==0)tempUser=scannerReader.nextLine();else if(lineCounter%2!=0)allCredentials.put(tempUser, scannerReader.nextLine());lineCounter++;}scannerReader.close();}catch (FileNotFoundException e) {e.printStackTrace();} catch (IOException e) {// TODO Auto-generated catch blocke.printStackTrace();}}} |