IT PAT: PHASE 1

EN//KRYPT

password management software ©

Muaaz Bayat

Curro Heritage House

1.1 Problem Summary

The application being developed would be a password management programme. There are many users on the internet who have a tendency to forget passwords. They therefore use the same password for many sites. This is a bad practice and could be fatal as databreaches happen all the time. Enkrypt allows these users to create strong unique individual passwords for each site without them having to remember them. It stores these entries in a database. Enkrypt displays these passwords to the users when they sign in through their single masterpassword. Enkrypt is a secure way to store users passwords as the data in the local MS Access database is encrypted.

1.2 Motivation and research

Like most applications, there are many alternative versions available (most of which are commercially developed). I personally use one such password management solution called LastPass ™ which is developed by Log Me In. Google also offers a built in password management solution that stores your passwords and presents them to you when you need it. A brief investigation of each is presented below:

1. LastPass ™ on https://www.lastpass.com

   A license or subscription to the software will allow you to use the android or IOs apps, the Windows or Mac applications and the Chrome extension. The application stores passwords along with credit card information and secure notes in their virtual vault which is 256-bit encrypted.

   (Why Should I Use LastPass | CalNet - Identity and Access Management, 2021)

2. Google Password manager on https://passwords.google.com/

   The password manager is offered as part of your gmail account. It is chrome based and stores saved passwords on their encrypted databases. The manager is built into chrome. It also allows you to generate secure passwords.

   (Google Password Manager: What is it and how to use it, 2021)

The problem I've found with both these programmes is that they can be compromised. It is easy for your google account to be compromised an thus access to all of your passwords. LastPass ™ is more secure however it is not local and is still susceptible to phishing attacks and pharming attacks.

Motivation:

I wanted to create a secure way of creating and storing my own passwords on my local machine. The application is more secure than entering them into a spreadsheet as the data in the local MS Access database is encrypted.

1.3 Specifications of Program Function

- Log In screen is displayed
- Input username and password
  - If new user : select Sign Up - redirected to sign up screen
  - Users details authenticated by querying and checking the database
  - If incorrect : display error message
  - If correct : home screen will be displayed
- If user selected sign up
  - Username, Password will be input
  - feature to test the strength of the password will be available
    - password will be ranked based on criteria
    - visual representation of the strength will be available
  - field values will be stored in database
  - once password match and other validations are done: home screen will be displayed
- User can view the home screen:
  - A table will display all of the password entries (sitename, username, password)
    - if there are none, the table will be blank
  - Users will be able to :
    - Create new password entries by selecting (create)
      - (the create screen will open up)

        Enter the websites name into the "sitename" field

        Enter the username into the "username" field

        Optional: Enter password into the "password" field

        Aleternatively: use the "generate" button and one will be auto-generated based on selected parameters (Caps, Numbers, Special Characters)
    - Delete already created password entries
      - search the database for the entry in the sitename field

        - entryfields will be populated
        - select delet button to delete the entry
    - Edit already created password entries
        - search the database for the entry in the sitename field
        - entryfields will be populated
        - update the fields with new data
        - select edit button to update the entry
- Users will be also able to :
    - access their account information and update it
    - access help and information as to how each of the functions work, as wekk as a walkthrough of each.

(Help icon will be available in the bottom of the screen at this menu as well as all other screens with detailed instructions and guides)

- Exit : all screens will be terminated

1.4 Specifications of User Interface

Readibility

❏ no spelling or grammar errors
❏ screen must not be too cluttered and easy to read
❏ instructions concise and easy to understand

Theme

Enkrypt		
Part	Colour	Hex Code
Accent 1	Green	#7ED957
Accent 2	Blue	#C9E265
Accent 3	Yellow	#FFDE59
Accent 4	Orange	FF971C
Accent 5	Red	#FF1616
Background		
Background	Black	#000000
Menu Buttons		
Selected Outline	Blue	#0078D7
Fill	White	FFFFFF
TEXT (ARIAL - CAPS)	Black	000000
General Aspects of Design		

PRIMARY TEXT (ARIAL-CAPS)	White	FFFFFF
Error Text (Arial lowercase)	Red	#FF1616

Navigation

- ❑ Home, exit and help buttons on all screens
- ❑ Text fields for input and labels for output where necessary

Layout

- ❑ consistent to the theme
- ❑ screens will be centred

Consistency

- ❑ Error labels below all input fields
- ❑ icon for the help screen will remain in the same position between all screens

1.5 Specifications of the Help Function

How it can be accessed

Users will be able to access the help function from any of the screens in the programme.

Each of these help functions will be able to be accessed via the help menu on the main screen as well as the respective funtions being linked to their respective pages.

Types of help

Help screens

The help screen will contain **simple to follow instructions** for the following categories:

- → Walkthrough of the respective screen
- → a step by step procedural guide to how users can interact with the programme
- → How the programme works
- → detailed breakdown of the owrkign of the programme
- → Contact support
- → details to support personal
- → Sumbit bugs and feature requests
- → details to submit requests to

Contextual help

Contextual help will be available when the user selects the help button to the respective screen. The help referred to in these buttons will be specific to that screen and functions thereof.

1.6 Specifications of Permanent data storage

Users:

Fields

Field	Variable Type
Name (stores users name)	String
Surname (stores users surname)	String
Username (stores users unique username)	String
Password (stores users master password)	String

When data is stored

Data is stored while the programme is running

When data is accessed

All userfields wil be loaded into RAM when the main screen is loaded

When data is updated

- ❏ Whenever a user changes their attributes in the account settings GUI

Entries:

Fields

Field	Variable Type
Sitename (stores entries sitename)	String
email (stores users signed in email)	String
Username (stores entries unique username)	String

Password (stores entries password)	String

When data is stored

Data is stored while the programme is running

When data is accessed

Entries will be loaded into RAM when the display table in the Home GUI is reloaded.

When data is updated

- ❑ Whenever a user updates, deletes or creates entries

Help:

Fields

For each GUI and unique function the help will be stored in a final string

When data is stored

when the system is created

When data is accessed

All help will be loaded into RAM when the help screen is loaded

When data is updated

- ❑ when backend updates help

1.7 System Requirements:

Programmer Requirements

Hardware

- ❏ Processor: 1 gigahertz (GHz) or faster processor
- ❏ RAM:Minimum 4GB
- ❏ Hard disk space:Minimum 30GB

Software

- ❏ Windows 10 operating system
- ❏ Java 8 or higher
- ❏ NetBeans 12
- ❏ MSAccess 365

User Requirements

Hardware

- ❏ Processor: 1 gigahertz (GHz) or faster processor
- ❏ RAM: Minimum 3 GB
- ❏ Hard disk space: Minimum 25 GB

Software

- ❏ Windows 10 operating system
- ❏ Java 8 or higher
- ❏ MS Access 365

Bibliography:

Calnetweb.berkeley.edu. 2021. *Why Should I Use LastPass | CalNet - Identity and Access Management*. [online] Available at: <https://calnetweb.berkeley.edu/calnet-me/lastpass-premium/why-should-i-use-lastpass> [Accessed 3 May 2021].

TechRadar. 2021. *Google Password Manager: What is it and how to use it*. [online] Available at: <https://www.techradar.com/news/google-password-manager-what-is-it-and-how-to-use-it#:~:text=Google%20Password%20Manager%20is%20a,the%20Google%20Chrome%20web%20browser.&text=When%20you%20revisit%20a%20site,you%20use%20across%20the%20internet.> [Accessed 5 May 2021].

IT PAT: PHASE 2

EN//KRYPT

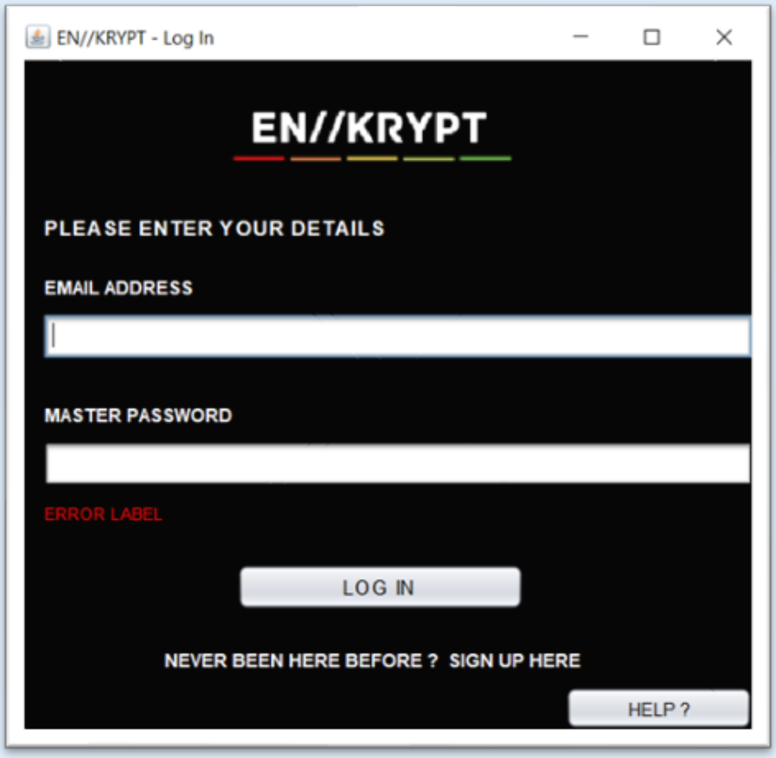
password management software ©

Muaaz Bayat

Curro Heritage House

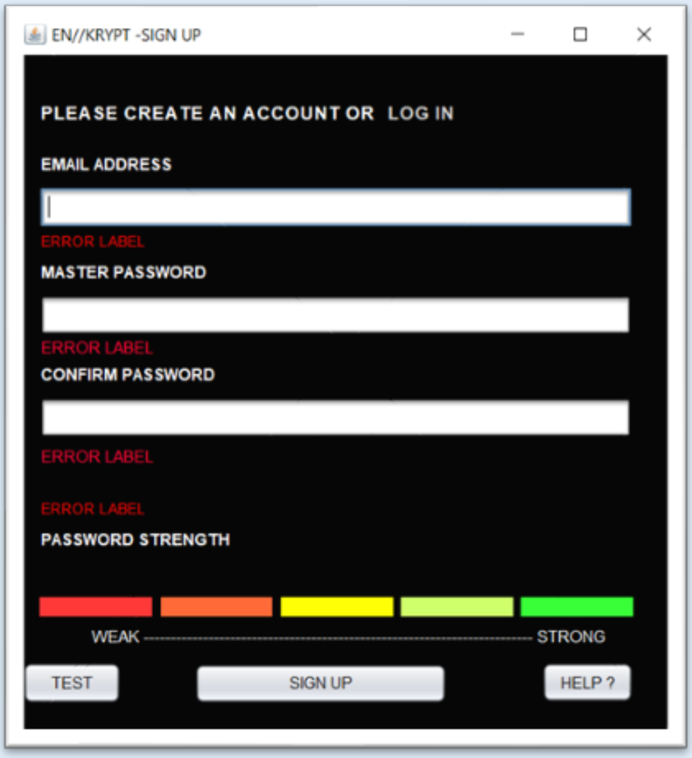
2.1 INERFACE DESIGN

LOG IN GUI



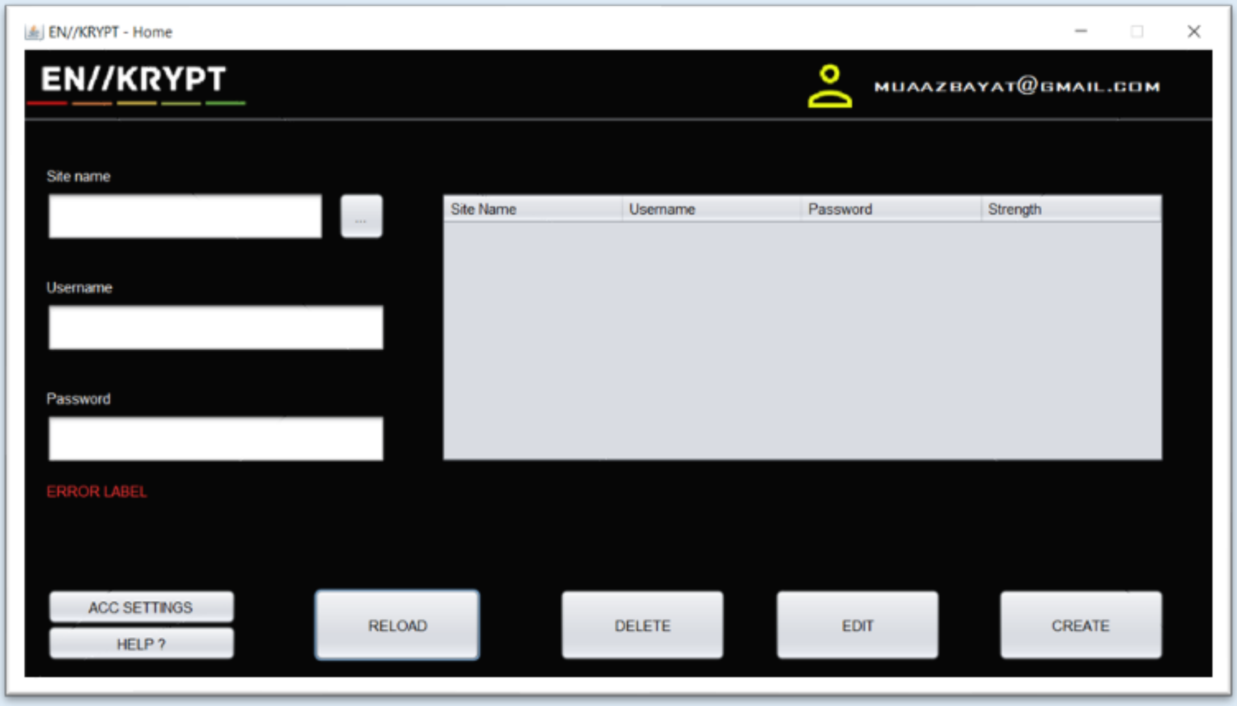
Description	This page will welcome the user. If they are registered on the database, they can login or go to the Sign up page where they can enter their details
Security Group	All Users
Data	Registered users data will be stored in a database Data for fields are temporarily stored in variables
Actions	Buttons: → Sign Up – Opens the sign up page → Help – Opens the help page → Log In – Verifies the Users details and continues to the main screen Text Fields: → Email Address– username will be entered → Master Password – password will be entered

SIGN UP GUI



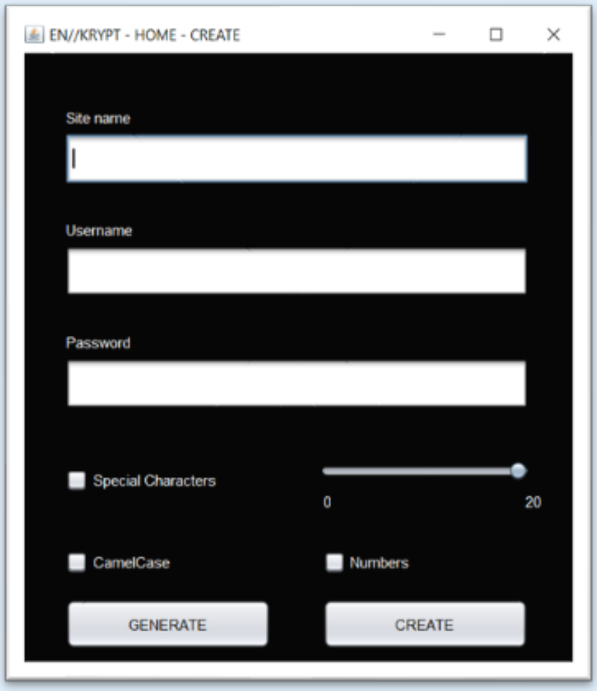
Description	This page allows users to enter their details and be registered on the database.
Security Group	All Users
Data	Users data will be stored in a database Data for fields are temporarily stored in variables
Actions	Buttons: → Sign Up – Registers the users details in the database → Help – Opens the help page → Test – Indicates the strength of the password → Log In Text– Opens the login page Text Fields: → Email Address – Email address will be entered → Master Password – password will be entered → Confirm Password – password will be re-entered

HOME GUI



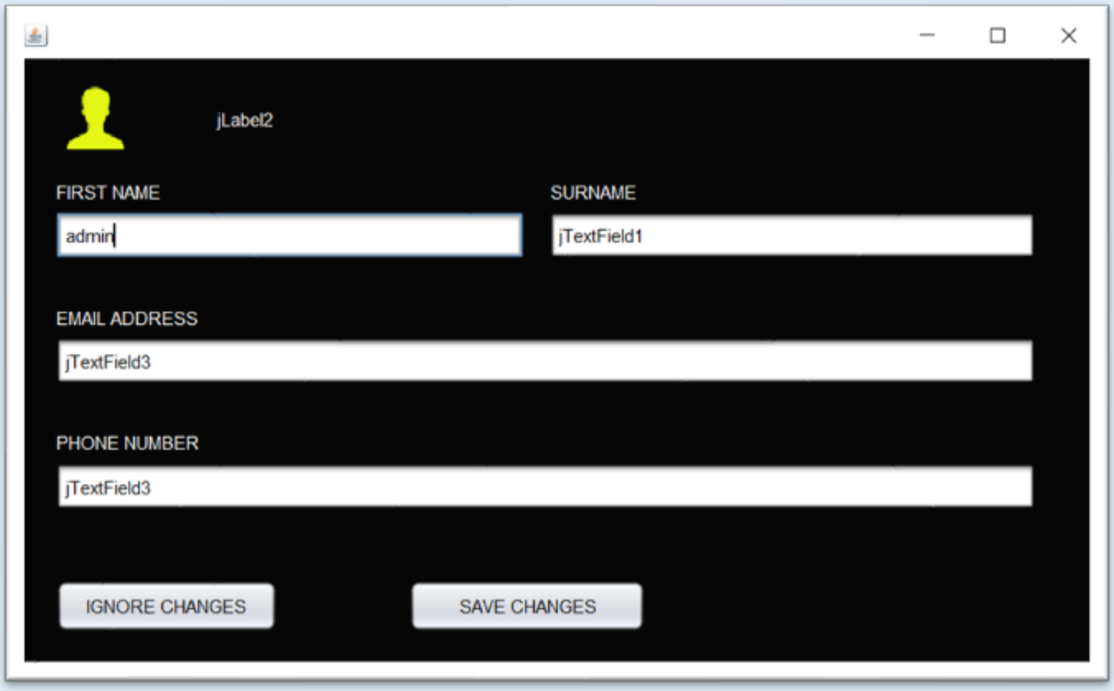
Description	Users can view their password entries, make changes to them and add new ones Button to view their account settings
Security Group	Authenticated Users
Data	Users Email will be stored in the database, and methods will be called to set the header to display it
Actions	Buttons: → Acc Settings – Opens the acc settings GUI → Reload – Refreshes the Table of entries → Delete – Deletes the desired entry → Edit – Saves changes to the desired entry → Help – Opens the help page → Create – Opens the create GUI → Search – populates the username and passwords fields Table: → Entry Table – displays all of the entries stored for the user Text Fields: → Site name – site name of desired entry will be entered → Username – username of entry will be populated and edited → Password – password entry will be populated and edited

CREATE GUI



Description	Users can create new password entries, automatically generate passwords based on selected attributes.
Security Group	Authenticated Users
Data	Users Email will be stored in the database, and methods will be called to set the header to display it
Actions	Buttons: → Generate – creates a password to the selected length based on attributes → Create – inserts the password entry into the database Text Fields: → Site name – site name will be entered → Username – username will be entered → Password – password will be entered Check Boxes: → Special Characters – adds special characters to password generator → Camel Case – adds capital letters to password generator → Numbers – adds numbers to password generator J Slider: → Selects the maximum length of password to be generated

ACC SETTINGS GUI



FIRST NAME: admin

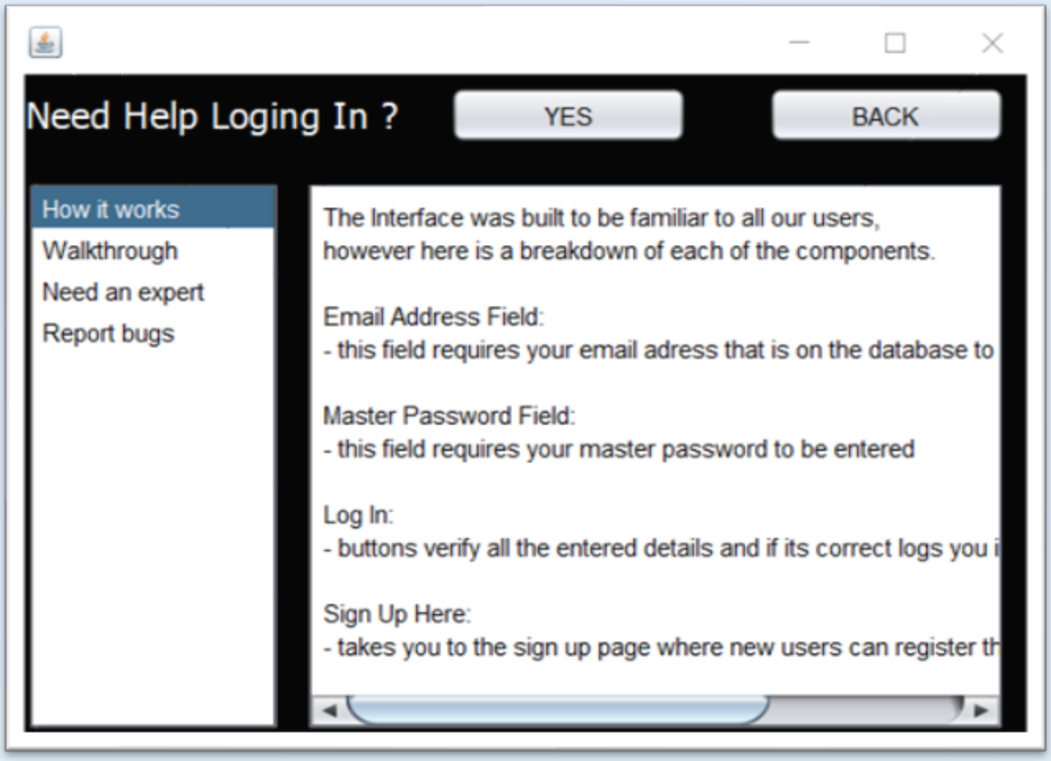
SURNAME: jTextField1

EMAIL ADDRESS: jTextField3

PHONE NUMBER: jTextField3

Description	Users can update and edit their data stored in the database.
Security Group	Authenticated Users
Data	Users details will be stored in the database, and methods will be called to set the fields to display it
Actions	Buttons: → Ignore Changes – exits the sub-GUI without doing anything → Save Changes – updates the database and saves changes Text Fields: → Name – name will be entered → Surname – surname will be entered → Email – email will be populated → Phone number – phone number will be entered

HELP (Log In) GUI



The Interface was built to be familiar to all our users, however here is a breakdown of each of the components.

Email Address Field:
- this field requires your email adress that is on the database to

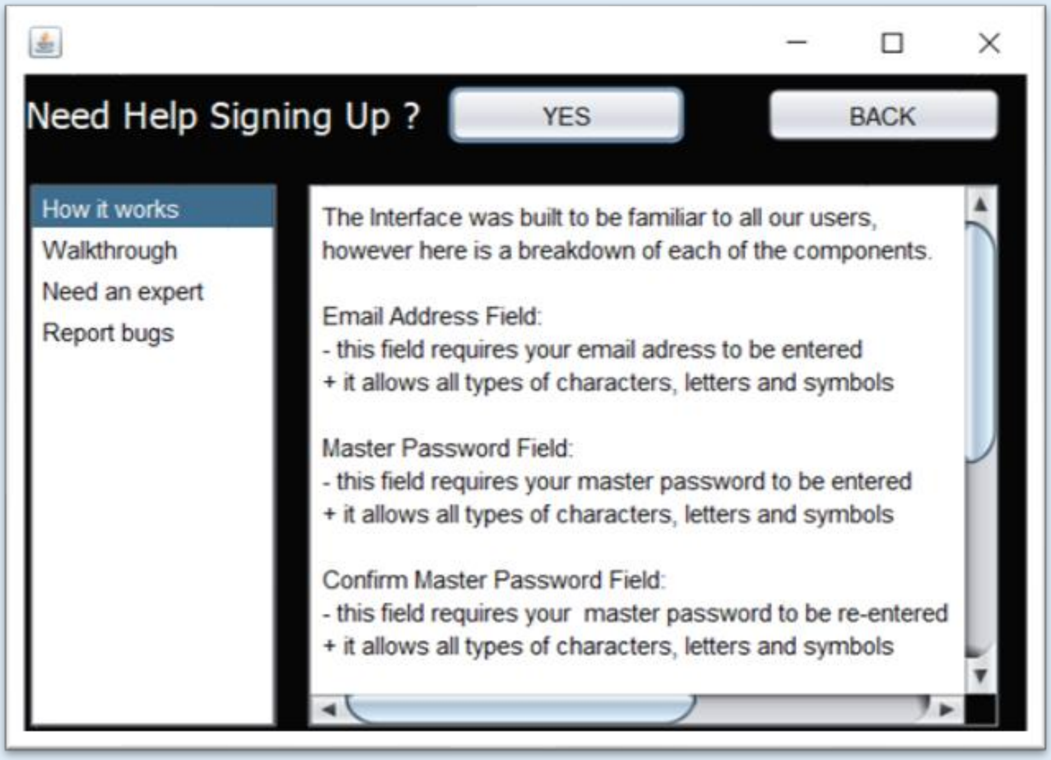
Master Password Field:
- this field requires your master password to be entered

Log In:
- buttons verify all the entered details and if its correct logs you i

Sign Up Here:
- takes you to the sign up page where new users can register th

Description	This page will show users a side panel of help options for the Log In screen and the respective solutions and guides.
Security Group	All Users
Data	All help data will be stored in Strings that will only be editable in the backend of the programme
Actions	Buttons: → Back – closes the help GUI → Yes – displays the respective help solution based on side menu selection Menu : → Displays a list of help options for the log in GUI Text Area : → Displays the solution and guide to the respective help option selected

HELP (Sign Up) GUI



The Interface was built to be familiar to all our users, however here is a breakdown of each of the components.

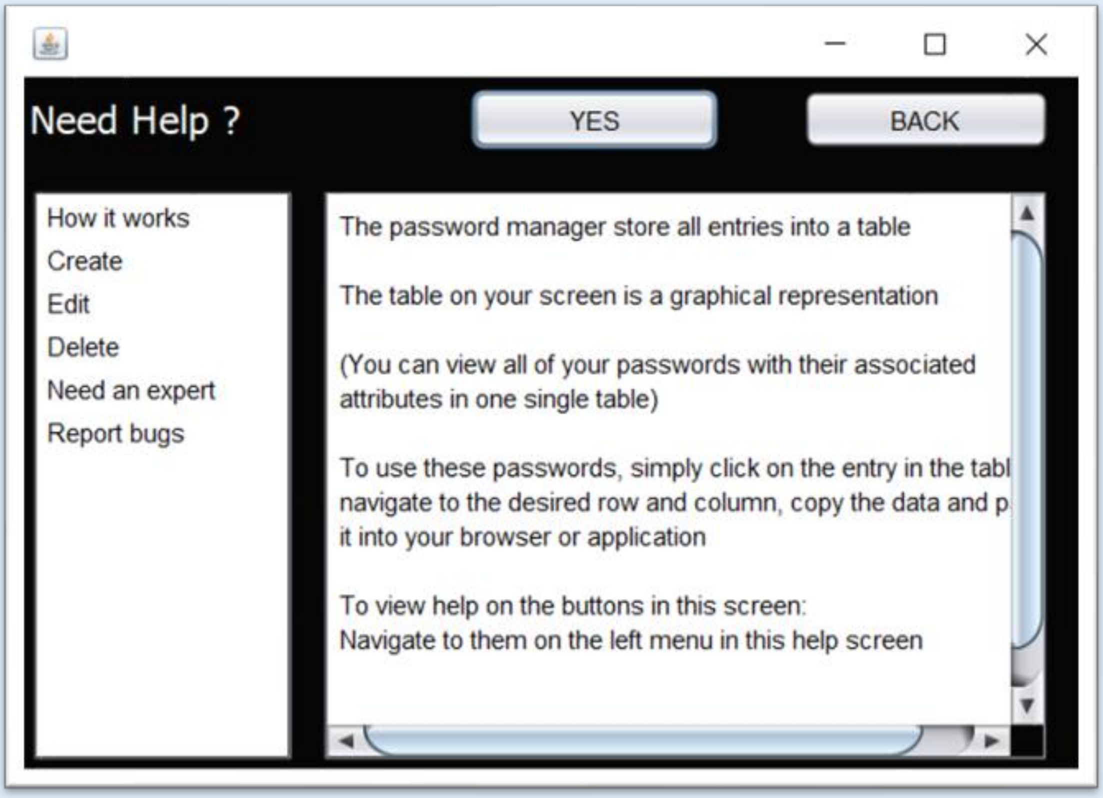
Email Address Field:
- this field requires your email adress to be entered
+ it allows all types of characters, letters and symbols

Master Password Field:
- this field requires your master password to be entered
+ it allows all types of characters, letters and symbols

Confirm Master Password Field:
- this field requires your master password to be re-entered
+ it allows all types of characters, letters and symbols

Description	This page will show users a side panel of help options for the Sign Up screen and the respective solutions and guides.
Security Group	All Users
Data	All help data will be stored in Strings that will only be editable in the backend of the programme
Actions	Buttons: → Back –closes Help GUI → Yes – displays the respective help solution based on side menu selection Menu : → Displays a list of help options for the Sign Up GUI Text Area : → Displays the solution and guide to the respective help option selected

HELP (Home) GUI



The password manager store all entries into a table

The table on your screen is a graphical representation

(You can view all of your passwords with their associated attributes in one single table)

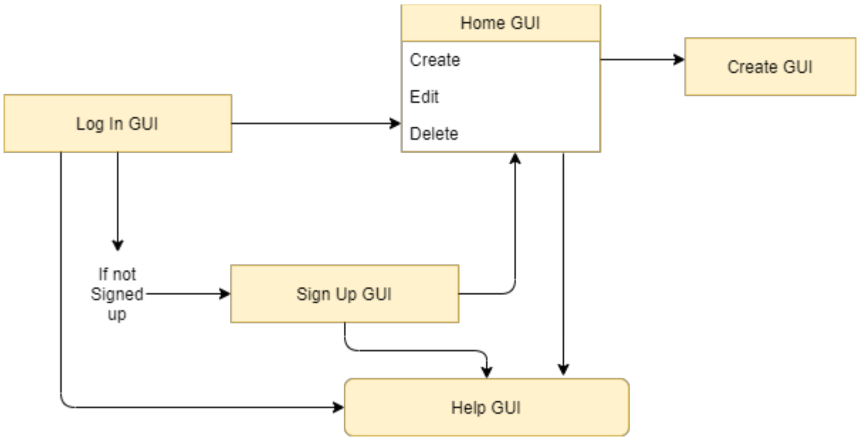
To use these passwords, simply click on the entry in the tabl navigate to the desired row and column, copy the data and p it into your browser or application

To view help on the buttons in this screen:
Navigate to them on the left menu in this help screen

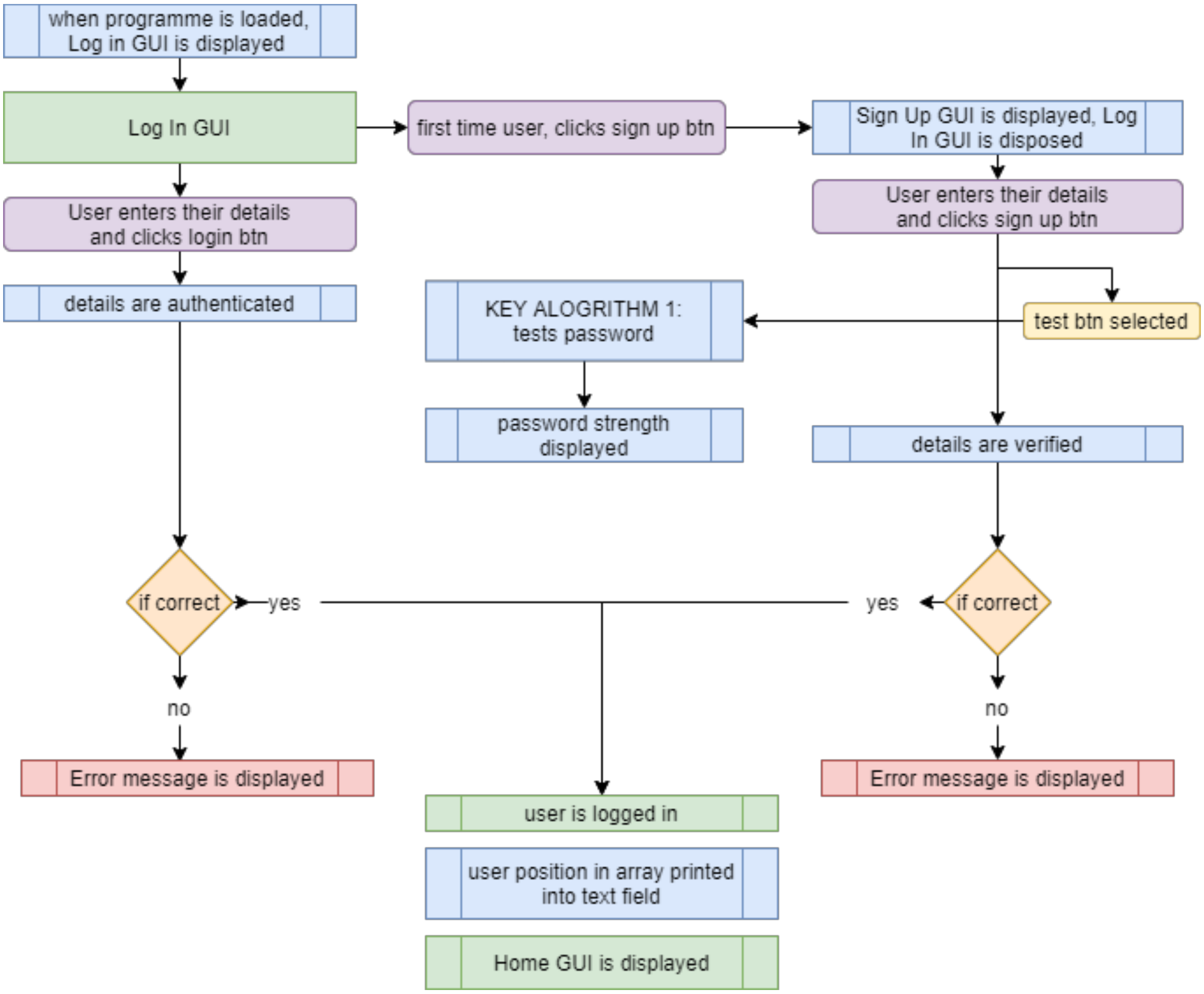
Description	This page will show users a side panel of help options for the Home screen and the respective solutions and guides.
Security Group	All Users
Data	All help data will be stored in Strings that will only be editable in the backend of the programme
Actions	Buttons: → Back –closes Help GUI → Yes – displays the respective help solution based on side menu selection Menu : → Displays a list of help options for the Home GUI Text Area : → Displays the solution and guide to the respective help option selected

2.2 Programme Flow

The diagram below is a broad overview of the *user flow of the programme*

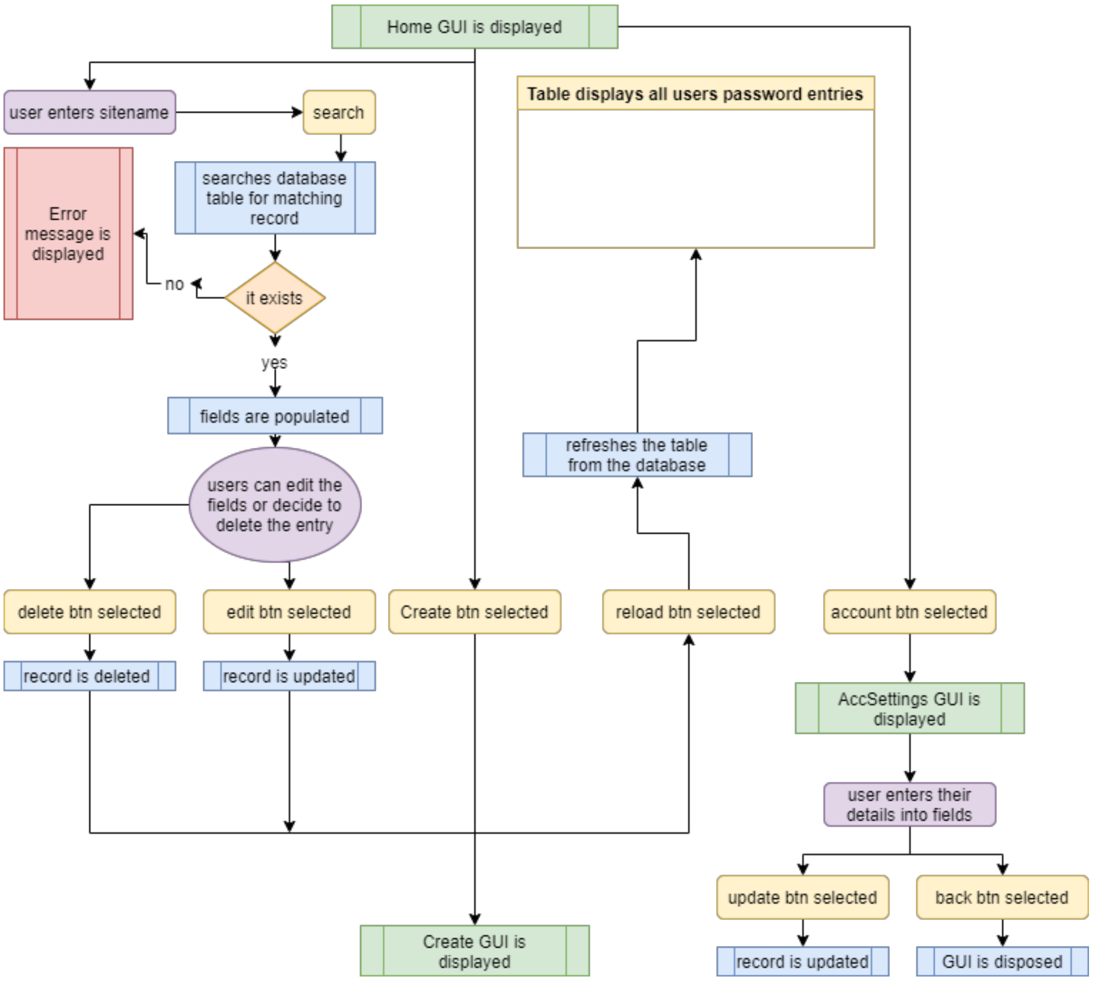


The diagram below shows a more detailed overview of the programme flow for the LogIn/SignUp phase

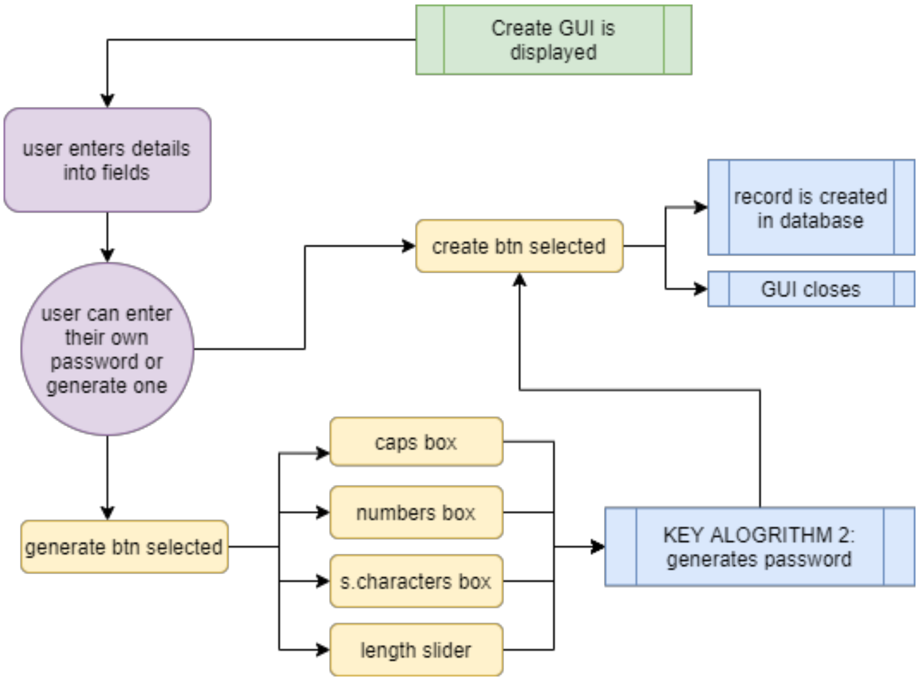


when programme is loaded, Log in GUI is displayed
Log In GUI
first time user, clicks sign up btn
Sign Up GUI is displayed, Log In GUI is disposed
User enters their details and clicks login btn
User enters their details and clicks sign up btn
details are authenticated
KEY ALOGRITHM 1: tests password
test btn selected
password strength displayed
details are verified
if correct
yes
yes
if correct
no
no
Error message is displayed
Error message is displayed
user is logged in
user position in array printed into text field
Home GUI is displayed

The diagram below shows a more detailed overview of the programme flow for the Home Interaction phase



The diagram below shows a more detailed overview of the programme flow for the create GUI


Create GUI is displayed
user enters details into fields
record is created in database
create btn selected
GUI closes
user can enter their own password or generate one
caps box
numbers box
KEY ALOGRITHM 2: generates password
generate btn selected
s.characters box
length slider

2.3 Class Design

Ucanaccess	
	Description
- Connection connection - PreparedStatement statement - ResultSet resultSet	Sets up connection to database Stores a SQL statement Stores database

+ ucanaccess()
+ ResultSet ExQuery(String stmt)
+ UpdateTable(String stmt)
	Connects to the database via a driver
	Execute Query method
	Update table method

User	
	Description
- String email - String masterpassword - String level	Holds user's email Holds user's master password Holds user's registered level
+ constructor(email: String, password: String, level:String) + getEmail() : string + toString() : string	Object parametised constructor Returns the objects email Converts the object to a string

CompleteUser	
	Description
- String phone number - String name - String surname	Holds user's phone number Holds user's name Holds user's surname
+ constructor(phoneNumber : String, name : String, surname : String) + getName() : string + getSurname () : string + getPhoneNumber() : string + toString() : string	Parametised constructor inheriting user Returns the objects name Returns the objects surname Returns the objects phonenumber Converts the object to a string

UserFunctions	
	Description
- int userpos	Stores the position of the user in the userArr

+ authenticate(inEmail: string, inMPass: string) :Boolean	verifies the users login details.
+ isFieldBlank(inStr: string) : Boolean	checks if a field is blank
+ doesUsernameExist(inStr: string) : Boolean	checks if the userEmail exists already
+ doPasswordsMatch(inStr : string) : Boolean	checks if the passwords entered match
+ verifyDetails(inEmail:string,inP1: string, inP2: string) : Boolean	performs the neccessary validations returning a flag value if they all are met
+	
+ signPos(inEmail : string)	writes the users position into the userpos.txt textfile
+ createUser(inEmail:string,inMasterPassword: string)	inserts the relevant data read in from parameters into the database
+ signUserIn(inEmail: string)	writes the users data into the userdata.txt text file

ProgrammeFunctions	
	Description
+ containsNumbers(String inStr):Boolean	checks to see if a string has numbers within it
+ containsSpecialCharacters(String inStr): Boolean	checks to see if a string has special characters
+ containsLetters(String inStr): Boolean	checks to see if a string has alphabets
+ containsCapital(String inStr): Boolean	checks to see if a string has capital letters
+ calcStrength(String inStr): int	ranks a string based on the diversity of characters
+ createEntry(String SiteName, String Username, String Password)	inserts an entry into the tblEntries table
+ updateEntryTbl(String inUsername, String inEmail, String inSiteName,String inPassword)	updates a record in the tblEntries table
+ deleteEntry(String inEmail, String inSiteName)	deletes a record in the tblEntries table
+ searchAndGetEntry(String inSiteName, String inUsername) : String	searches the database for the record with a field value and returns the record

+ getPassType(boolean inSchars, boolean inCaps, boolean inNum) : int	checks to see which attributes a generated password must have
+ generatePass(passType : int, len : int) : String	generates a password based on the type needed from the getPassType() method
+ generateAlphaCharsCaps(len : int) : String	creates a random string containing: alphabets + special characters + capitals
+ generateAlphaChars(len : int) : String	creates a random string containing: alphabets + special characters
+ generateAlphaNumCaps(len : int) : String	creates a random string containing: alphabets + numbers + capitals
+ generateAlphaNumChars(len : int): String	creates a random string containing: alphabets + numbers + special characters
+ generateAlphaNum(len : int): String	creates a random string containing:alphabets + numbers
+ generateAlphaNumCapsChars(len : int) : String	creates a random string containing:alphabets + capitals + numbers + special characters
+ generateAlphaCapsNum(len : int): String	creates a random string containing: alphabets + capitals + numbers
+ generateAlphaCaps(len : int): String	creates a random string containing:alphabets + capitals
+ generateAlpha(len : int) : String	creates a random string containing: only alphabets

2.4 Secondary Storage Design

Database Design

tblUsers			
Field	Type	Description	Example
emailaddress 🔑	Short Text	Stores the user’s emailaddress	Muaazbayat@gmail.com
masterpassword	Short Text	Stores users’ password	Mlm438jJD#*
level	Short Text	Stores the user’s registered level	complete
name	Short text	Stores the user’s first name	Muaaz
Surname	Short text	Stores the user’s last name	Bayat

tblEntries			
Field	Type	Description	Example
SiteName 🔑	Short Text	Stores the user’s emailaddress	www.enkrypt.com
Email 🔑	Short Text	Stores user’s emailaddress	muaazbayat@gmail.com
Username 🔑	Short Text	Stores the user’s registered level	Muaaz_Bayat
Password	Short text	Stores the user’s first name	Mlm438jJD#*

*three primary keys are used as a user can have different usernames for the same site

Additional Secondary Storage: Text Files

Userdata.txt	
Sample data stored	“admin~1234~incomplete~-~-~-“
Delimiter used	“~”
Max file length	1 line holding a user object in string form

Breakdown of sample data					
EmailAddress	Password	Level	PhoneNumber	Name	Surname
Admin	12340	incomplete	-	-	-

2.5 Explanation of Secondary Storage Design

A database stores the permanent information such as the User's email, password, and other fields. It can store very large numbers of records efficiently. It makes it quick and easy to find information. It is easy to add new data and to edit or delete old data. The database allows for secure encryption that cannot be easily hashed if it has to be compromised.

Enkrypt makes use of MS Access DMBS

tblUsers

tblUsers is the table in the "enkrypt.accdb" database which stores encrypted user information. It stores all the fields involved in processing user functions such as the users name, surname, email, password and level. These values can be accessed in the programme and edited and updated if the user is authenticated.

tblEntries

tblEntries is the table in the "enkrypt.accdb" database which stores encrypted user password entries. It stores all the fields involved in processing programme functions such as the sitename, username and password for the entry. These values can be accessed in the programme and edited and updated if the user is authenticated.

Enkrypt also makes use of the "userdata.txt" text file

userdata.txt

A text file is a suitable solution for holding user data once the user is signed in. This data can easily be accessed between classes very quickly. It is overwritten everytime a new user signs in. Because the data is very small and is overwritten often, the use of a text file is justified. The data in the text file is not sensitive and therefore does not have to be encrypted.

2.6 Explanation of Primary Data Structure

Primary

The classes are stored in primary storage.

Ucanaccess class

The Ucanaccess class is needed to connect the programme to the database. This class allows the other classes in the project to communicate with the “enkrypt.accdb” database. It stores the path between the database and java files. It also enables communication between Primary and Secondary storage.

User, CompleteUser, userArr classes

The User class instantiates a user object. The completeUser class inherits the attributes of the user class. The userArr class stores an array of users and complete users.

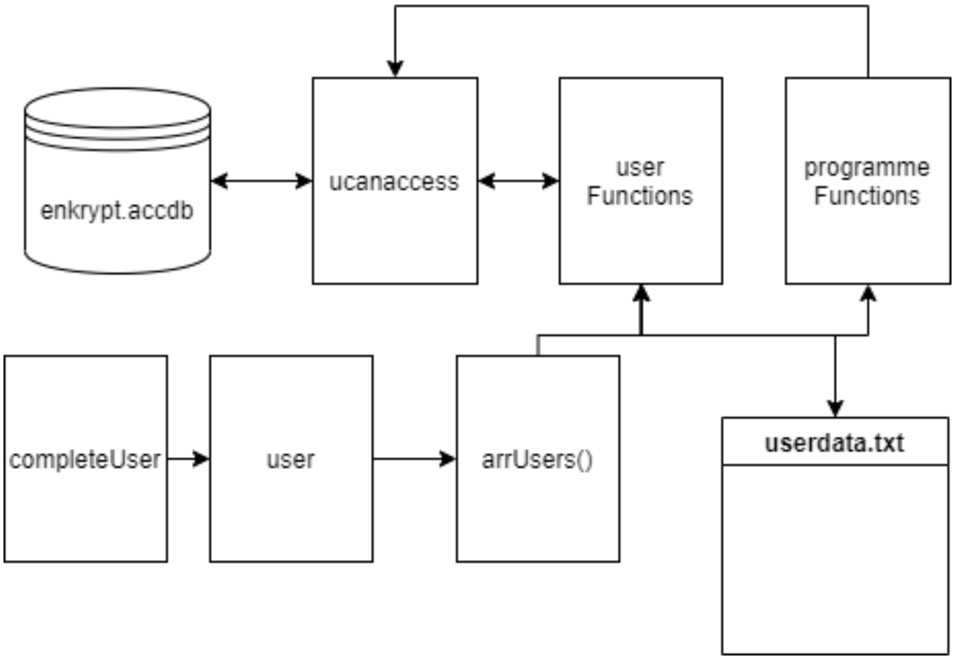
UserFunctions class

The data in this class comes in from arrUsers class. The class writes data to the userdata.txt text file as well as uploads data directly into the database via the ucanaccess class.

ProgrammeFunctions class

The data in this class comes in from the database. The class uploads data directly into the database via the ucanaccess class.

This diagram shows an overview of the communication between classes and the primary data structure link.



IT PAT: PHASE 4

EN//KRYPT

password management software ©

Muaaz Bayat

Curro Heritage House

4.1.1 Externally Sourced Code

No external code was used in this project except for syntax and the use of the random library.

```
public String generateAlpha(int len)
   {
      int leftlimit = 97;
      int rightlimit = 122;
      int targetStringLenght = len;
      Random random = new Random();
      String generatedString = random.ints(leftlimit, rightlimit + 1)
            .limit(targetStringLenght).collect(StringBuilder::new, StringBuilder::appendCodePoint, StringBuilder::append)
            .toString();
      return generatedString;
   }
```

The code was adapted to the programme from : <https://www.baeldung.com/java-random-string>

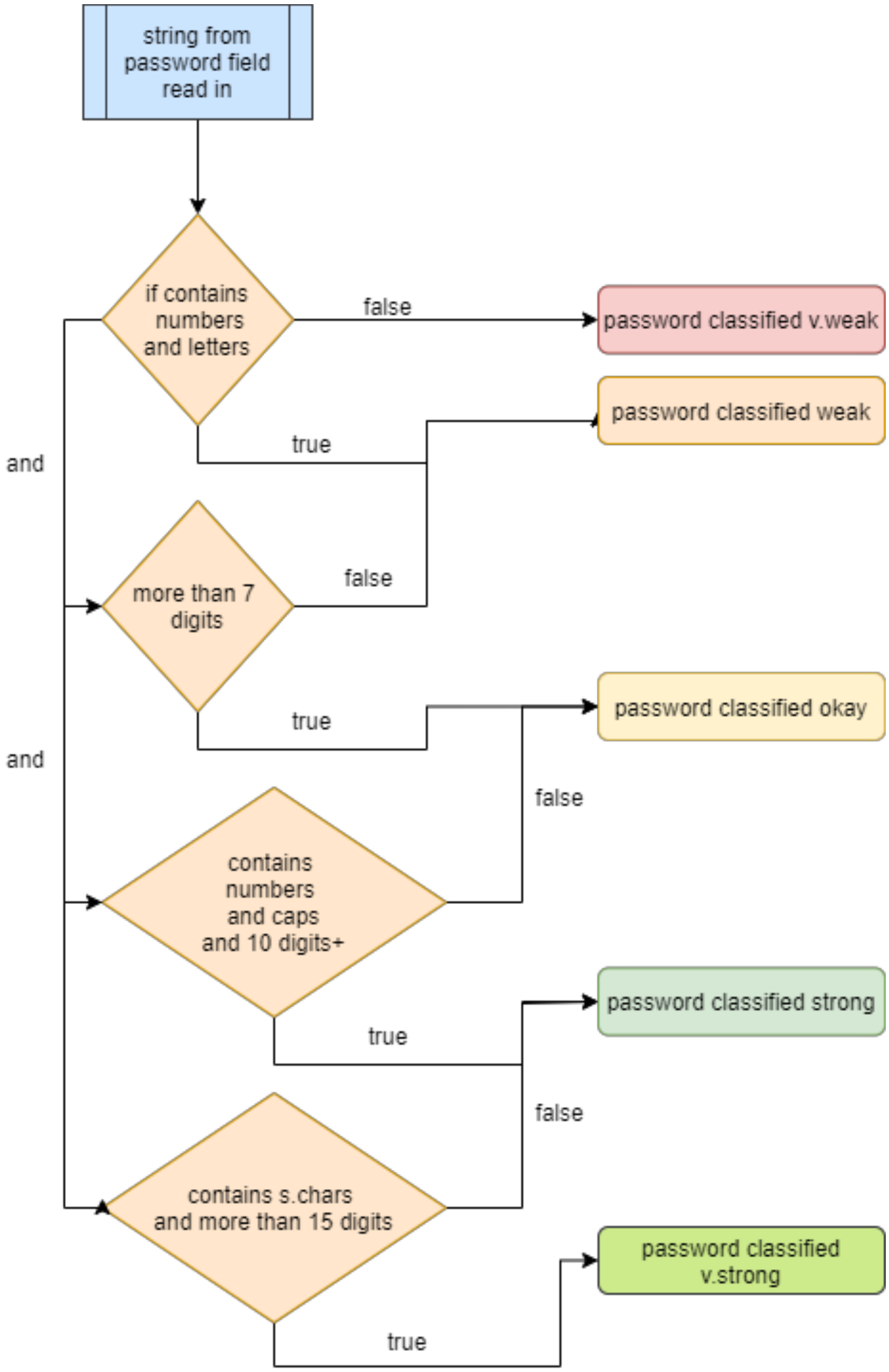
4.1.2 Explanation of Critical Algorithms

Enkcrypt has two critical algorithms:

1. Password Ranking Algorithm

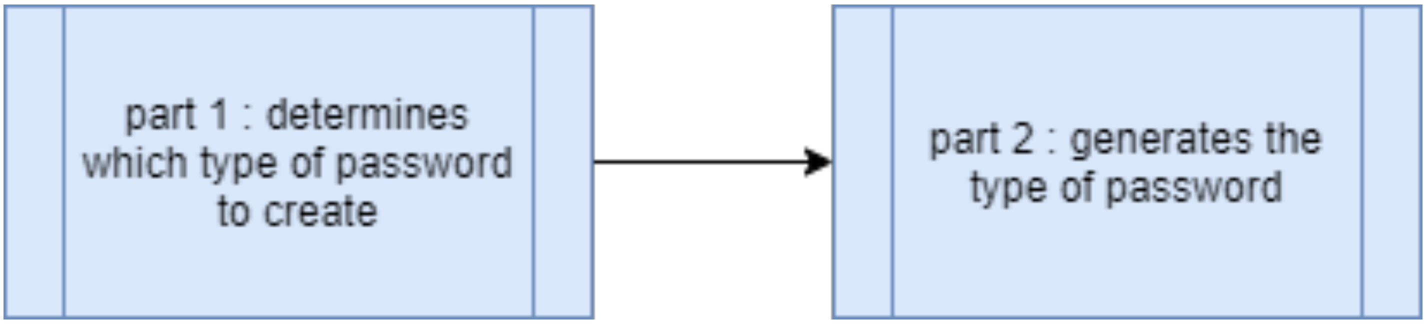
2. Password Generating Algorithm

The password ranking algorithm is as follows:

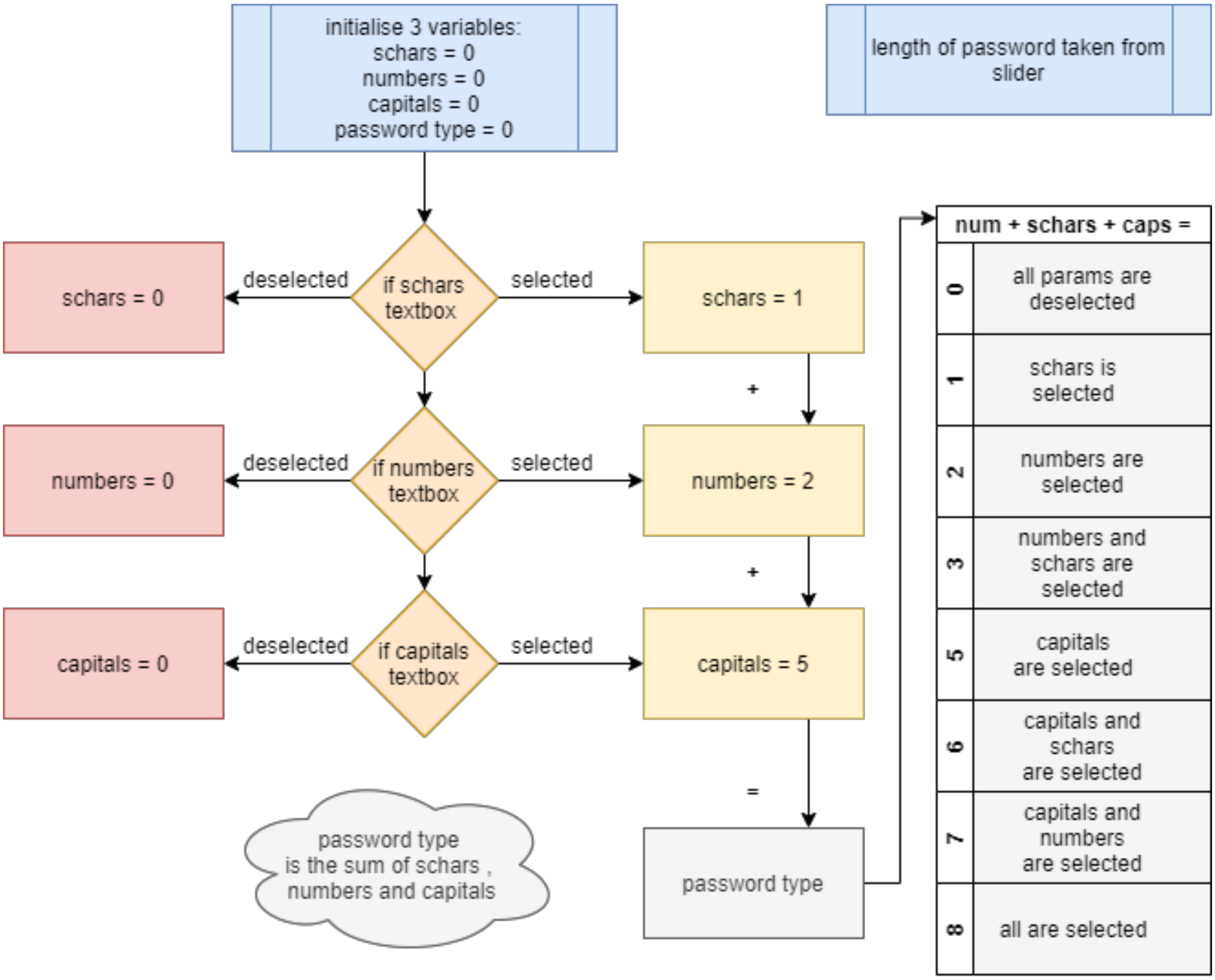

string from password field read in
if contains numbers and letters
false
password classified v.weak
true
password classified weak
and
more than 7 digits
false
true
password classified okay
and
false
contains numbers and caps and 10 digits+
true
password classified strong
false
contains s.chars and more than 15 digits
password classified v.strong
true

1.

Overview of Password Generator Algortihm:

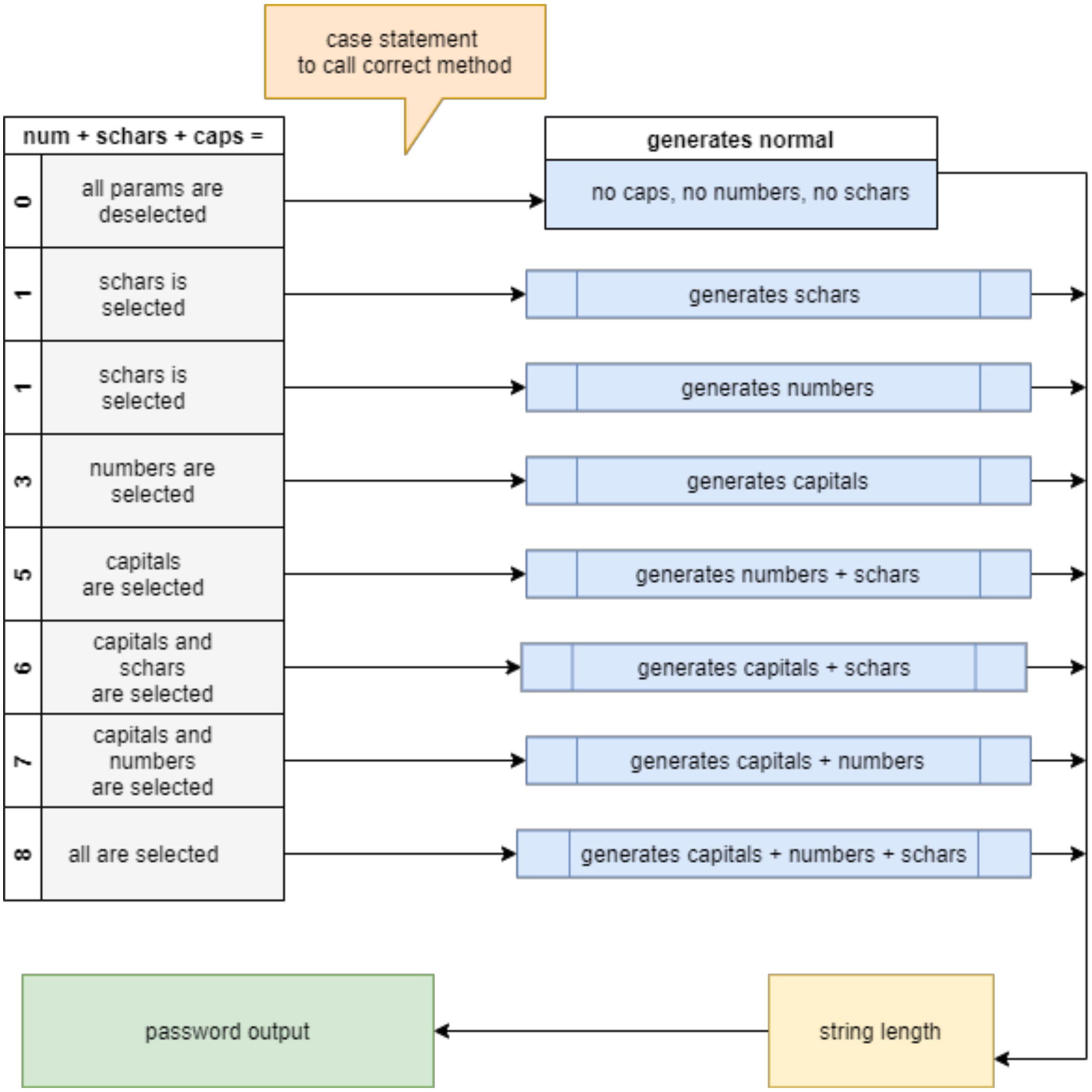

part 1 : determines which type of password to create
part 2 : generates the type of password

Part 1 :



num + schars + caps =	
0	all params are deselected
1	schars is selected
2	numbers are selected
3	numbers and schars are selected
5	capitals are selected
6	capitals and schars are selected
7	capitals and numbers are selected
8	all are selected

Part 2:


case statement
to call correct method
num + schars + caps =
0
all params are deselected
1
schars is selected
1
schars is selected
3
numbers are selected
5
capitals are selected
6
capitals and schars are selected
7
capitals and numbers are selected
8
all are selected
generates normal
no caps, no numbers, no schars
generates schars
generates numbers
generates capitals
generates numbers + schars
generates capitals + schars
generates capitals + numbers
generates capitals + numbers + schars
string length
password output

4.1.3 Advanced Techniques

Encrypt makes use of:

Encryption

The data written to and derived from the database is encrypted and decrypted respectively using an encryption algorithm.

The algorithm is as follows:

Array of (inherited) Objects

The complete user object inherits the attributes of the user object. The arrUser class holds these objects in an array of objects.

Contextual Help

The help in the programme is specific to each function. The help is relevant to the respective GUI and buttons thereof.

4.2.1 Evaluation of solution

Original Problem: The application being developed would be a password management programme. There are many users on the internet who have a tendency to forget passwords. They therefore use the same password for many sites. This is a bad practice and could be fatal as databreaches happen all the time. Enkrypt allows these users to create strong unique individual passwords for each site without them having to remember them. O It stores these entries in a database. Enkrypt displays these passwords to the users when they sign in through their single masterpassword. O Enkrypt is a secure way to store users passwords as the data in the local MS Access database is encrypted. O

The three circles denote the core functions of the programme.

After comprehensive analysis, it is concluded that the core goals of the programme are met however, there are areas and features for improvement namely:

- The UI could be more user intuitive (the search function)
  - Perhaps allowing the user to click on the jtable and edit directly with the programme saving the changes is a more intuitive solution
- Integrations to the respective websites could be built in
  - Clicking on the website link could take users directly to the login page of the website

4.2.2 Functional Testing

	14/09/2021	17/09/2021
	Muaaz Bayat	Hamzah Bayat
Log In screen is displayed	Y	Y
Input username and password	Y	Y
If new user : select Sign Up - redirected to sign up screen	Y	Y
Users details authenticated by querying and checking the database	Y	Y
If incorrect : display error message	Y	Y
If correct : home screen will be displayed	Y	Y
If user selected sign up	Y	Y
Username, Password will be input	Y	Y
feature to test the strength of the password will be available	Y	Y
password will be ranked based on criteria	Y	Y

visual representation of the strength will be available	Y	Y
field values will be stored in database	Y	Y
once password match and other validations are done: home screen will be displayed	Y	Y
User can view the home screen:	Y	Y
A table will display all of the password entries (sitename, username, password)	Y	Y
if there are none, the table will be blank	Y	Y
Users will be able to :	Y	Y
Create new password entries by selecting (create)	Y	Y
(the create screen will open up)	Y	Y
Enter the websites name into the "sitename" field	Y	Y
Enter the username into the "username" field	Y	Y
Optional: Enter password into the "password" field	Y	Y
Aleternatively: use the "generate" button and one will be auto-generated based on selected parameters (Caps, Numbers, Special Characters)	Y	Y
Delete already created password entries	Y	Y
search the database for the entry in the sitename field	Y	Y
entryfields will be populated	Y	Y
select delet button to delete the entry	Y	Y
Edit already created password entries	Y	Y
search the database for the entry in the sitename field	Y	Y
Log In screen is displayed	Y	Y
Input username and password	Y	Y
If new user : select Sign Up - redirected to sign up screen	Y	Y
Users details authenticated by querying and checking the database	Y	Y
If incorrect : display error message	Y	Y
If correct : home screen will be displayed	Y	Y
If user selected sign up	Y	Y

Username, Password will be input	Y	Y
feature to test the strength of the password will be available	Y	Y
password will be ranked based on criteria	Y	Y
visual representation of the strength will be available	Y	Y
field values will be stored in database	Y	Y
once password match and other validations are done: home screen will be displayed	Y	Y
User can view the home screen:	Y	Y
A table will display all of the password entries (sitename, username, password)	Y	Y
if there are none, the table will be blank	Y	Y
Users will be able to :	Y	Y
Create new password entries by selecting (create)	Y	Y
(the create screen will open up)	Y	Y
Enter the websites name into the "sitename" field	Y	Y
Enter the username into the "username" field	Y	Y
Optional: Enter password into the "password" field	Y	Y
Aleternatively: use the "generate" button and one will be auto-generated based on selected parameters (Caps, Numbers, Special Characters)	Y	Y
Delete already created password entries	Y	Y
search the database for the entry in the sitename field	Y	Y
entryfields will be populated	Y	Y
select delete button to delete the entry	Y	Y
Edit already created password entries	Y	Y
search the database for the entry in the sitename field	Y	Y
Delete already created password entries	Y	Y
search the database for the entry in the sitename field	Y	Y
entryfields will be populated	Y	Y
update the fields with new data	Y	Y

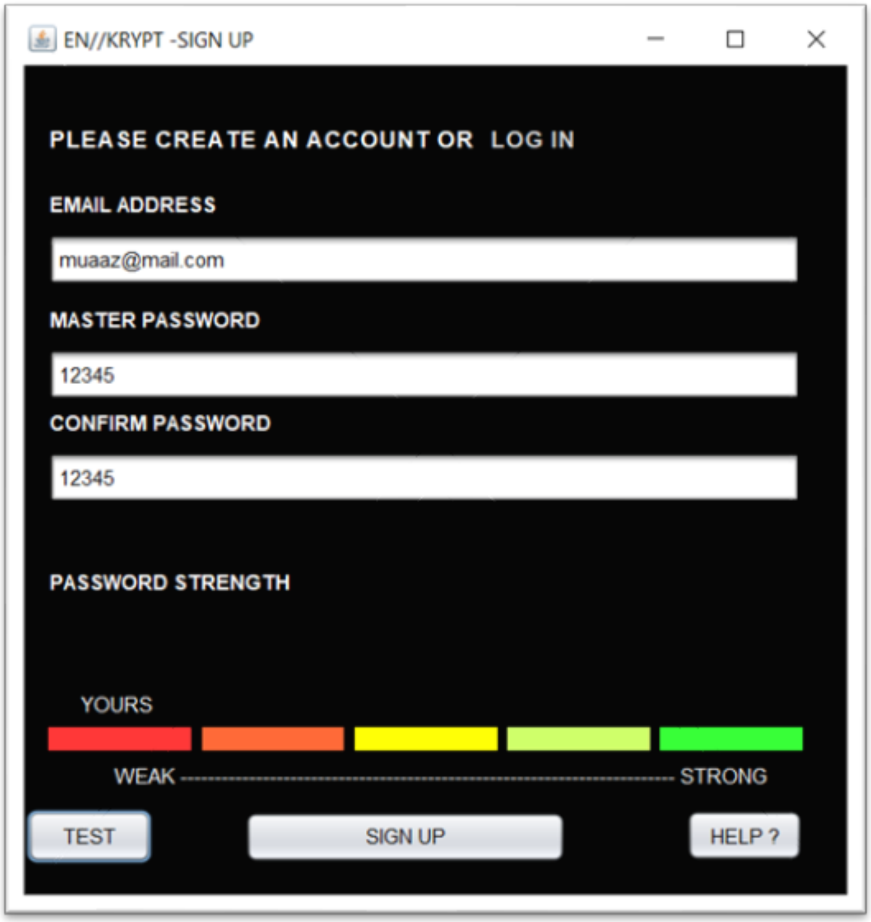
select edit button to update the entry	Y	Y
Users will be also able to :	Y	Y
access their account information and update it	Y	Y
access help and information as to how each of the functions work, as wekk as a walkthrough of each.	Y	Y
(Help icon will be available in the bottom of the screen at this menu as well as all other screens with detailed instructions and guides)	Y	Y
Exit : all screens will be terminated	Y	Y

4.2.3 Test Plan and Results

Username Variable (SignUpGUI)			
	Values Tested	Expected Results	Actual Results
Standard	“muaaz@mail.com”	Accepted	Accepted (fig 1)
Extreme	“m”	Accepted	Accepted (fig 2)
Abnormal	“”	Error Message	Error Message (fig 3)

Masterpassword Variable (SignUpGUI)			
	Values Tested	Expected Results	Actual Results
Standard	“tHis3-0JHn#)92i”	Accepted	Accepted (fig 4)
Extreme	“m”	Accepted	Accepted (fig 5)
Abnormal	“”	Error Message	Error Message (fig 6)

Fig 1


EN//KRYPT -SIGN UP

PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS

muaaz@mail.com

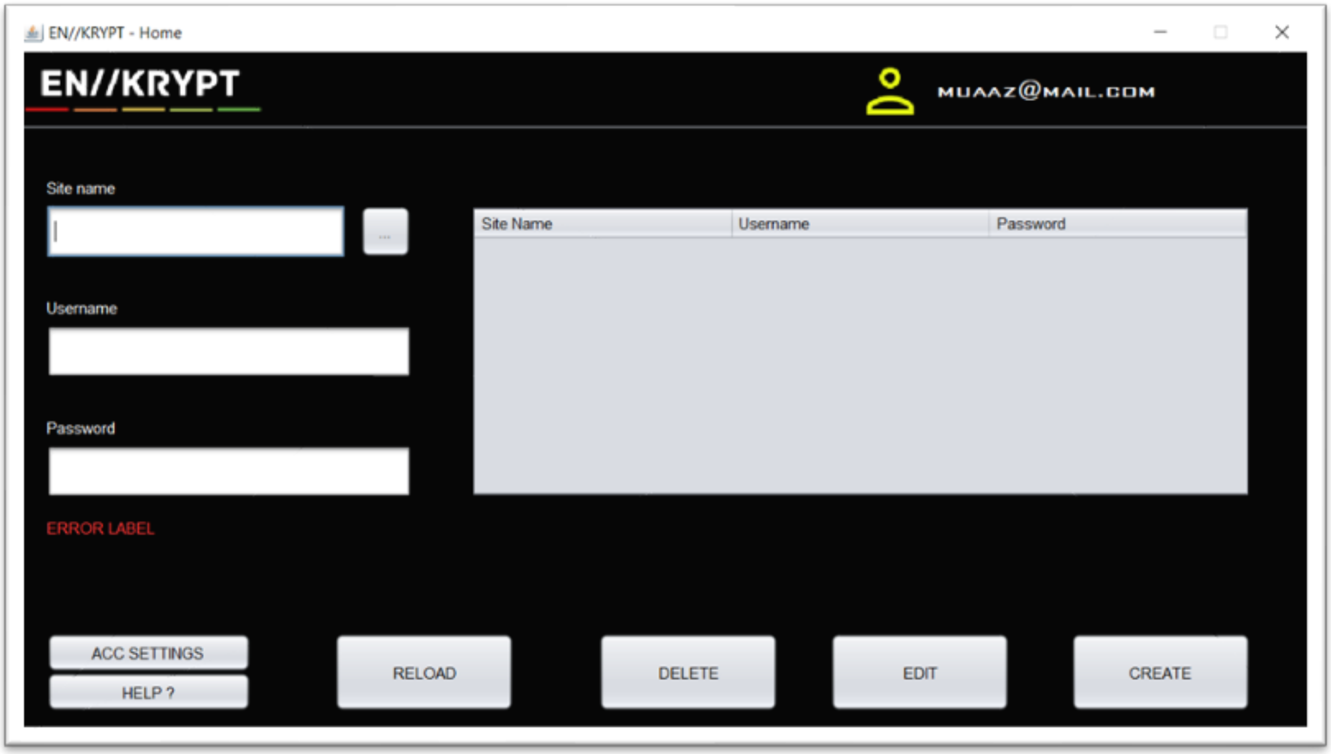
MASTER PASSWORD

12345

CONFIRM PASSWORD

12345

PASSWORD STRENGTH


EN//KRYPT - Home

EN//KRYPT

MUAAZ@MAIL.COM

Site name

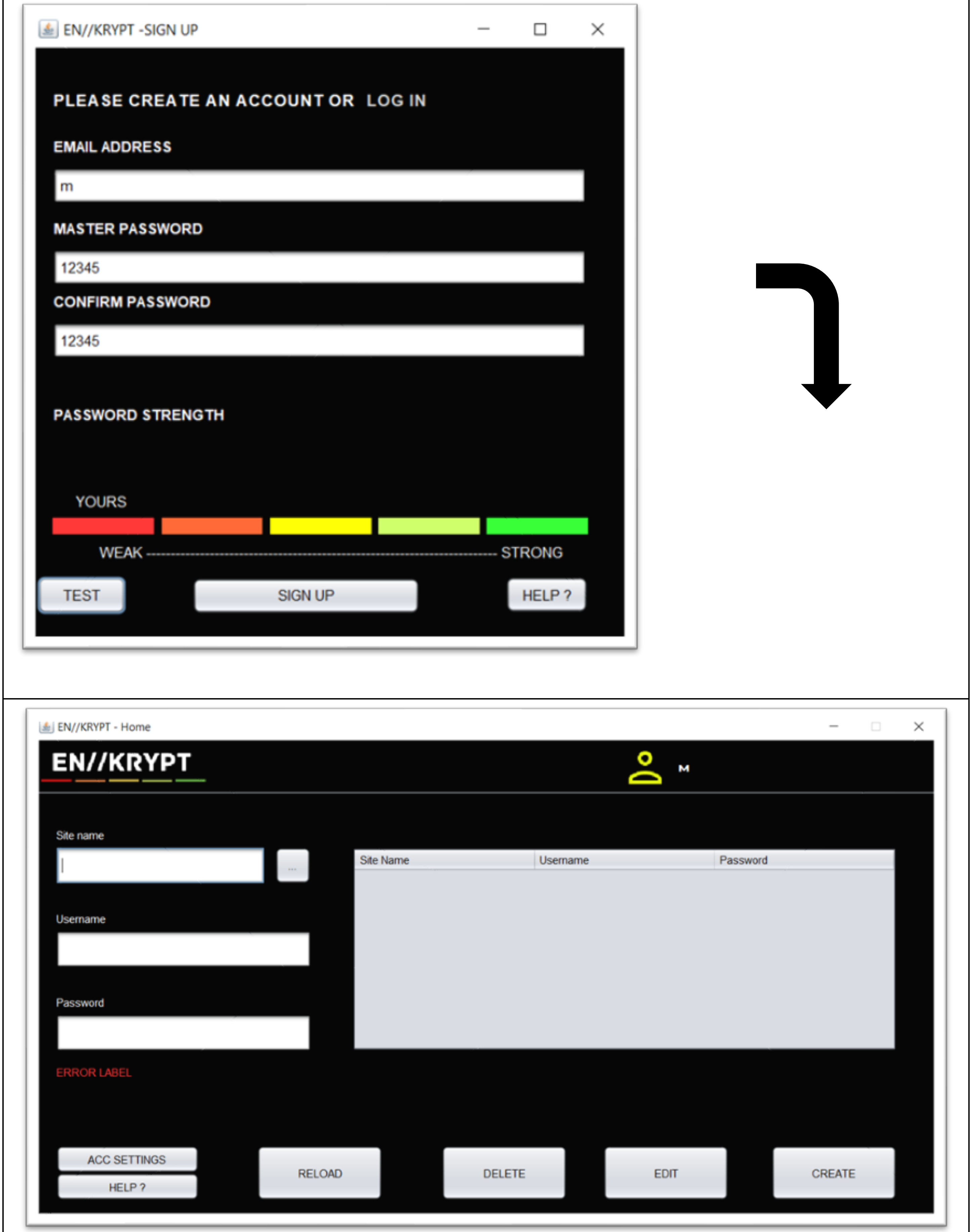
Username

Password

Site Name	Username	Password

ERROR LABEL

Fig 2



EN//KRYPT -SIGN UP

PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS: m

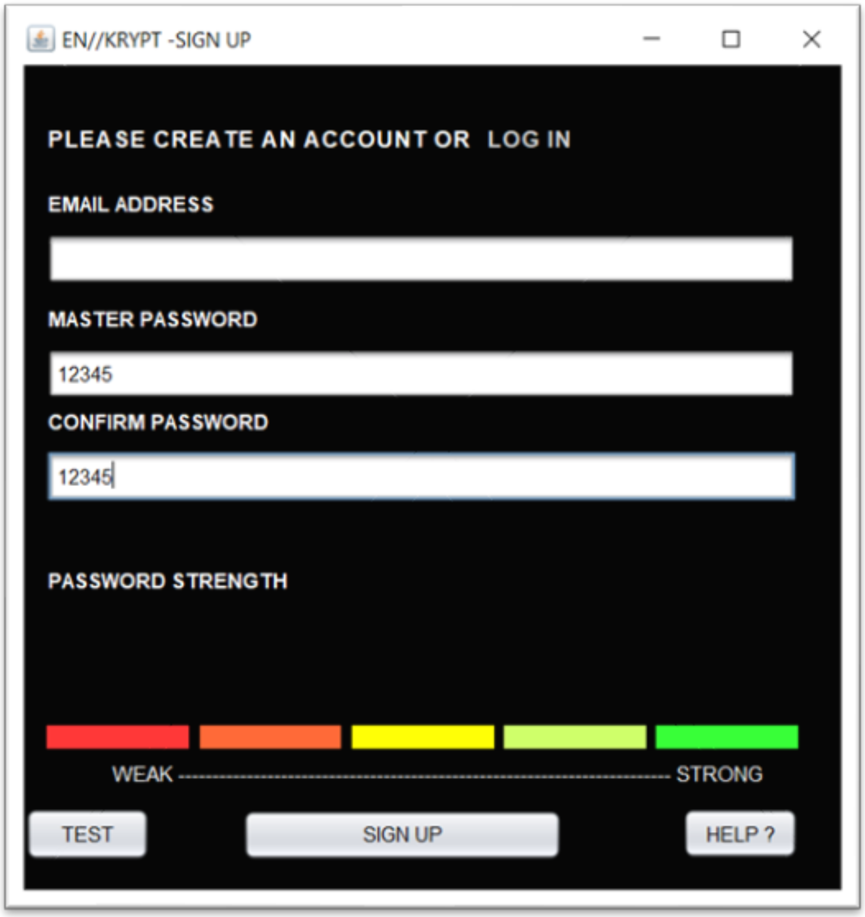
MASTER PASSWORD: 12345

CONFIRM PASSWORD: 12345

PASSWORD STRENGTH

EN//KRYPT - Home

Fig 3



EN//KRYPT -SIGN UP

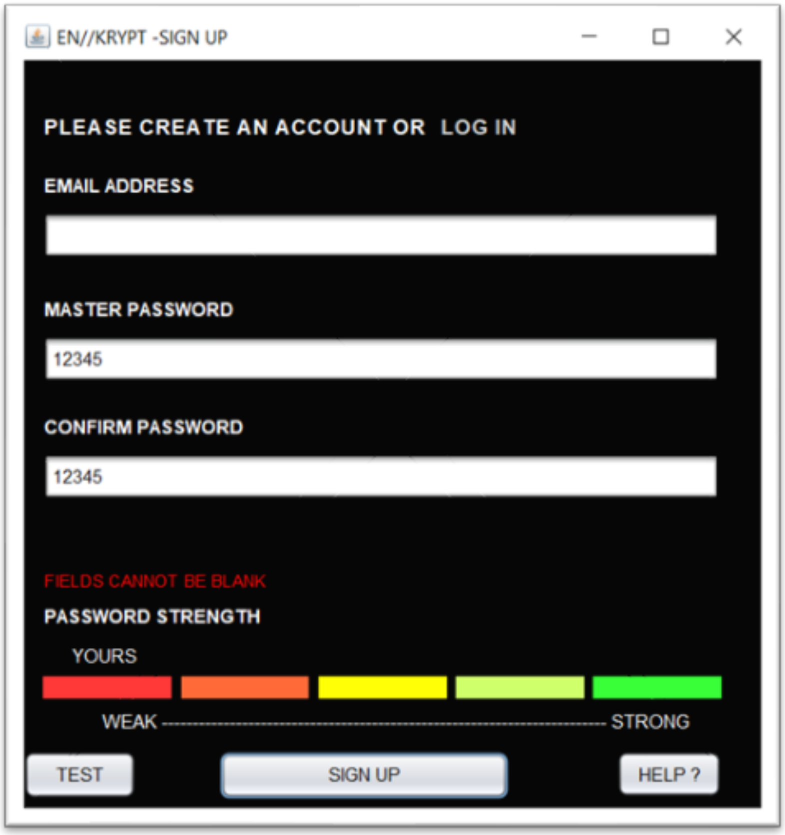
PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS

MASTER PASSWORD: 12345

CONFIRM PASSWORD: 12345

PASSWORD STRENGTH



EN//KRYPT -SIGN UP

PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS

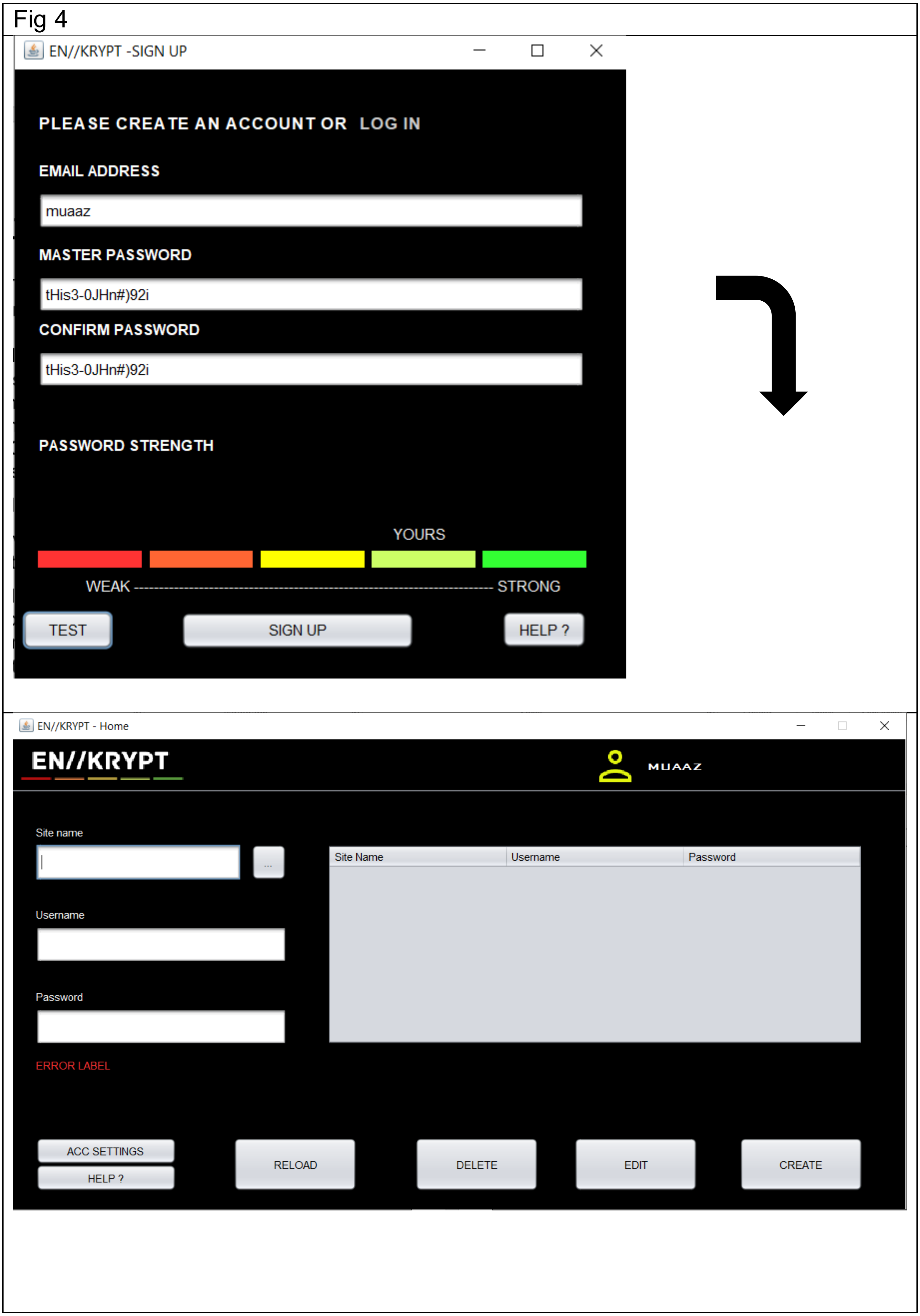
MASTER PASSWORD: 12345

CONFIRM PASSWORD: 12345

FIELDS CANNOT BE BLANK

PASSWORD STRENGTH

Fig 4



EN//KRYPT -SIGN UP

PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS: muaaz

MASTER PASSWORD: tHis3-0JHn#)92i

CONFIRM PASSWORD: tHis3-0JHn#)92i

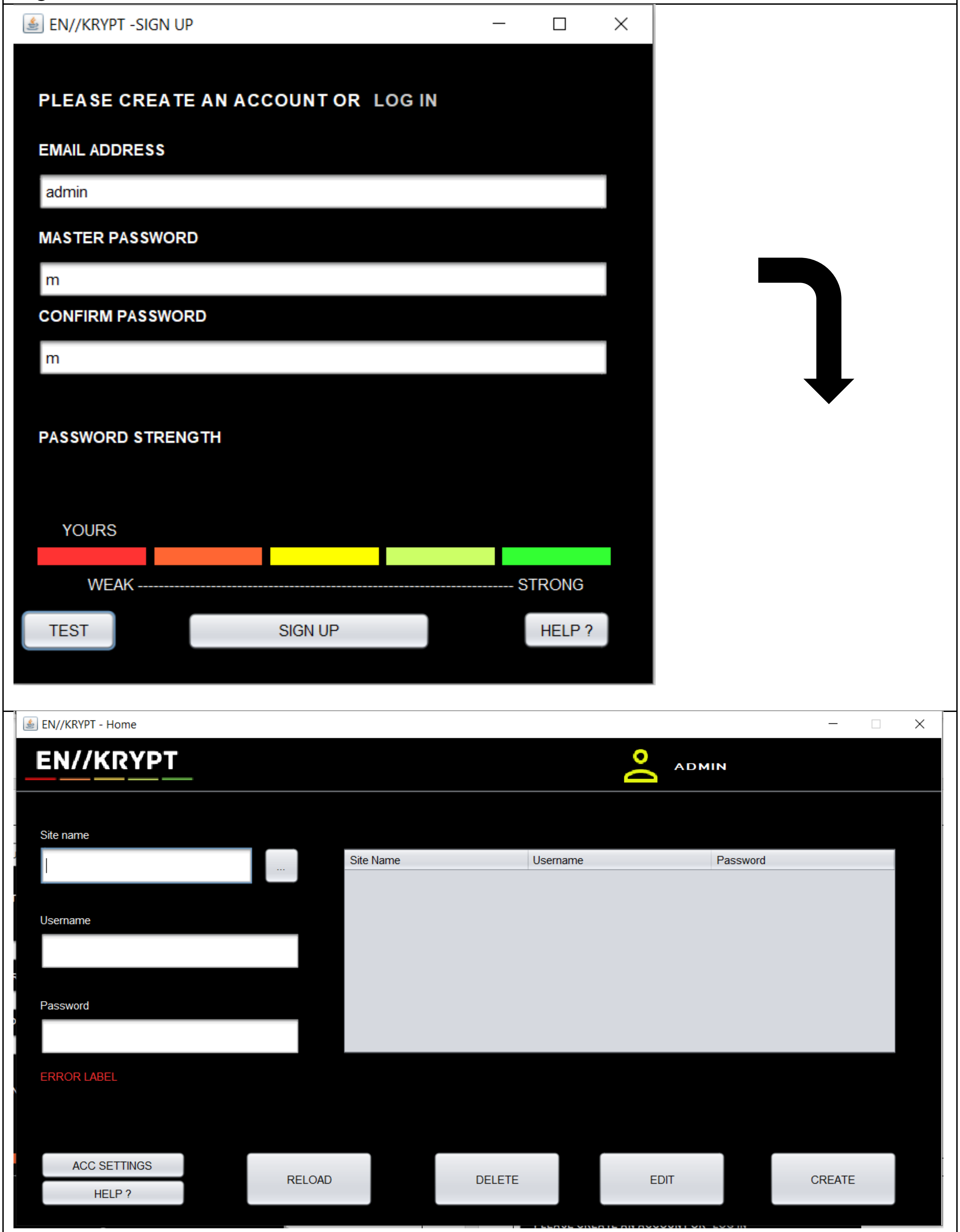
PASSWORD STRENGTH

YOURS
WEAK --- STRONG
TEST
SIGN UP
HELP ?

EN//KRYPT - Home

Site Name	Username	Password

Fig 5



EN//KRYPT -SIGN UP

PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS: admin

MASTER PASSWORD: m

CONFIRM PASSWORD: m

PASSWORD STRENGTH

EN//KRYPT - Home

Site name:

Username:

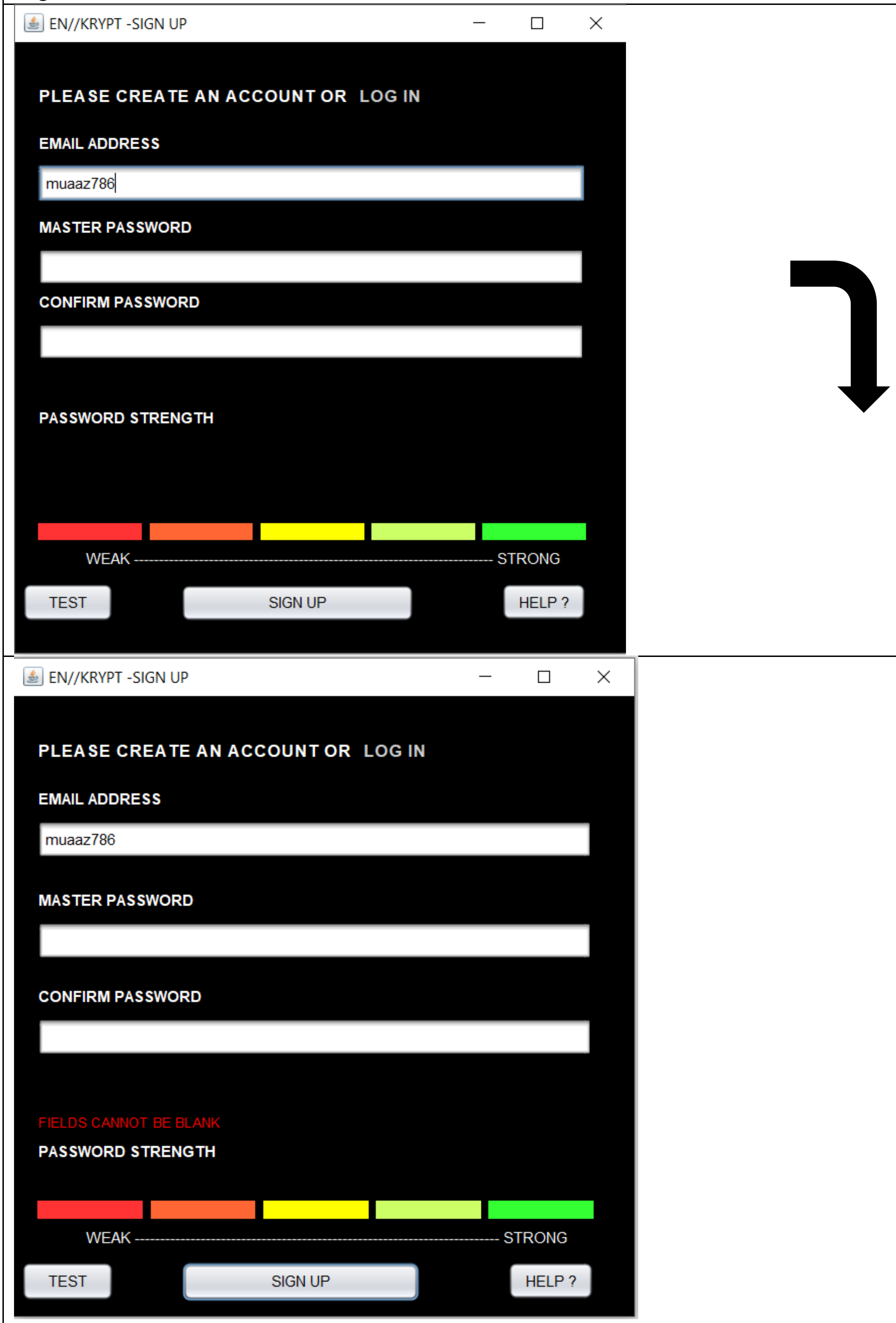
Password:

ERROR LABEL

Site Name	Username	Password

YOURS
WEAK -- STRONG
TEST
SIGN UP
HELP ?
EN//KRYPT
ADMIN
...
ACC SETTINGS
HELP ?
RELOAD
DELETE
EDIT
CREATE

Fig 6



EN//KRYPT -SIGN UP

PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS: muaaz786

MASTER PASSWORD:

CONFIRM PASSWORD:

PASSWORD STRENGTH

WEAK -- STRONG
TEST
SIGN UP
HELP ?

EN//KRYPT -SIGN UP

PLEASE CREATE AN ACCOUNT OR LOG IN

EMAIL ADDRESS: muaaz786

MASTER PASSWORD:

CONFIRM PASSWORD:

FIELDS CANNOT BE BLANK

PASSWORD STRENGTH

WEAK -- STRONG
TEST
SIGN UP
HELP ?