Make This

Chapter 9 – Password Authentication Using TinyDB Component

Logging into apps or web sites is very common. In this tutorial, you will set up a login screen for an app that checks data stored on your phone to authenticate the user (that the user name and password are correct). You will also create a set up screen that allows the user to create (or change) their credentials.

This tutorial teaches the following skills:

- **Using multiple screens in an app**
- **Using the TinyDB Component**
- **Comparing text strings to see if they match**

Note: Before attempting this exercise, complete the Chapter 2 and 3 exercises to familiarize yourself with the App Inventor interface, getting your Android device connected, and starting a new project.

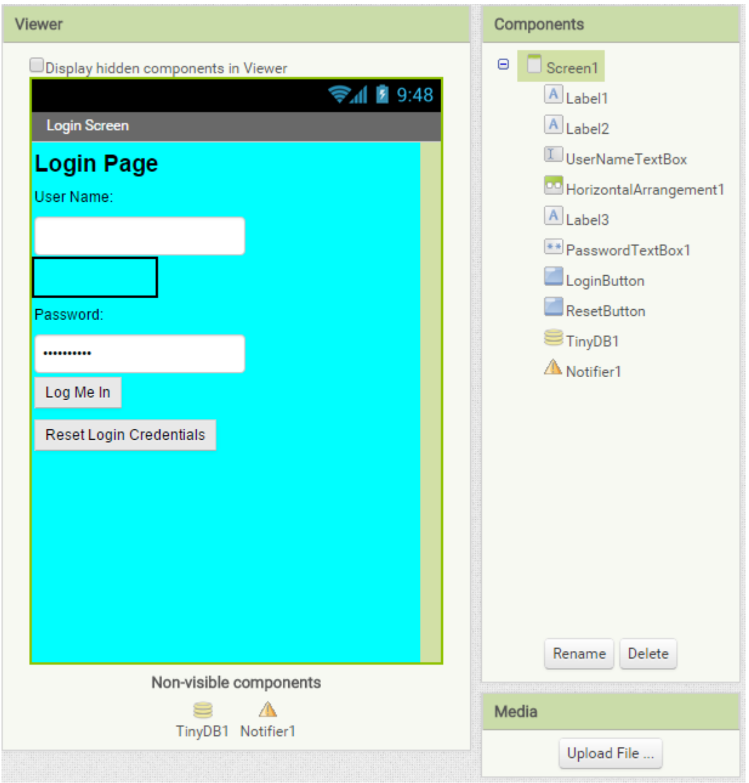
Building the Login App

1. Navigate to http://appinventor.mit.edu/explore/. If necessary, sign in with your Google Account.
2. Start a new project named *LoginApp*. Change the **Title** of **Screen1** to *Login Screen* and the **BackgroundColor** to *Cyan*.
3. Connect App Inventor to your Android device.

Part 1 – Constructing the Interface for Screen1

You are constructing an app that accepts login credentials. The interface in this example is simple: the user enters their user name and password, then clicks a button to login.

From previous chapter exercises, you should be familiar with constructing an interface. Your layout for Screen1 should look like this:



Build the interface shown above by dragging out the components shown in the following table:

Component	Palette Group	Component Name	Function
Label	User Interface	Label1	Page title
Label	User Interface	Label2	User name box label
TextBox	User Interface	UserNameTextBox	Box for entry of user name
HorizontalArrangement	Layout	HorizontalArrangement1	Provide spacing between boxes
Label	User Interface	Label3	Password box label
PasswordTextBox	User Interface	PasswordTextBox1	Box for entry of password (characters do not display when entered)
Button	User Interface	LoginButton	Click to check login information is recognized
Button	User Interface	ResetButton	Click to navigate to the SetUpScreen to establish (or change) credentials
TinyDB	Storage	TinyDB1	Component used to store information on

			your device
Notifier	User Interface	Notifier1	Component used to generate feedback messages

Set the properties of each component in the following ways:

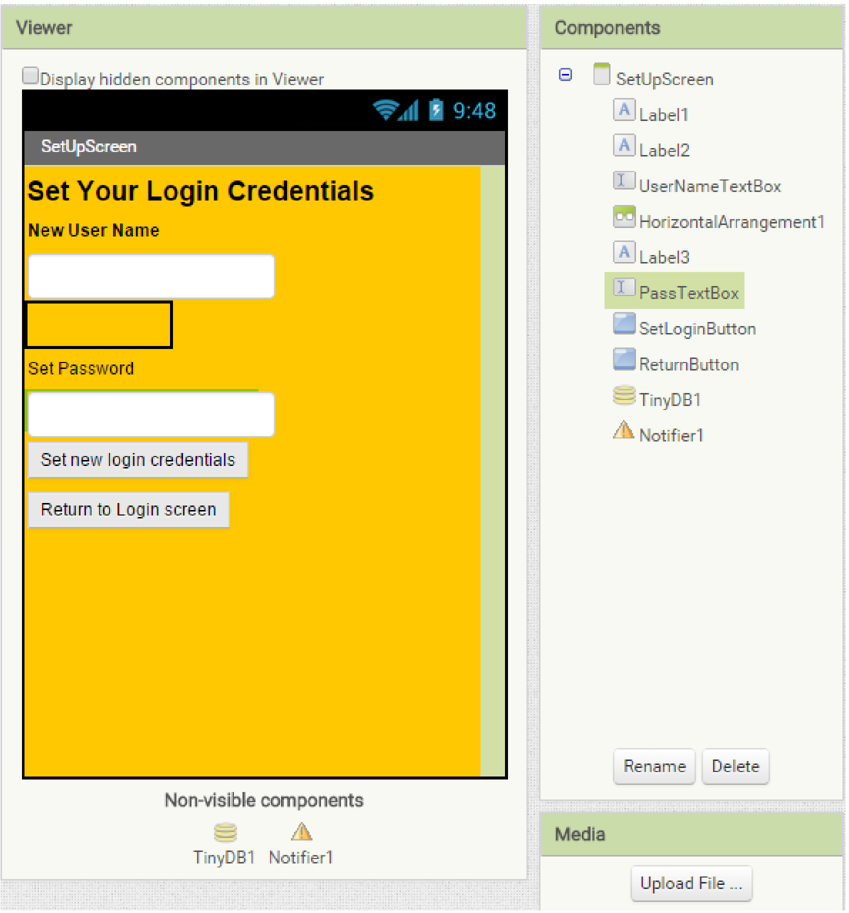
- Set the **Text** of **Label1** to *Login Page*. Change the **FontSize** to *22*. Check the **FontBold** check box to select it.
- Set the **Text** of **Label2** to *User Name:*
- Set the **Hint** of **UserNameTextBox** to *Enter login name*
- Set the **Height** of **HorizontalArrangement1** to *30 pixels*.
- Set the **Text** of **Label3** to *Password:*
- Set the **Hint** of **PasswordTextBox1** to *Enter password*
- Change the **Text** for **LoginButton** to *Log Me In*
- Change the **Text** for **ResetButton** to *Reset Login Credentials*

Part 2 – Constructing the Interface for a Second Screen

To avoid one screen being cluttered, or to clearly identify separate operations, apps often feature more than one screen. In this section of the tutorial, you will construct the interface for a second screen that is used when a user has to enter their login credentials for the first time to establish them.

1. At the top of the App Inventor interface, click the Add Screen ... button. In the **New Screen** dialog box, enter *SetUpScreen* in the **Screen name:** box. Click **OK**. A blank screen named SetUpScreen appears in the **Viewer**.

All the elements placed on a screen (such as Screen1) are distinct to that screen. You will need to add elements to the SetUpScreen as well. Your layout for the SetUpScreen should look like this:



Build the interface shown above by dragging out the components shown in the following table:

Component	Palette Group	Component Name	Function
Label	User Interface	Label1	Page title
Label	User Interface	Label2	User name box label
TextBox	User Interface	UserNameTextBox	Box for entry of user name

HorizontalArrangement	Layout	HorizontalArrangement1	Provide spacing between boxes
Label	User Interface	Label3	Password box label
TextBox	User Interface	PassTextBox	Box for entry of password
Button	User Interface	SetLoginButton	Click to set login credentials
Button	User Interface	ReturnButton	Click to return to Screen1
TinyDB	Storage	TinyDB1	Component used to store information on your device
Notifier	User Interface	Notifier1	Component used to generate feedback messages

Set the properties of each component in the following ways:

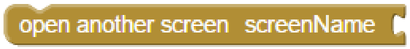
- Set the **BackgroundColor** of **SetUpScreen** to *Orange*.
- Set the **Text** of **Label1** to the **Set Your Login Credentials**. Change the **FontSize** to *22*. Check the **FontBold** check box to select it.
- Set the **Text** of **Label2** to the *New User Name*
- Set the **Hint** of **UserNameTextBox** to *Enter user name*
- Set the **Height** of **HorizontalArrangement1** to *30 pixels*.
- Set the **Text** of the **Label3** to *Set Password*
- Set the **Hint** of **PassTextBox** to *Enter password*
- Change the **Text** for **SetLoginButton** to *Set new Login credentials*
- Change the **Text** for **ReturnButton** to *Return to Login screen*

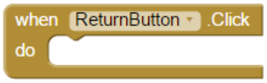
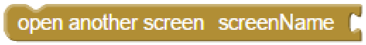
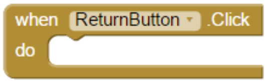
Tip: When using multiple screens in an app, make sure you provide the user with the functionality to navigate between screens. In this app, the ReturnButton will provide this functionality.

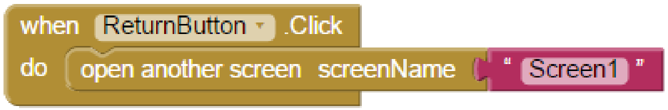
Part 3 – Program Functionality of the SetUpScreen

The first time this app is used, there are no credentials stored in the app. The user will need to come to the SetUpScreen first to create his or her credentials.

1. Change to **Blocks** view.

You need to program the **ReturnButton** to send users back to the Login Screen (**Screen1**). This is accomplished with a  block.

2. From the **ReturnButton** drawer, select a  block.
3. From the **Control** drawer, select a  block and place it inside the  block.
4. From the **Text** drawer, select a block and insert it in the open socket. Insert the text *Screen1* inside the block. Your block should now look like:



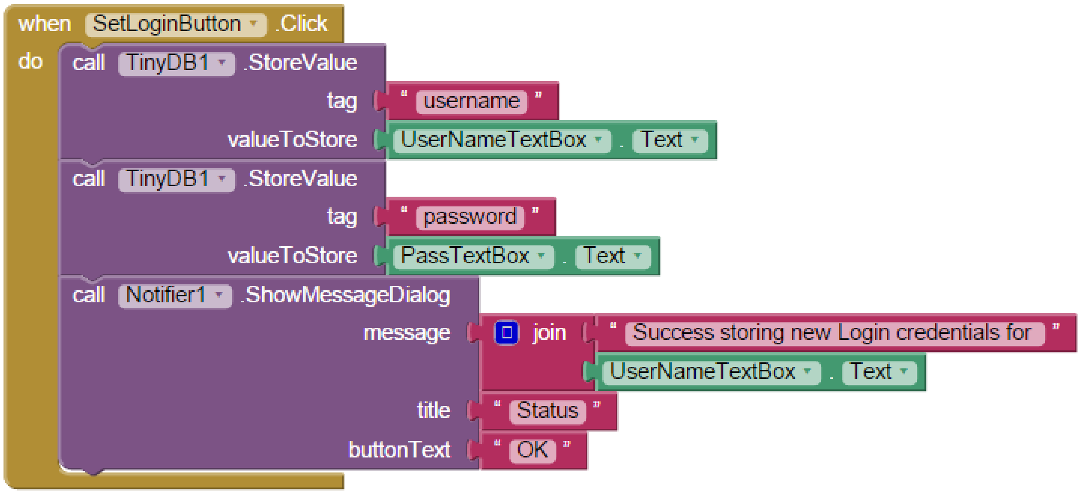
Alert: When using the block, it is imperative that the text in the text string block be the exact name of another screen in your app.

Now you need to program the functionality of the SetLoginButton. When pressed, this button needs to store the user's username and password for later use. A handy component for storing information in App Inventor is the TinyDB component.

Each time an app created with App Inventor runs, the app is initialized which sets variables to default values. If the user sets a value of a variable, the value of that variable will be lost when the user quits the app. In many apps, values need to be stored as **persistent data**. Persistent data is stored permanently so that its value can be retrieved any time the app is run (such as saving the high score of a game).

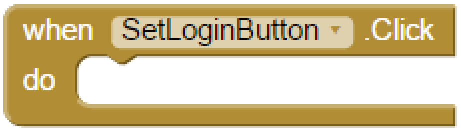
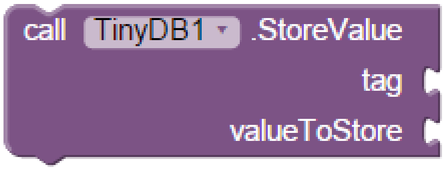
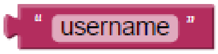
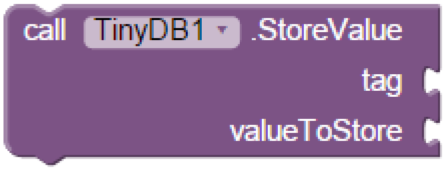
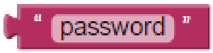
TinyDB is a persistent data store that will allow your app to store information ***to the device where the app is installed***. Data is stored as a string (a group of characters or numbers) that are identified by *tags*. A **tag** allows a piece of data to be identified. You can retrieve stored data by referencing the data's tag. For more information on TinyDB, see the App Inventor Component Documentation.

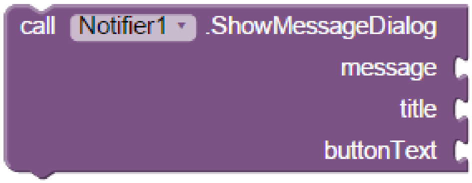
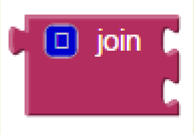
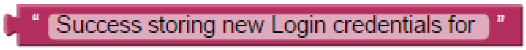
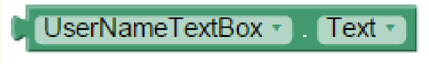
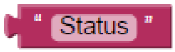
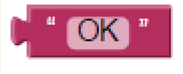
You'll need to create this block with the following instructions.



```
when SetLoginButton .Click
do  call TinyDB1 .StoreValue
                        tag  " username "
               valueToStore  UserNameTextBox . Text
    call TinyDB1 .StoreValue
                        tag  " password "
               valueToStore  PassTextBox . Text
    call Notifier1 .ShowMessageDialog
                    message  join  " Success storing new Login credentials for "
                                   UserNameTextBox . Text
                      title  " Status "
                 buttonText  " OK "
```

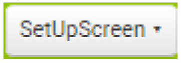
5. Drag out the blocks (as indicated in the table below) and arrange them as shown above.

Block	Drawer	Function
 when SetLoginButton .Click do	SetLoginButton	Controls what happens when SetLoginButton is clicked
 call TinyDB1 .StoreValue tag valueToStore	TinyDB1	Stores a value (as defined by valueToStore) with the tag specified
 " username "	Text	Name for the tag used to store the value entered as username
UserNameTextBox . Text	UserNameTextBox	Returns the value entered by the user as their username
 call TinyDB1 .StoreValue tag valueToStore	TinyDB1	Stores a value (as defined by valueToStore) with the tag specified
 " password "	Text	Name for the tag used to store the value entered as password
PassTextBox . Text	PassTextBox	Returns the value entered by the user as their password

 call Notifier1 .ShowMessageDialog message title buttonText	Notifier1	Display message dialog box
	Text	Join two strings to create dialog box message
	Text	Define first part of dialog box message to user
	UserNameTextBox	Return value of username – used as second part of the message to user
	Text	
	Text	Define text of button in message dialog box

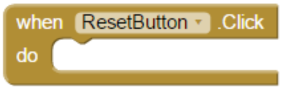
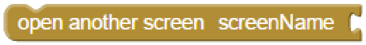
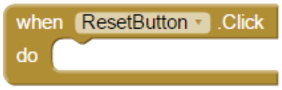
Test the functionality of this block by entering a username and password and clicking the **Set new login credentials** button. You should see a notification dialog box that requires clicking OK to dismiss. The password and username you entered have now been stored on your device by the TinyDB component. Click the **Return to Login Screen** button to return to **Screen1**.

Part 4 - Program Functionality of the Screen1 (Login Screen)

1. At the top of the App Inventor interface, click the down arrow on the  button and select *Screen1* from the dropdown list.

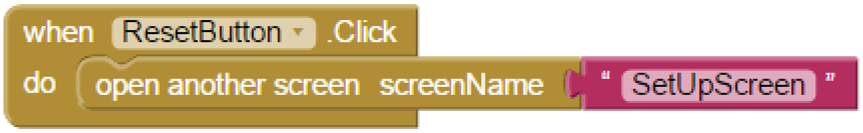
Alert: You may already have navigated to Screen1 during your testing of the SetUpScreen.

Note that there are no blocks in the Viewer window when you switch to Screen1. Blocks provide functionality to a *specific* screen, *not* to multiple screens. Therefore, you now need to build new blocks to code the functionality of Screen1.

2. From the **ResetButton** drawer, select a  block.
3. From the **Control** drawer, select a  block and place it inside the  block.

when ResetButton .Click
do
open another screen screenName
when ResetButton .Click
do

4. From the **Text** drawer, select a block and insert it in the open socket. Insert the text *SetUpScreen* inside the block. Your block should now look like:

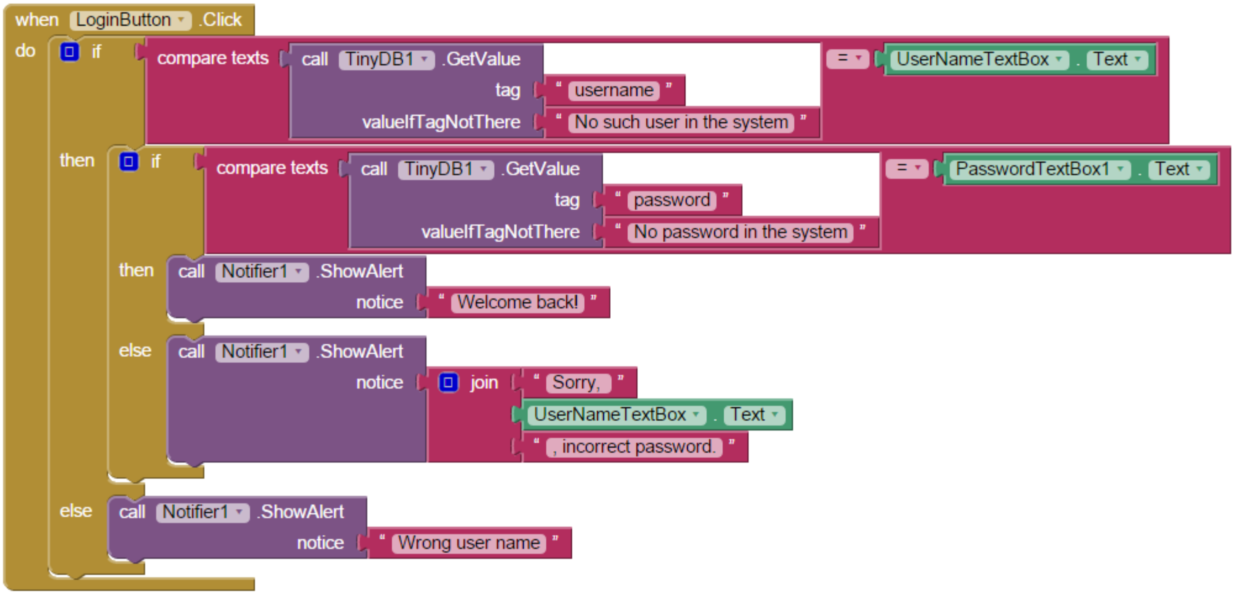


```
when ResetButton .Click
do  open another screen  screenName  " SetUpScreen "
```

Think about what needs to happen in your app when the **LoginButton** is pressed. The information that the user entered (user name and password) need to be checked against the data stored (by TinyDB) to see if it matches (i.e. – it is valid). There are several things that might happen and you need to check for them in this order:

- See if the user name matches the value stored in the database
  - If true – then check to see if the password entered matches
  - If false – then the user name was mistyped or doesn't exist…generate an onscreen alert
- See if the password matches the value stored in the database
  - If true – then generate a successful login message (Welcome back!)
  - If false – Password was mistyped (since we already know the user name is valid)…generate an onscreen alert for incorrect password

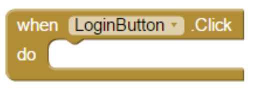
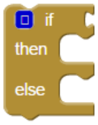
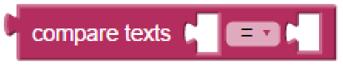
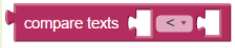
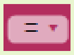
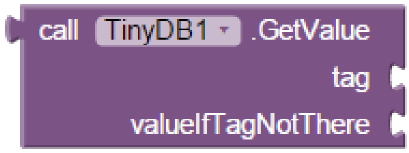
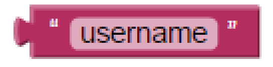
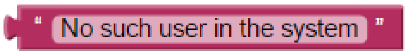
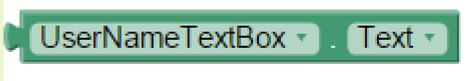
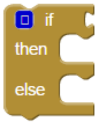
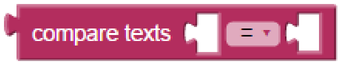
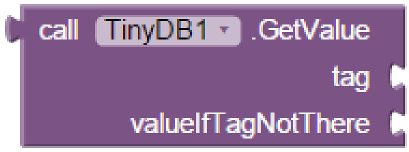
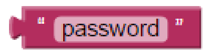
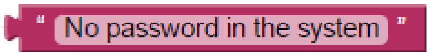
To accomplish these steps, you need to use a **nested if_then_else statement,** which consists of one if_then_else statement inserted inside another one. Your finished block will look like this:

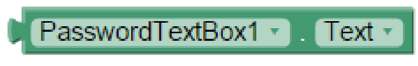
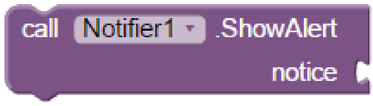
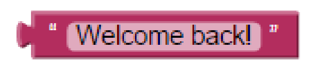
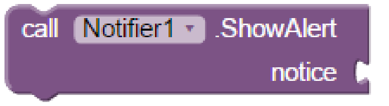
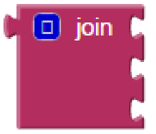
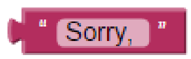
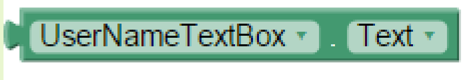
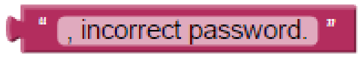
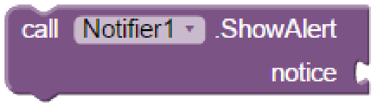
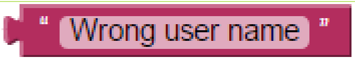


```
when LoginButton .Click
do  if  compare texts  call TinyDB1 .GetValue
                                      tag " username "
                        valueIfTagNotThere " No such user in the system "
                       =  UserNameTextBox . Text
    then  if  compare texts  call TinyDB1 .GetValue
                                            tag " password "
                              valueIfTagNotThere " No password in the system "
                             =  PasswordTextBox1 . Text
          then  call Notifier1 .ShowAlert
                          notice " Welcome back! "
          else  call Notifier1 .ShowAlert
                          notice  join " Sorry, "
                                       UserNameTextBox . Text
                                       " , incorrect password. "
    else  call Notifier1 .ShowAlert
                    notice " Wrong user name "
```

5. Drag out the blocks as indicated in the table below and arrange them as shown above.

Tip: If you need more than one of the same block, you can duplicate blocks (by right-clicking on them) instead of dragging them out of the drawer multiple times!

Block	Drawer	Function
 when LoginButton .Click do	LoginButton	Controls what happens when LoginButton is clicked
 if then else	Control – Drag out if_then block and modify it to if_then_else by clicking the mutator	Checks if condition is true or false – then either does one thing or another
 compare texts =	Text – Drag out a  compare texts < block and change to an  = block	Used to compare two text strings to see if they are equal
 call TinyDB1 .GetValue tag valueIfTagNotThere	TinyDB	Retrieves value from database based on tag indicated
 " username "	Text	Tag to search for in TinyDB
 " No such user in the system "	Text	Value returned if tag not found
 UserNameTextBox . Text	UserNameTextBox	Returns value entered by user as user name
 if then else	Control	if_then_else block to be nested inside first if_then_else block
 compare texts =	Text	Used to compare two text strings to see if they are equal
 call TinyDB1 .GetValue tag valueIfTagNotThere	TinyDB	Retrieves value from database based on tag indicated
 " password "	Text	Tag to search for in TinyDB
 " No password in the system "	Text	Value returned if tag not found

 PasswordTextBox1 . Text	PassWordTextBox1	Returns value entered by user as password
 call Notifier1 .ShowAlert notice	Notifier1	Displays alert that fades away automatically (for correct login)
 " Welcome back! "	Text	Alert message text for correct login
 call Notifier1 .ShowAlert notice	Notifier1	Displays alert that fades away automatically (for incorrect password)
 join	Text	Joins three text strings for incorrect password message
 " Sorry, "	Text	First part of incorrect password message (**Note**: insert a space after the comma)
 UserNameTextBox . Text	UserNameTextBox	Returns value entered as user name – 2nd part of incorrect password message
 " , incorrect password. "	Text	3rd part of incorrect password message
 call Notifier1 .ShowAlert notice	Notifier1	Displays alert that fades away automatically (for incorrect user name)
 " Wrong user name "	Text	Incorrect user name message

Test your app. Login credentials must be established otherwise nothing is stored in the database to be retrieved. So click **Reset Login Credentials** and enter a user name and password. Then return to the Login screen and try logging in as follows:

1. With an incorrect user name
2. With a correct user name and incorrect password
3. With a correct user name and password

Ensure you receive an appropriate alert message in each case.

Extensions to This Project

1. Since TinyDB stores information by tag, this program currently will only store one username and password at a time. If you set credentials for a second user, the app uses the tags *username* and *password* to store the new credentials…overwriting the previous information. Modify the program so that it accepts and stores the login credentials for more than one user.
    a. Hint – Think about using lists. Check out the Map It tutorial for some ideas.
2. TinyDB only supports storing data on one device. If your app needs to access persistent data for multiple users on multiple devices, you need to use the component TinyWebDB. Modify your app to use TinyWebDB instead of TinyDB. See Creating a Custom TinyWebDB Service for information on setting up this component.

Resources

- AI2 Storage Components: TinyDB
- AI2 Text Blocks: Compare Texts
- User Guide for App Inventor 2
- Guide to Understanding Blocks