Unit1 Programming

Jlaal Mohammed

305357

Assignment 2

Unit 1 Programming

Table of Contents

**Table of Contents……………………………………………………………………………………………………1**

**Write a program that implements an algorithm using an IDE.2**

**Enhance the algorithm written, using the features of the IDE to manage the development process………………………………………………………………………………………………………………3**

**Explain the debugging process and the debugging facilities available in the IDE. 4**

**Examine how the debugging process can be used to help develop more secure, robust applications.5**

**Explain the coding standard you have used in your code6**

**References7**

P4 Write a program that implements an algorithm using an IDE.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Diagram

A diagram of a flowchart

Description automatically generated

**Pseudocode**

# Importing libraries

IMPORT tkinter.

IMPORT os

# Delete functions

FUNCTION to delete widget

FUNCTION which deletes multiple widgets at once

# Login success pop-up window message function

FUNCTION login success

CREATE pop up window message

SET window title to Login success

SET window size to “a X y”

CREATE Label saying “Login success!”

CREATE Button saying “OK” and contains delete function

# Password not recognised pop-up window message function

FUNCTION password not recognised

CREATE pop up window message

SET window title to “Password not recognised”

SET window size to “a X y”

CREATE Label saying “Password Incorrect!”

CREATE Button saying "OK" and contains delete function

# Username not found pop-up window message function

FUNCTION username not found

CREATE pop up window message

SET window title to “User not found”

SET window size to “a X y”

CREATE Label saying “User not found!”

CREATE Button OK and contain delete function

# Staff ID correct pop-up window message function

FUNCTION staffID correct

CREATE pop up window message

SET window title to “staffID correct”

SET window size to “a X y”

CREATE Label saying "StaffID correct!"

CREATE Button “OK” and contain delete function

# Staff ID incorrect pop-up window message function

FUNCTION staffID incorrect

CREATE pop up window message

SET window title to "StaffID not found"

SET window size to "a X y"

CREATE Label saying "StaffID Incorrect!"

CREATE Button “OK” and contain delete function

# Function to verify login credentials

FUNCTION login\_verify():

GET username1 from username\_verify.get()

GET password1 from password\_verify.get()

GET staffID1 from staffID\_verify.get()

DELETE contents of username\_entry2

DELETE contents of password\_entry2

DELETE contents of staffID\_entry2

# Check if username exists in files

GET list of files from directory

IF username1 IN list of files THEN

OPEN file1 with username1 for reading

READ and verify from file1 and split by lines

IF password1 IN verify THEN

CALL login success

ELSE

CALL password not recognised

ELSE

CALL username not found

IF staffID1 IN verify THEN

CALL staffID correct

ELSE

CALL staffID incorrect

# Function to register a new user

FUNCTION register user

GET username info from username.get

GET password info from password.get

GET staffID info from staffID.get

# Write username, password, and staffID to a file

OPEN file with username info for writing

WRITE username info to file

WRITE password info to file

WRITE staffID info to file

CLOSE file

# Clear entry fields

DELETE contents of username entry

DELETE contents of password entry

DELETE contents of staffID entry

# Display account creation message

CREATE Label in window saying "Account created!"

CREATE Label in window saying "Press Main menu when done!"

# Function to open registration window

FUNCTION register

CREATE pop up window message

SET window title to "Account Creation"

SET window size to "a X y"

SET window background to blue

# Global variables

SET username, password, staffID, username entry, password entry, staffID entry

# Store variables

SET username to string variable

SET password to string variable

SET staffID to string variable

# Create labels and entry fields for username, password, and staffID

CREATE Label inside window saying "Welcome to the registration page, please enter your details below in order to create an account, Thank you!"

CREATE Label inside window saying "Username"

CREATE Entry with text variable username

CREATE Label inside window saying "Password"

CREATE Entry with text variable password

CREATE Label i inside window saying "Staff ID"

CREATE Entry with text variable staffID

# Create Register and Main Menu buttons

CREATE Button in window saying "Register" and use command register user

CREATE Button in window saying "Main Menu" and command delete

# Function to open login window

FUNCTION login

PRINT "Welcome to Information Systems LTD

CREATE pop up window

SET window title to "Information Systems Login page"

SET window size to "a X y "

# Set variables for username, password, and staffID

SET username Verify, password Verify, staffID Verify to String Variable

# Create labels and entry fields for username, password, and staffID

CREATE Label in window saying "Welcome to the login page ,please enter your details below in order to continue, Thank you!".

CREATE Label inside window saying "Username"

CREATE Entry with text variable username verify

CREATE Label inside window saying "Password"

CREATE Entry with text variable password verify

CREATE Label inside window saying "Staff ID"

CREATE Entry with text variable staffID verify

# Create Login button

CREATE Button in window with text "Login" and command login verify

# Main window function

FUNCTION main window

CREATE window

SET window title to "Information Systems Login and Registration Page"

SET window size to "a X y"

SET window background to blue

# Create labels and buttons for login and registration

CREATE Label saying “Information Systems Login and Registration page"

CREATE Button saying "Login" and command login

CREATE Button saying "Create an account" and command register

CREATE Label inside window saying "If you do not have an account please create an account using your staff ID!"

START main event loop

CALL main window

M3 Enhance the algorithm written, using the features of the IDE to manage the development process

**Appendix**

**A screenshot of a computer screen

Description automatically generated**

This is the main menu form which introduces you to two option the login option which is a button and the create an account option which is also uses a button. The message below indicates if you don’t have an account you cannot login therefore you register one.

**A screenshot of a computer screen

Description automatically generated**

This is the login page which can be entered by clicking on the login button. The message above tells the user to enter the details in the appropriate text boxes and click login. When that action has been preformed a message box will appear and depending on if the credentials inputted were correct it will say “Login Success”, if not it will say “User not found”/”Password Incorrect”.

This is the register page which allows the user to create an account as per instructions shown at the top of the page. When the details are entered, and the user clicks on register the details will be stored in a file that the system will be able access when it needs to. Then a message appears to prompt the user to click on the main menu button and login.

A screenshot of a computer screen

Description automatically generated

|  |  |
| --- | --- |
| **Screenshot of development** | **Description** |
|  | Libraries such as Tkinter which allows the creation of a GUI and the OS in order to access the internal system in which the program is running on. |
|  | The main window function displays the main menu which contains the window size, title name and background colour. Beneath that the widgets which include labels, buttons, font sizes and colour. |
|  | This is the login function which displays the login page. The global keywords are there to make sure that those variables highlighted can be used outside this function and be used in others. The StringVar is used so Tkinter can be retrieved.  Below that are labels for username, password and staffID with their corresponding entry boxes which allow the user to input their details. Lastly a button to confirm the login action. |
|  | This is the register function which is the creation of the register page. The global variables are used so they can be used outside of this function. The StringVar are used so that the information that the user has inputted to register will be stored.  Below that are the associated labels, entries and buttons. |
|  | The register user function will be connected to the register function as this function allows the program to store the details the user inputted into a file to use when needed and then clear the entry fields when it has done registering. |
|  | The login verify function will be connected to the login function as this function accesses the created file that the user had registered and then verify using if statements if the details the user has inputted are correct if so another function will if incorrect a different function will appear. |
|  | These are the functions that will appear from the login verify function. These functions will appear as a small pop message window depending on which one is needed. Each of these functions contain a command with a delete function to close multiple or individual windows when a button is clicked. |
|  | The first delete function only allowed me to delete one widget per command so by creating the second below allowed me to delete multiple widgets within one command. |

Explain the debugging process and the debugging facilities available in the IDE

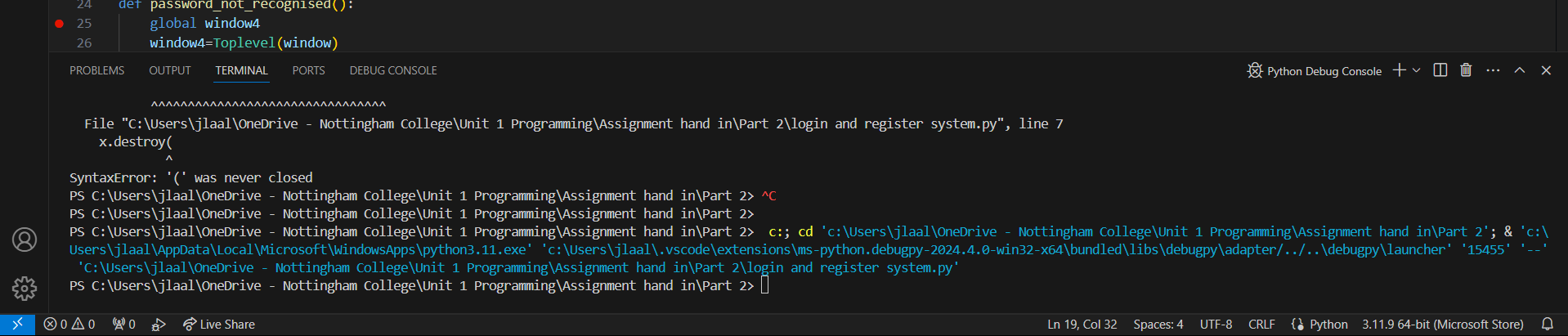
The debugging process is an important procedure which plays its role in testing to allow the user to find bugs and errors and eliminate them. The user can either manually search for bugs within the code or the user can use a code debugging software which in most cases are built in such as PyCharm or Visual studio code, to search for bugs but also find them and give a brief description of what the error is. The basic steps for debugging would be to replicate the bug, then identify what is causing the bug to occur and then how do you go fixing it. Debugging facilities are tools which help the user to identify and fix bugs in the programming environment. The many debugging tools are:

* Call stack examination- This allows the user to examine the call stack and to see the flow of functions up to the point of execution. This can help identify where errors are occurring in the program.
* Step by step execution- The user will go through the code one line at a time observing the behaviour of the program. This can help the user identify where exactly the bug occurs and shows how the state of the program changes.
* Breakpoints- These are used as markers telling the debugger to stop the execution. When that occurs, this allows the user to inspect and analyse the state of the programme.
* Conditional breakpoints- These are specific conditions which the debugger will pause the execution. This helps the user if the bug only appears in specific conditions.
* Stack traces- These provide details of the function leading up to the error. This helps the user to find the exact location and find out what caused the error.
* Variable inspection- Observing the values of variables in different points of the execution. This helps the user identify where errors are what the state of the programme is in.
* Logging- Logging allows the user to output errors to a log console or file while the program is being executed. This is especially important for tracing the flow of information during the execution as well as monitoring the program in real time and resolving issues.

M4 Examine how the debugging process can be used to help develop more secure, robust applications.

Explain the facilities available in an IDE to debug your application.

The IDE that was used in order to create the code for the login and register system was Visual Studio Code, below will be screenshots explaining the debugging process and the facilities available in Visual Studio Code IDE.



This picture shows the integrated debugging interface which allows the user to see what problems the code has, the output, terminal, and the debugging console itself. The red dot above the console is a breakpoint that can be used to stop the flow of the code when it is executed.

A screenshot of a computer

Description automatically generated

Visual Studio Code supports multiple languages such as C#, C/C++,Java, Python, JavaScript, HTML/CSS and more. This makes Visual Studio Code one of the most diverse IDE’s that a user could have as they you could also use it in projects such as websites, databases, or API’S. They also contain debugger extensions which allow the debugger to preform with more capabilities and offer more features such as debugging tools, custom debuggers, and third-party tools.

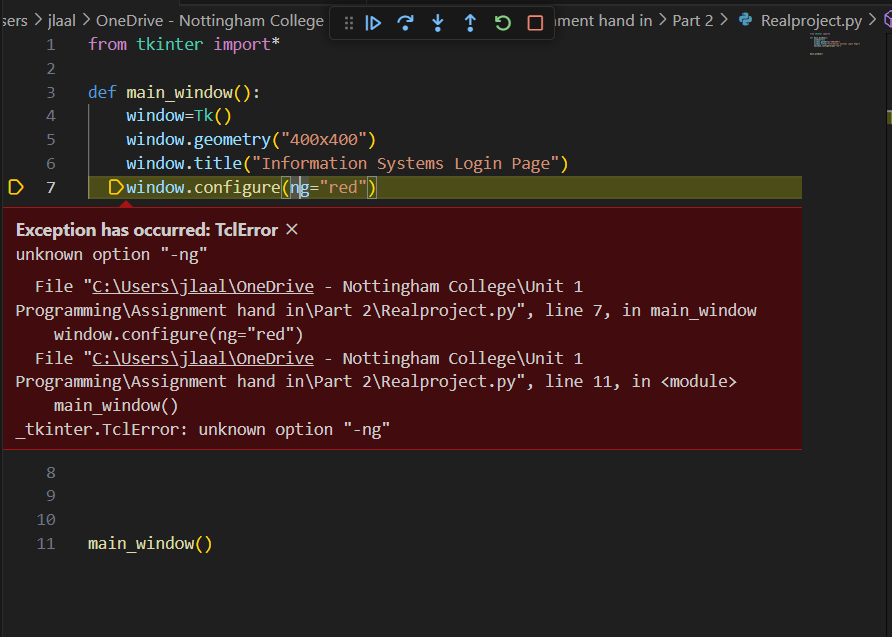
A screenshot of a computer program

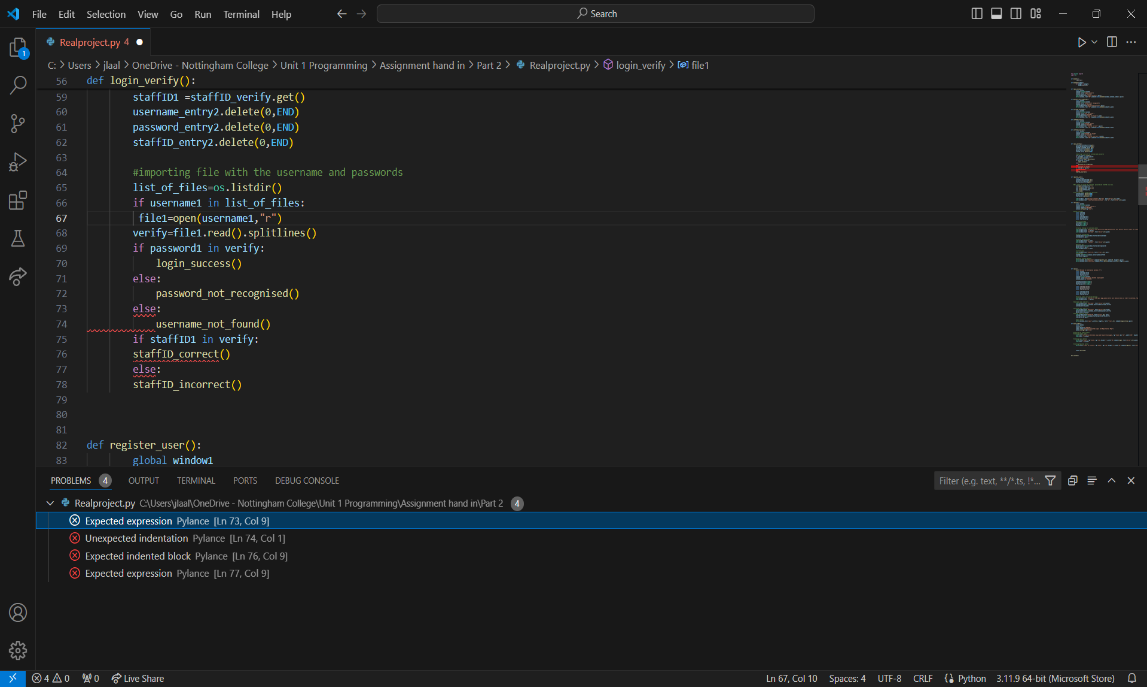
Description automatically generatedThis is an interactive debugging feature which allows you to hover over lines of code to inspect what the problem is and evaluate the expressions in that line this allows the developer a much easier method to view and solve issues without scrolling through the debugging panel.

Examine how the debugging process can be used to make your applications more robust and secure.

The debugging process can be used to make applications robust and secure by searching for issues within the code such as errors, bugs, and vulnerabilities within the code. By searching and addressing these problems in the code it will allow the developer to know what issues are causing the code to not execute so that the developer can fix them. You can also look through logs or stack traces to understand where the location of the issue is coming from. By going through the logs and tracing the flow of code with stack traces developers can stop security flaws from being present so they cannot be exploited. They can also go through testing various parts of the code in order to reproduce errors, testing can also be used to see how well the security of the code works such as authorisation checks, data validation and encryption and by replicating scenarios in which the code is attacked the developers will have the know how and understanding on how to stop them. You could also use tools to help find vulnerabilities in the code as they will be able to find issues that may not be apparent when debugging manually but these tools can also help make debugging a more efficient process.

You also need print screens showing any problems you have encountered when developing your application and any enhancements developed

In this scenario I have simply not used the correct spelling when wanting to set the background colour. There were also many other instances where errors occurred due to not checking spelling correctly so in order to solve these problems I would have to manually go through the code and change them.



Above shows multiple errors which again have appeared many times when working on this project is that making sure all indentations are done correctly and making sure they also spaced correctly otherwise going through the code to manually solve the problem or in some cases I have had to re-write code several times in order to get rid of the problem.



A screen shot of a computer program

Description automatically generated

Above shows how I tried to use the delete function to delete multiple windows at once which created the error above. This error occurred due to using one function to do delete multiple windows which the system didn’t allow so in order to fix this I created another delete function shown below which allowed me to delete multiple windows at once. I than implemented this in the code as shown using a lambda function which fixed the problem.

A screen shot of a computer screen

Description automatically generated



P6 Explain the coding standard you have used in your code

The coding standards that were used in my code are:

* Imports should usually be on separate lines, Always written at the top of the file
* Avoid using white space in situations where you use brackets, braces, and parentheses
* When identing only use up to 4 spaces
* Using white space when trying to make code easier to read and to follow.
* Using blank spaces to separate functions and classes
* Using comments to explain code which may be difficult to understand, but also explaining what may happen what may go wrong. Also make sure that comments are up to date.
* Using class names, they should be formatted in the CapWords i.e. UserName, StringVar
* A limit all line lengths to 79 characters
* String quotes should be decided to either use single-quoted string or a double-quoted string do not interchange between both. If using triple-quoted always use double in order to be consistent with docstrings standard.

**References**

* [What is Debugging? - Definition from Techopedia](https://www.techopedia.com/definition/16373/debugging)
* [Nine anti-debugging techniques for application security - Security Boulevard](https://securityboulevard.com/2020/09/nine-anti-debugging-techniques-for-application-security/)
* [Debugger Security - Visual Studio (Windows) | Microsoft Learn](https://learn.microsoft.com/en-us/visualstudio/debugger/debugger-security?view=vs-2022)
* [What is Debugging in Software Engineering? - GeeksforGeeks](https://www.geeksforgeeks.org/software-engineering-debugging/)