

**GROUP ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**AAPP010-4-2-PWP**

**PROGRAMMING WITH PYTHON**

**UCDF2007ICT(ITR), UCDF2007ICT(DI), UCDF2007ICT(SE), UCDF2007ICT, UCDF2007BIT**

**HAND OUT DATE: 02nd JUNE 2021**

**HAND IN DATE: 13TH AUGUST 2021**

**WEIGHTAGE: 100%**

**INSTRUCTIONS TO CANDIDATES:**

1. Submit your assignment online in MS Teams unless advised otherwise
2. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld
3. Cases of plagiarism will be penalized
4. You must obtain at least 50% in each component to pass this module

**Table of Contents**

[**Introduction and Assumptions** 3](#_Toc79767237)

[**Design of the Program** 4](#_Toc79767238)

[**Pseudocode** 4](#_Toc79767239)

[**Flowcharts** 23](#_Toc79767240)

[**Program Source Code with Explanation** 35](#_Toc79767241)

[**Additional Features Source Code with Explanation** 43](#_Toc79767242)

[**Screenshots of Sample Input / Output with Explanation** 44](#_Toc79767243)

[**Conclusion** 56](#_Toc79767244)

[**Reference** 56](#_Toc79767245)

Group Members

|  |  |
| --- | --- |
| Name | TP.NO |
| **Lee Jia Qian** | **TP 061863** |
| **Tee Chor Yang** | **TP 061339** |

# **Introduction and Assumptions**

SPIDERMAN ONLINE FOOD SERVICES(SOFS) is one of the fast-growing Online Food Services in Malaysia which helps customer to save their time by ordering food online from their place such as home, office and various other location. Furthermore, in this document, SOFS have requested us to enhance their online order services by including various kind of dishes in one click. Therefore, we have created a neat and simple program to help the company. Firstly, we created a program which allows user to access system, register, view, modify, add and many more features. The program starts off with the user being a guest which then allow the user to access the system such as admin or customer page. To access the admin or customer page, user is required to login. Moreover, user is able to register an account to be a customer while admin ais fixed to those who only have access. Once user have login to the customer page, they are able to view menus, order and proceed to check-out once order have been made while user who is able to access the admin page have access to feature such as modifying, add food items and view records.

# **Design of the Program**

## **Pseudocode**

Start Page

START

DECLARE option

DEFINE Start()

DISPLAY “Welcome To SOFS”

DISPLAY “1. View as Guest.”

DISPLAY “2. Login as Customer.”

DISPLAY “3. New Registration.”

DISPLAY “4. Login as Admin.”

DISPLAY “5. Exit.”

READ option “Select Option: ”

IF (option = 1) THEN

CALL FUNCTION Guest()

ELSE

IF (option = 2) THEN

CALL FUNCTION Customer()

ELSE

IF (option = 3) THEN

CALL FUNCTION Register()

ELSE

IF (option = 4) THEN

CALL FUNCTION Adminlogin()

ELSE

IF (option = 5) THEN

DISPLAY “Exiting”

ELSE

DISPLAY “Invalid Input”

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

END

Registration Page

START

DECLARE Username, Password

DEFINE Register()

DISPLAY “Registration Page”

READ Username “Enter Username: ”

READ Password “Enter Password: ”

OPENFILE “CustomerInfo.txt” FOR READ

READFILE “CustomerInfo.txt”, Username

CLOSEFILE “CustomerInfo.txt”

IF Username IN “CustomerInfo.txt” THEN

OPENFILE “CustomerInfo.txt” FOR READ

DISPLAY “Invalid Sign Up”

CLOSEFILE “CustomerInfo.txt”

CALL FUNCTION Start()

ELSE

OPENFILE “CustomerInfo.txt” FOR APPEND

WRITEFILE “CustomerInfo.txt” Username, Password

CLOSEFILE “CustomerInfo.txt”

DISPLAY “Sign Up Complete”

CALL FUNCTION Start()

ENDIF

END

Menu

START

DECLARE categoryrecord

DEFINE Menu()

DISPLAY “Menu”

DISPLAY “Category”

OPENFILE “MasterCategory.txt” FOR READ

READFILE “MasterCategory.txt”

CLOSEFILE “MasterCategory.txt”

READ categoryrecord

OPENFILE “(categoryrecord) + .txt” FOR READ

READFILE “(categoryrecord) + .txt”

CLOSEFILE “(categoryrecord) + .txt”

END

Guest

START

DECLARE option

DEFINE Guest()

DISPLAY “Guest Page”

DISPLAY “1. View Menu”

DISPLAY “2. Register”

DISPLAY “3. Back”

READ option

IF (option == 1) THEN

CALL FUNCTION Menu()

ELSE

IF (option == 2) THEN

CALL FUNCTION Register()

ELSE

IF (option == 3) THEN

CALL FUNCTION Start()

ENDIF

ENDIF

ENDIF

END

Registered Customer / Customer Page

START

DECLARE option, OrderID, choosecategory, Cart, addmoreitems, Check, Pay

DEFINE RegisteredCust()

DISPLAY “Customer Page”

DISPLAY “1. View Menu”

DISPLAY “2. Order”

DISPLAY “3. Proceed To Payment”

DISPLAY “4. Back”

READ option

IF (option == 1) THEN

CALL FUNCTION Menu()

ELSE

IF (option == 2) THEN

READ OrderID

DEFINE OrderPage()

DISPLAY “ Order Page”

OPENFILE “MasterCategory.txt” FOR READ

READFILE “MasterCategory.txt”

CLOSEFILE “MasterCategory.txt”

READ “(choosecategory) , .txt”

OPENFILE “choosecategory.txt” FOR READ

READFILE “choosecategory.txt”

CLOSEFILE “choosecategory.txt”

READ Cart

IF Cart IN “CustomerOrder.txt” THEN

OPENFILE “CustomerOrder.txt” FOR APPEND

APPENDFILE “CustomerOrder.txt”

DISPLAY “Order Have Been Added To Cart”

ELSE

DISPLAY “Invalid Input”

CALL FUNCTION OrderPage()

ENDIF

READ addmoreitems

IF (addmoreitems == 1) THEN

CALL FUNCTION OrderPage()

ELSE

IF (addmoreitems == 0) THEN

CONTINUE

ELSE

DISPLAY “Invalid Option”

CALL FUNCTION OrderPage()

ENDIF

ENDIF

ELSE

IF (option == 3) THEN

DEFINE Payment()

DISPLAY “Payment Page”

DISPLAY “1.Check Cart”

DISPLAY “2.Check-Out”

DISPLAY “3.Back”

READ option

IF (option == 1) THEN

READ Check “Enter Username”

Display “ These Are Your Food Items”

OPENFILE “CustomerOrder.txt” FOR READ

READFILE “CustomerOrder.txt”

CLOSEFILE “CustomerOrder.txt”

CALL FUNCTION Payment()

ELSE

IF (option == 2) THEN

READ Pay (“Enter Username”)

DISPLAY “Select Payment Method”

DISPLAY “1.Cash”

DISPLAY “2.Card”

DISPLAY “3.Back”

READ option (“Select Option”)

IF (option ==1) THEN

OPENFILE “CustomerOrder.txt’ FOR READ

READFILE “CustomerOrder.txt”

CLOSEFILE “CustomerOrder.txt”

OPENFILE “CustomerPayment.txt” FOR APPEND

APPENDFILE “CustomerPayment.txt”

CLOSEFILE “CustomerPayment.txt”

DISPLAY “You Paid By Cash”

DISPLAY “Thanks For Ordering From SOFS”

ELSE

IF (option ==2) THEN

OPENFILE “CustomerOrder.txt’ FOR READ

READFILE “CustomerOrder.txt”

CLOSEFILE “CustomerOrder.txt”

OPENFILE “CustomerPayment.txt” FOR APPEND

APPENDFILE “CustomerPayment.txt”

CLOSEFILE “CustomerPayment.txt”

DISPLAY “You Paid By Card”

DISPLAY “Thanks For Ordering From SOFS”

ELSE

IF (option ==3) THEN

CALL FUNCTION Payment()

ENDIF

ENDIF

ENDIF

IF (option == 3) THEN

CALL FUNCTION Payment()

ENDIF

ENDIF

ENDIF

IF (option == 4) THEN

CALL FUNCTION Start()

ELSE

CALL FUNCTION RegisteredCust()

ENDIF

ENDIF

ENDIF

ENDIF

END

Customer Login

START

DECLARE Cust\_User, Cust\_Pass

DEFINE Customer()

DISPLAY “Customer Login Page”

READ Cust\_User

READ Cust\_Pass

OPENFILE “CustomerInfo.txt” FOR READ

IF Cust\_User,Cust\_Pass IN “CustomerInfo.txt” THEN

DISPLAY “Login Successful”

CALL FUNCTION RegisteredCust()

ELSE

DISPLAY “Login Failed”

CALL FUNCTION Start()

ENDIF

CLOSEFILE “CustomerInfo.txt”

END

Admin Login

START

DECLARE adminuser, adminpass, adminpage

DEFINE AdminLogin()

OPENFILE “AdminInfo.txt” FOR READ

READ adminuser “Please Enter Your Admin ID:”

READ adminpass “Please Enter Your Password:”

READFILE “AdminInfo.txt”, adminuser, adminpass

IF (adminuser IN “AdminInfo.txt”) AND (adminpass IN “AdminInfo.txt”)

THEN

CALL FUNCTION AdminPage()

ELSE

DISPLAY “Invalid Input”

ENDIF

CLOSEFILE “AdminInfo.txt”

END

Admin Page

START

DECLARE option, exit

DEFINE FUNCTION AdminPage()

DISPLAY “1. Add Food Item Category-wise”

DISPLAY “2. Modify Food”

DISPLAY “3. Display Records”

DISPLAY “4. Search Records”

DISPLAY “5. Exit”

READ option “Choose your Option (1-5)”

IF (option = 1) THEN

CALL FUNCTION AddCategory()

ELSE

IF (option = 2) THEN

CALL FUNCTION ModifyItems()

ELSE

IF (option = 3) THEN

CALL FUNCTION DisplayRecord()

ELSE

IF (option = 4) THEN

CALL FUNCTION SearchRecord()

ELSE

IF (option = 5) THEN

COMPUTE exit

ELSE

DISPLAY “Invalid Option”

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

END

Add Category

START

DECLARE category, additem, itemid, itemname, itemprice

DEFINE FUNCTION AddCategory()

DISPLAY “Enter New Category Name to Create a New Category and Add Items”

DISPLAY “Enter Exiting Category Name to Add New Items”

READ category “Enter Category Name:”

OPENFILE “MasterCategory.txt” FOR READ

IF category IN “MasterCategory.txt” THEN

CLOSEFILE “MasterCategory.txt”

PASS

ELSE

CLOSEFILE “MasterCategory.txt”

OPENFILE “MasterCategory.txt” FOR APPEND

WRITEFILE “MasterCategory.txt”, category

CLOSEFILE “MasterCategory.txt”

ENDIF

OPENFILE “(category) + .txt” FOR APPEND

CLOSEFILE “(category) + .txt”

DISPLAY “Enter 1 to Add Items, 2 to Stop”

READ additem

IF (additem == 1) THEN

WHILE TRUE

READ itemid “Enter Item ID:”

READ itemname “Enter Item Name:”

READ itemprice “Enter Item Price:”

OPENFILE “(category) + .txt” FOR APPEND

WRITEFILE “(category) + .txt”, itemid, itemname, itemprice

CLOSEFILE

ENDWHILE

ELSE

DISPLAY “Stopped”

ENDIF

END

Modify Items

START

DECLARE categorymodify, itemmodify, newele

DEFINE FUNCTION ModifyItems()

OPENFILE “MasterCategory.txt” FOR READ

READFILE “MasterCategory.txt”

CLOSEFILE “MasterCategory.txt”

READ categorymodify

OPENFILE “(categorymodify) + .txt” FOR READ

READFILE “(categorymodify) + .txt”

CLOSEFILE “(categorymodify) + .txt”

READ itemmodify “Enter Item ID to Modify”

IF itemmodify IN “(categorymodify) + .txt” THEN

READ newele

OPENFILE “(categorymodify) + .txt” FOR WRITE

WRITEFILE “(categorymodify) + .txt”, newele

CLOSEFILE “(categorymodify) + .txt”

END

Display Record

START

DECLARE recordopt, choosecategory

DEFINE FUNCTION DisplayRecord()

DISPLAY “1. Food Category”

DISPLAY “2. Food Items Category-wise”

DISPLAY “3. Customer Orders”

DISPLAY “4. Customer Payment”

DISPLAY “5. Exit to Admin Page”

READ recordopt

If recordopt == 1 THEN

OPENFILE “MasterCategory.txt” FOR READ

READFILE “MasterCategory.txt”

CLOSEFILE “MasterCategory.txt”

ELSE

IF recordopt == 2 THEN

OPENFILE “MasterCategory.txt” FOR READ

READFILE “MasterCategory.txt”

CLOSEFILE “MasterCategory.txt”

READ choosecategory “Enter Category Name:”

OPENFILE “(choosecategory) + .txt” FOR READ

READFILE “(choosecategory) + .txt”

CLOSEFILE “(choosecategory) + .txt”

ELSE

IF recordopt == 3 THEN

OPENFILE “CustomerOrder.txt” FOR READ

READFILE “CustomerOrder.txt”

CLOSEFILE “CustomerOreder.txt”

ELSE

IF recordopt == 4 THEN

OPENFILE “CustomePayment.txt” FOR READ

READFILE “CustomerPayment.txt”

CLOSEFILE “CustomerPayment.txt”

ELSE

IF recordopy == 5 THEN

DISPLAY “Exit to Admin Page”

CALL FUNCTION AdminPage()

ELSE

DISPLAY “Invalid Option”

CALL FUNCTION DisplayRecord()

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

END

Search Record

START

DECLARE searchopt, searchorder, searchorder1

DEFINE SearchRecord()

DISPLAY “1. Customer Order”

DISPLAY “2. Customer Payment”

DISPLAY “3. Exit to Admin Page”

READ searchopt

IF searchopt == 1 THEN

READ searchorder “Enter Username”

OPENFILE “CustomeOrder.txt” FOR READ

READFILE “CustomerOrder.txt” searchorder

CLOSEFILE “CustomerOrder.txt”

ELSE

IF searchopt == 2 THEN

READ searchorder1 “Enter Username”

OPENFILE “CustomePayment.txt” FOR READ

READFILE “CustomerPayment.txt” searchorder1

CLOSEFILE “CustomerPayment.txt”

ELSE

IF saerchopt == 3 THEN

DISPLAY “Exit to Admin Page”

CALL FUNCTION AdminPage()

ELSE

DISPLAY “Invalid Option”

CALL FUNCTION SearchRecord()

ENDIF

ENDIF

ENDIF

END

## **Flowcharts**

Start Page

Diagram, engineering drawing

Description automatically generated

Registration Page

Diagram

Description automatically generated

Menu

Diagram

Description automatically generated

Guest

Diagram

Description automatically generated

Registered Customer / Customer Page

Diagram, schematic

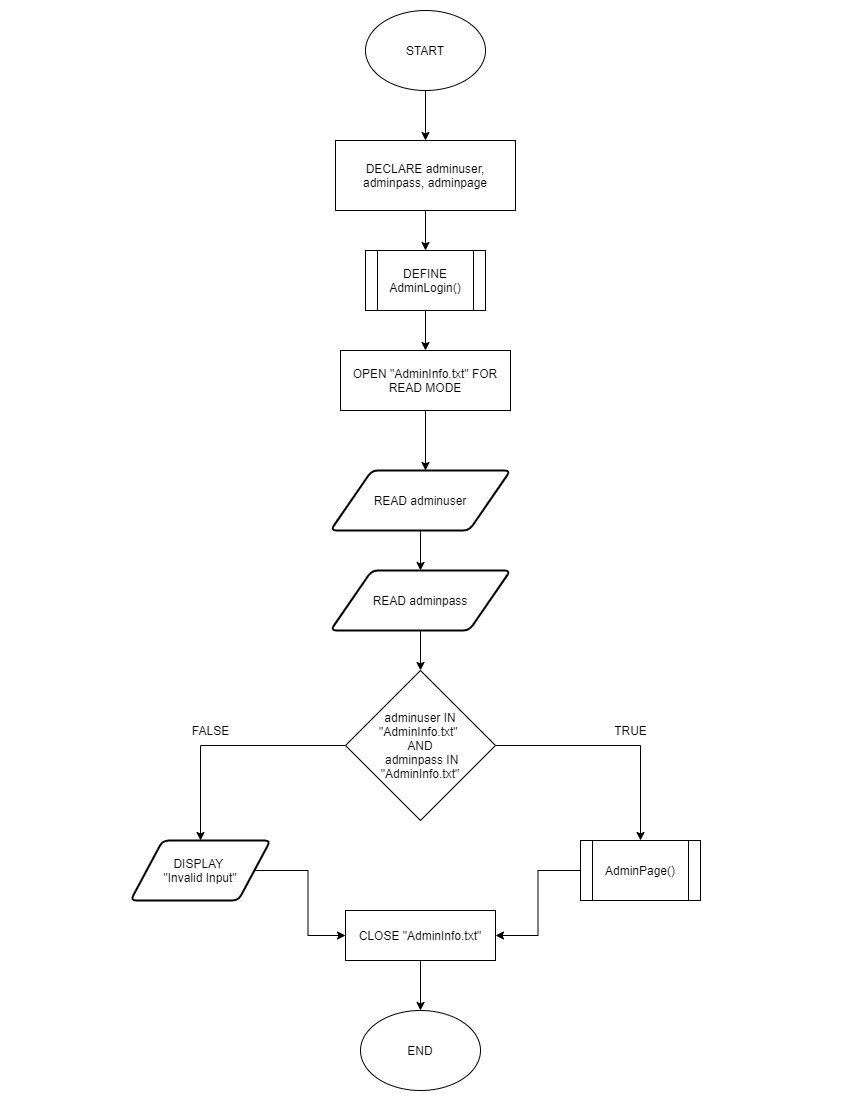
Description automatically generated

Customer Login

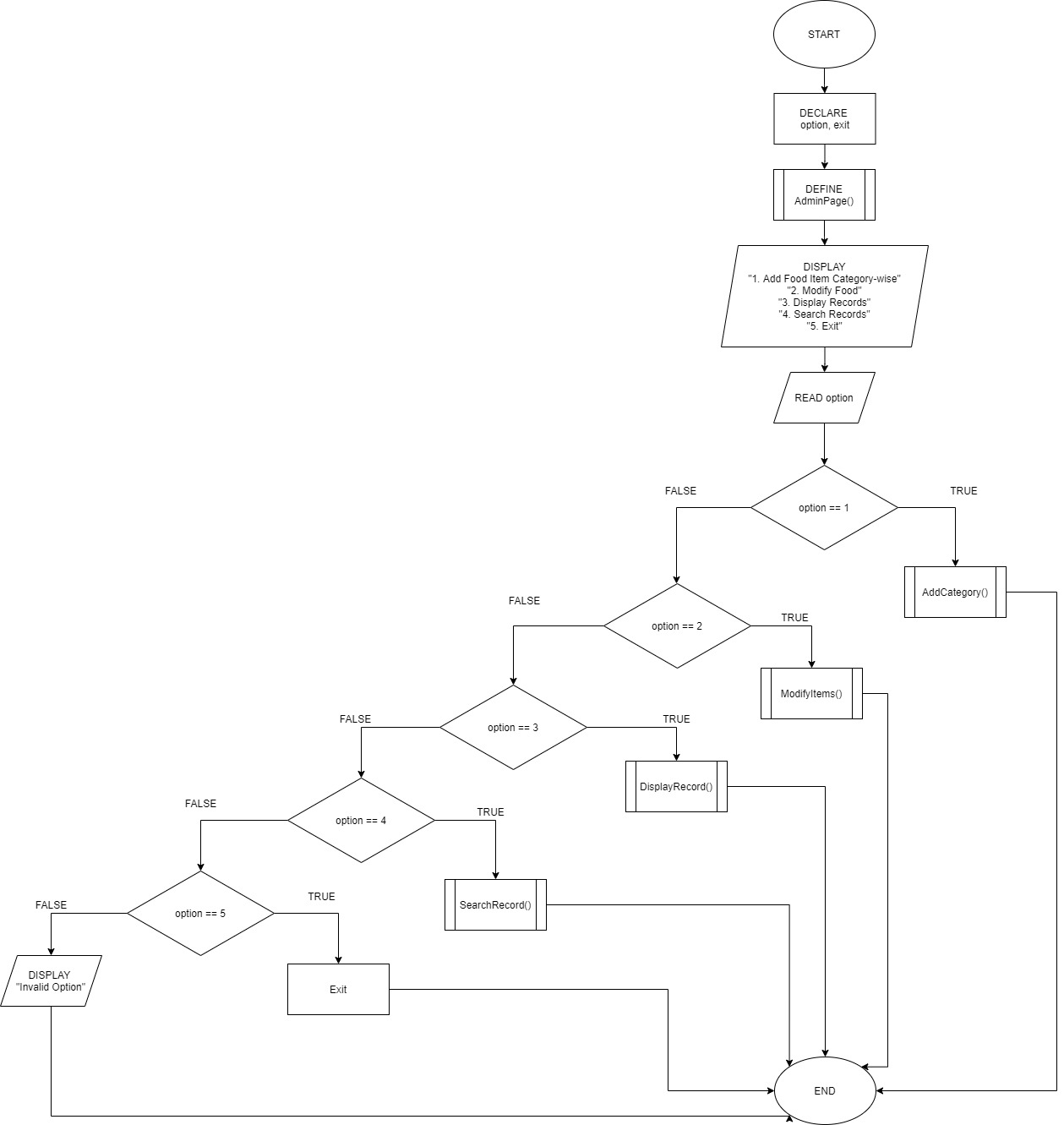
Diagram

Description automatically generated

Admin Login



Admin Page



Add Category

Diagram

Description automatically generated

Modify Items

Diagram

Description automatically generated

Display Record

Diagram, schematic

Description automatically generated

Search Record

Diagram

Description automatically generated

# **Program Source Code with Explanation**

1. **‘input’**

The ‘input’ function allow us to ask the user for inputs. An example is shown below.



In this example, we ask user to input an integer to help us decide what will happen next.

1. **‘def’**

The def is a function which is a block of code that only runes when it is called. An example is shown below.

Text

Description automatically generated

In this example, we use ‘def’ to define the variable ‘Register’. Hence when the code ‘Register()’ is called, the block of code above will be called and process.

1. ‘for’ loop

The ‘for’ loop is used to iterate over a sequences that is either a list, tuple, dictionary, set or a string. An example is shown below.

Text

Description automatically generated

In this example, the program first open its file in read mode as ‘f’, then, we assign a variable as ‘line’. Next, we use the variable ‘line’ to start a ‘for’ loop with the code of for ‘line’ in ‘f’, then, the example above creates another variable as ‘y’ to split the data in the variable ‘line’ as the variable ‘line’ is now the data in the file ‘f’. Lastly, the program then prints the variable ‘y’ which is the data in the file.

1. **File Handling**

In file handling there are many function available, the function in the program we use are

* ‘open()’
* ‘with open’
* ‘.close’
* ‘.read’
* ‘.strip()’
* ‘.split()’
* ‘.write’
* ‘.append’

The ‘open()’ function is a key function to work with files in Python. The ‘open()’ function takes two parameters which are filename and mode. There are 4 different type of modes for opening a file. The modes are as shown: (W3school, 1999)

* ‘r’ – Read = Default value. Opens a file in reading mode, error if file does not exist
* ‘a’ – Append = Opens file for appending, creates a file if file does not exist
* ‘w’ – Write = Opens file for writing, creates a file if it does not exist
* ‘x’ – Create = create a specified file, error if file exist

Additionally, u can also specify if the file will be handled as binary or text mode

* ‘t’ – Text = Default value. Text mode
* ‘b’ – Binary = Binary mode (e.g. images)

Next, the ‘.read’ function is used to read a file that is assigned to it while the ‘.write’ function is used to write into a file that is assigned to it, it will overwrite any existing content in it. Then, the ‘.close’ function is to close the file after it is done using it.

An example of ‘open()’, ’.read’, ’.close’, ‘.write’ is shown below.

Text

Description automatically generated

In this example, variable ‘r’ is assigned to open the file “MasterCategory.txt” in read mode which have been specified with ‘r’ using the ‘open()’ function. Next, a variable is assigned as ‘readmaster’ to read the file using ‘.read’. Furthermore, with the function ‘.write’, the variable ‘s’ is able to write into the file “MasterCategory.txt” in append mode. Lastly, the file “MasterCategory.txt” is close by using the ‘.close’.

Moreover, the ‘.append’ function allow use to write to an existing file, list and other data storage. The .append function will add the data to the end of the storage. An example is shown below.

Text

Description automatically generated

In this example, the user input “itemsid’, ‘itemsname’ and ‘itemsprice’ is appended into a list as ‘item’. The output of the example would be ‘301’ as itemsid, ‘coconut’ as itemsname, ‘5.00’ as itemsprice with the user input ‘301’,’coconut’ and ‘5.00’.

Moving on, the ‘with open()’ function works very similar too the ‘open()’ function. The only different is the ‘with open()’ function doesn’t require a ‘.close’ function to close the file due to it being able to close the file itself after the process is done. An example is shown below.

**Text

Description automatically generated**

In this example, the program opens the file “MasterCategory.txt” in read mode, without use the ‘.close’ function, the ‘with open()’ function is able to close the “MasterCategory.txt” file and proceed to the next process.

Lastly, the ‘.strip()’ and ‘.split()’ function. The ‘.strip()’ removes spaces at the beginning and the end of the string while the ‘.split()’ is a method to splits a string into a list. We can also specify the separator or as default, the separator is any whitespace. An example is shown below.

Text

Description automatically generated

In this example, the file is open and read. ‘MD’ as the variable is being strip and split. The ‘.strip()’ removes any leading spaces at the beginning and the end while ‘.split()’ splits the string into a list with the separator of “;”. An example of before ‘.strip()’’.split()’ and after ‘.strip()’’.split()’ is shown below.

**Before**

Graphical user interface, text

Description automatically generated

**After**

Text

Description automatically generated

**‘print’**

Python ‘print’ function is displaying the given object to the standard output device. An example is shown below.



**Text

Description automatically generated**

In this example, we see that the output is based on the source code.

**‘\n’ & ‘\t’**

The cooperation of a backslash and the letter n is to create a new line at the point where the ‘\n’ is after.

****

**Text

Description automatically generated**

In this example, we can tell that when ‘\n’ is added into a line of code, it will create a new line.

Next, ‘\t’ is kind like same as ‘\n’. Because it is not creating a new line, but it is to indent the output like we use Tab key on the keyboard.

****

****

As shown above we can see that there is some empty space in the output after (Modifying:), which is caused by ‘\t’.

**‘if’ & ‘elif’ & ‘else’**

‘if’ is to make decision between two alternative paths, which is depended on the result of a condition is true of false.

‘elif’ is short form for else if, it is to check multiple expressions, which can simply understand as if the previous conditions were not true, then try this condition.

‘else’ is when the ‘if’ condition is false, then the content of ‘else’ will be executed.

**Text

Description automatically generated**

This example has shown one if, four elif, and one ‘else’ statement. In this case the program will check is first condition satisfied, then is second condition satisfied, and so on, until it gets in touch with the right condition. And if ‘if’ and ‘elif’ is no right for any condition then program will execute the ‘else’ statement.

**‘pass’**

‘pass’ statement is a null statement, and it is used as a placeholder for future code. (W3school, 1999)

A screenshot of a computer

Description automatically generated with medium confidence

In this case, the pass statement is used to allow program to continue for next line of code instead of turning the program off.

**‘while’ & ‘return’ & ‘true’ & ‘false’**

In python, ‘while’ is a loop which is execute a set of statements as long as a condition is true.

‘return’ is hand back a value to its caller when program function exit.

‘true’ and ‘false’ is a Boolean Values, which is evaluate any expression and get one of two answers.

Text

Description automatically generated

In this example, ‘while’ loop will keep execute until the program handed back(‘return’) the value is ‘false’.

**‘continue’**

The purpose of ‘continue’ is to end the current iteration in a loop and continue to next iteration.

**Text

Description automatically generated**

This example shown that when the condition is satisfied, then it will continue to next iteration which is back in the loop(‘while’) again.

**‘.replace()’**

‘.replace()’ is to change the specific character to anything that is specified and returns a copy of string. And the syntax for ‘.replace()’ is ‘string.replace(old, new, count)’. (W3school, 1999)

Text

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

The example shown, there are two different results. And the original string(HelloWorld) has (.txt).

The ‘1’ is without using the ‘.replace()’, therefore the (.txt) still exist.

And ‘2’ is using the ‘.replace()’ to replace the (.txt) to empty (), therefore the result show nothing after the string(HelloWorld).

# **Additional Features Source Code with Explanation**

**‘import os’**

Python have one of the features called Modules; it is a file contained a set of function which like a code library. And the ‘import’ is to bring in the Modules to the code which is importing the definitions of a Module to another Module or the interactive interpreter in Python. (Akanksha\_Rai, 2021)

And ‘os’ is one of the modules, it provided a function that allow to interact with the Operating System information or control processes up to a limit. There are many functions in ‘os’, for example ‘os.name’, ‘os.environ’, ‘os.execvp’, and many more. (Python, 2001)

We have used ‘os.system('pause')’ in this assignment, it is to execute the command (string) in a subshell. Therefore, this function provide program to allow user to press any key to continue whenever user want.

Text

Description automatically generated

Text

Description automatically generated

In this example, we can see there is a message said ‘Press any key to continue . . .’ in the output, it is because ‘os.system('pause')’ . This function provides user to stop the program no matter how long and the program will continue to next action right after when user press any key.

# **Screenshots of Sample Input / Output with Explanation**

**Start Page**

At the start of the program, user is requested to select an option to proceed to the next function. At the main page, there are 5 option to choose from ranging from 1 to 5. If the user input ’1’ into the program, it will show the guest page, if the user input ‘2’, the customer login page will be shown, If the user input ‘3’, the registration will be shown, if the user input ‘4’, it will bring the user to the admin login page and lastly if the user input ‘5’, the program will be terminated. Below are example of the output ranging from option 1 to 5.

*Start Page Option 1 Option 2*

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

*Option 3 Option 4 Option 5*

Text

Description automatically generated

**Guest Page**

At the guest page, there are 3 option to chose from, the first option brings u the menu while the second option takes u to the registration page and lastly the third option brings the user back to the main page. Sample of output are below.

*Option 1 Option 2 Option 3*

**Text

Description automatically generated**Text

Description automatically generated

At the menu, which is option 1, user is able to choose different category of item that are shown, the example below shows when the category name Pizza is inputted by the user.

Input ‘Pizza’

**Text

Description automatically generated**

After finish viewing the menu, the user is then sent back to the main page.

**Registration Page**

At the registration page, the user is require to register in order to access the ordering system.

*Registration Page*



*Sample of register*

***Text

Description automatically generated***

*Output stored in file*

*Text

Description automatically generated*

After a successful sign up, the user is then able to login to access more system.

**Customer Login Page/ Customer Page**

At the customer login page, the user is required to login first. If the input matches the data in the file, the customer is then able to access the system.

*Customer Login*

**Text

Description automatically generated**

After login, there are 4 options, the first is to view menu, second is to order, third is to proceed to payment and lastly forth is to return to the main page. The view menu is the same as the guest menu.

*Option 1: View Menu Option 2: Order Option 3: Payment*

**Text

Description automatically generated**A picture containing text

Description automatically generatedText

Description automatically generated

Next, at the order page, user is to choose from the category available to order their foods. An example of ‘Pizza’ is shown below.

**Text

Description automatically generated**

The user then is able to choose from the Pizza menu and add to cart.

*Add To Cart*

Text

Description automatically generated

Lastly, the payment page. There are 3 options, first is to check order cart, second is to check out and lastly third is to return to order page.

Check Cart

Text

Description automatically generated

During the check out, u is required to choose from payment with card or cash.

Text

Description automatically generated

Once user chose a type of payment, the program will then end.

Payment By Cash/Card

Text

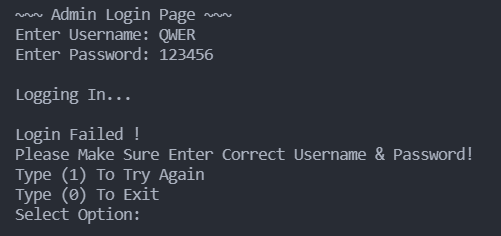
Description automatically generated

**Admin Login Page**

Admin Login Page is allowed whoever have enter the right Admin Username and Admin Password to login.

Enter the correct Username and Password --- Enter the incorrect Username or Password

Text

Description automatically generated --- 

If user enter right for both username and password, then the program will allow user access the next page. But if user enter wrong one of the username or passwords, then program will not allow user to access and the program will ask user to decide to try again or exit. It is a loop; program will not process to next step until user enter right username and password or choose option to exit.

**Admin Page**

After user logged in, there are five options for user to choose. Each option has different functions.

Text

Description automatically generated

A.Option 1: After user choose option 1. Add Food Item Category-wise. It will require user to enter Category Name, user can enter a new category name if want to create a new category, or user can enter an existing category name to add new items.

Text

Description automatically generated

In this case, user have entered a new category name ‘Burger’. And the program will create a new text file by naming as ‘Burger.txt’ and program will save this name (Burger) into a Master Category text file which store all the category name. If user want to add item in this category just simply enter ‘1’, or user can enter ‘0’ for no add item.

Text

Description automatically generated

If user choose to add item, program will ask user to enter the data (Item ID, Item Name, Item Price). And program will store data into text file. Next, user can choose to add more items or not. It is a loop and will stop until user enter ‘0’.

Text

Description automatically generated

A.Option 2: Program will show user category name and require user to enter the existing category name.

Text

Description automatically generated

After enter category name, program will require user to choose which items want to modify and show the item detail. After user modify the item, system will save the new data into text file and ask user want to modify again or not.

Text

Description automatically generated Text

Description automatically generated

A.Option 3: User can choose what record they want to check by enter 1 to 5.

Text

Description automatically generated

Opt1: Show all existing category.

Text

Description automatically generated

Opt2: Require user to enter category name and show all the item inside the category.

A picture containing text

Description automatically generated

Opt3: Show all the customer orders

Text

Description automatically generated

Opt4: Show the detail of customer payment each time they order.

Text

Description automatically generated

Opt5: Exit back to Admin Page

Text

Description automatically generated

A.Option 4: User can search specific records

Text

Description automatically generated

Opt1: User can check the record of the order after entering the order username.

A picture containing table

Description automatically generated

Opt2: User can check the record of the payment after entering the order ID.

Timeline

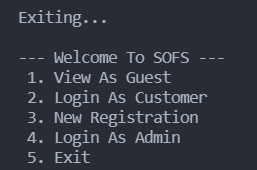
Description automatically generated

Opt3: Exit to Admin Page

Text

Description automatically generated

A.Option5: Exit to Start Page



# **Conclusion**

To conclude this documentation, we have created and designed a system for SOPS to enhances their system. The system allows users with access to view, modify, add and many more features. Moreover, we provided some relevant information such as pseudocode and flowchart as for the design of the program. Next, we also explained the source code and its additional features while also giving some examples of input and output with provided explanation. With the program we created, the order system now provides various additionally features and the process of execution is also smooth and neat.

# **References**

Akanksha\_Rai, 2021. *GeeksForGeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/os-module-python-examples/  
[Accessed 6 8 2021].

Python, 2001. *Python.* [Online]   
Available at: https://docs.python.org/3/tutorial/modules.html  
[Accessed 9 8 2021].

W3school, 1999. *W3school.* [Online]   
Available at: https://www.w3schools.com/python/python\_file\_handling.asp  
[Accessed 8 8 2021].

W3school, 1999. *W3school.* [Online]   
Available at: https://www.w3schools.com/python/ref\_string\_replace.asp  
[Accessed 8 8 2021].

W3school, 1999. *W3school.* [Online]   
Available at: https://www.w3schools.com/python/ref\_keyword\_pass.asp  
[Accessed 8 8 2021].