**Task 2**

**“I confirm that this assignment is my own work. Where I/we have referred to academic sources, I have provided in-text citations and included the sources in the final reference list. “**

**Module Code : FC712**

**Class/Group: OFC Comp 1**

**Module Title: Programming**

**Assessment Title: Project**

**Tutor Name: Sophie Norman**

**Student GUID Number: P299883**

**Date of Submission: 11th July 2021**

**Task 2 – Hotel Management**

**Section 2.1: Algorithm**

The program is divided into 6 python files, 1 dat file and 1 txt file.

The 6 python files are main.py, checkin\_gui\_and\_program.py, checkuoutgui.py, getinfoui.py, listgui.py and receipt.py.

**Main.py -**

1. Import libraries/modules
2. Define different functions which call upon respective python files when being called
3. Define a class called HOTEL\_MANAGEMENT
4. Define an init function under the class where we design and configure the window or the panel using tkinter

**Checkin\_gui\_and\_program.py -**

1. Import libraries
2. Define a new variable details\_list and initialize it to be an empty list
3. Define a new function file\_save which will save information of the guest from details\_list
   1. Opens the hotel.dat file , dumps (saves) all the information on guest in the hotel.dat file
   2. Opens recipe.txt and writes the information on guest there as well
   3. Calls the recit.py program
4. Initialise new tuples as different room types in which tuples with room numbers for each type of room is being stored
5. Define a class which would save all the temporary details on guest which would in turn make it permanent
6. Define a class HOTEL\_MANAGMENT\_checkin
   1. In init initialize all the details variable to 0/””/null
   2. Define multiple functions to check if each detail is valid or not.
   3. Define a function for calculation of bill and payment options
   4. we design and configure the window or the panel using tkinter

**Checkoutgui.py -**

1. Import libraries/modules
2. Define a new function file\_save which will save information of the guest from details\_list
   1. Opens the hotel.dat file, dumps (saves) all the information on guest in the hotel.dat file
   2. Opens recipe.txt and writes the information on guest there as well
   3. Calls the recit.py program
3. Define a function for restarting the program
4. Define a new class New\_Toplevel
   1. Define init
      1. Define a new function checkroom which if the room is valid remove the data of the guest from the database of the program which technically does check out for the guest
      2. Design and configure the window or the panel using tkinter

**Getinfoui.py -**

1. Import libraries/modules
2. Define a new function file\_save which will save information of the guest from details\_list
   1. Opens the hotel.dat file , dumps (saves) all the information on guest in the hotel.dat file
   2. Opens recipe.txt and writes the information on guest there as well
   3. Calls the recit.py program
3. Define a function for restarting the program
4. Define a class which would save all the temporary details on guest which would in turn make it permanent
5. Define a class HOTEL\_MANAGEMENT
   1. Define init
      1. Define a function gotinfo
         1. Take in the room number and check fs the room number is valid
         2. Open the database and print the details of the guest staying in the room
      2. Design and configure the window or the panel using tkinter

**Listgui.py -**

1. Import libraries/modules
2. Define a new function file\_save which will save information of the guest from details\_list
   1. Opens the hotel.dat file , dumps (saves) all the information on guest in the hotel.dat file
   2. Opens recipe.txt and writes the information on guest there as well
   3. Calls the recit.py program
3. Define a function for restarting the program
4. Define a class which would save all the temporary details on guest which would in turn make it permanent
5. Define a new class HOTEL\_MANAGEMENT\_checkin
   1. Define init
      1. Design and configure the window or the panel using tkinter
6. Open hotel. Dat file
   1. List all the guests staying and their respective room numbers
   2. Close the dat file
   3. Call the HOTEL\_MANAGEMENT\_checkin class

**Recipt.py -**

1. Import Libraries/modules
2. Open receipt.txt file
3. Store the content of the text file in a new list
4. Delete elements in index from 1-5
5. In list1 form 0 to 4th index slice the elements in such a way that last character is deleted
6. Store the text for receipt in a string  with all the details of guests as well as the total bill
7. Define a class receipt
   1. Define init
      1. Design and configure the window or the panel using tkinter
8. Call the receipt class

**Section 2.2 - Technical Overview**

**Key variables -**

1. **Checkin\_gui\_and\_program.py**

* Details\_list : list storing the information on guests
* Different tuples for different room types containing room numbers in form of a tuple

1. **Checkoutgui.py**

* Details\_list : list storing the information on guests

1. **Getinfoui.py**

* Details\_list : list storing the information on guests
* L2 and G = empty lists used for temporary storing purpose

1. **Listgui.py**

* Details\_list : list storing the information on guests
* L2 and G = empty lists used for temporary storing purpose

1. **Receipt.py**

* List1:  for storing content of receipt.txt
* P: String storing text to be displayed on the ticket

**Key Functions and Classes**

* Function file\_save being used in almost all the python files, this function saves the data of the guest into the hotel database which is the hotel.dat file
* Function restart\_program which is being used a lot, this function basically restarts the program
* Class Save, a class which would save all the temporary details on guest which would in turn make it permanent
* Class HOTEL\_MANAGMENT, HOTEL\_MANAGEMENT\_checkin, HOTEL\_MANAGEMENT\_checkout, New\_Toplevel: in these classes the layout, configuration and designing of the window/panel takes place using tkinter

**Section 2.3: Testing**

For testing the program, I ran through the program from the point of view of the guest. I simultaneously kept running the program side by side as I was building it, to check whether or not my coding works. I also gave invalid inputs to make sure that the program can handle all the cases. After creating the program, I ran it countless times to check whether all the functions work properly.

Below are the screenshots of the program:

Graphical user interface, application, Word

Description automatically generatedA picture containing text, screenshot, monitor

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Section 2.4: Summary**

This program is a hotel management system used for checking in, check out and many other functions.

Upon opening the program, the user is presented with the main menu where the user has the option to choose what he wants to perform.

Upon clicking check in, the user is given a form where they must fill in the details about themselves and choose the kind of room they want. At the end of the check in the user is given the final bill which also mentions the room number.

Upon clicking the show guest list, the user is shown all the guests and their respective room numbers.

Upon clicking the Checkout option, the user is asked to enter the guest’s room numbers and upon being entered the valid room number the guest’s details is removed from the database which can be checked by clicking the show guest list option.

Upon clicking the get info on any guest button, the user is asked to enter a room number, and all the details about the guest staying in that room is printed in the terminal

Upon clicking the Exit button, the program is closed.

According to me this is a pretty good solution for the hotel management system, it is simple and easy to use. Though it is simple that can be a disadvantage as well, so i think in terms of improvements I could have designed the program much better in terms of the looks and could have added more functions.

**References**

<https://stackoverflow.com/questions/14907067/how-do-i-restart-a-program-based-on-user-input>

<https://stackoverflow.com/questions/13070065/creating-buttons-with-python-gui>

<https://www.programmersought.com/article/97748006413/>

https://www.kite.com/python/docs/tkinter.ttk