Section 3 - Design

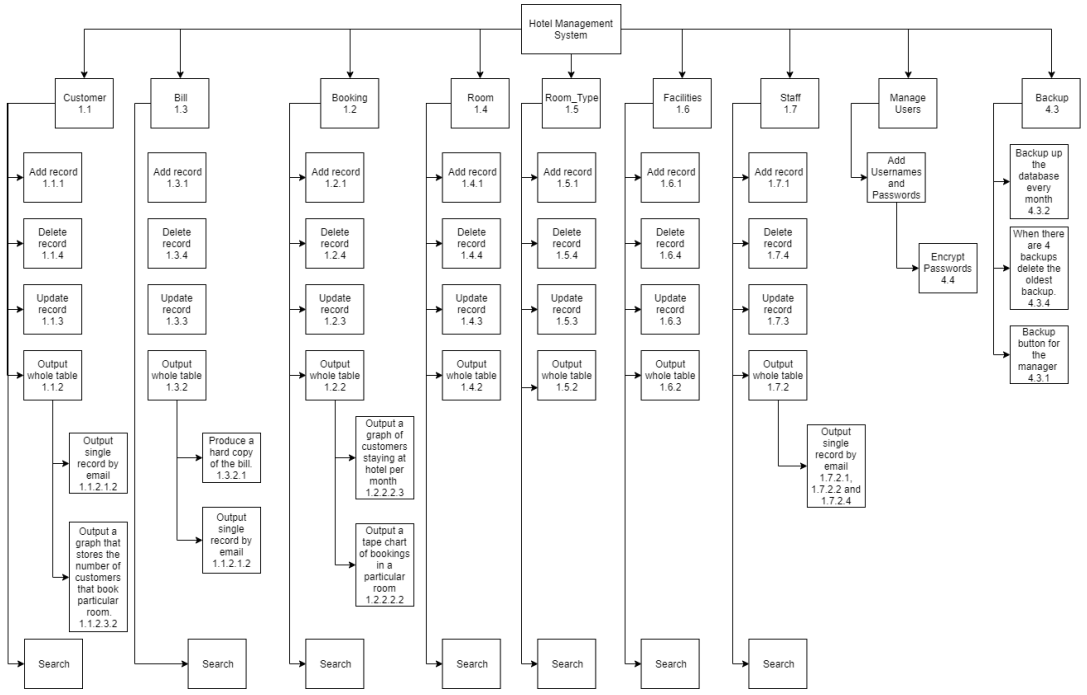
Table of contents:

Breaking down of the problem

Decomposition will apply to my solution as I plan to split my solution into different scripts and within these scripts and within these scripts break it down further into modules. An example of this is that there will be a seperate script for each window my program will have. The login window will be created within a different script than the main window. This will make testing the system easier as I can just test that script instead of the whole system.

I will also be creating many user-defined libraries in an attempt to reuse my code. Another benefit of user-defined subroutines is that they can be tested separately and therefore debugging the code will be easier. Many of my user-defined subroutines will be used in data handling. There will be a delete, add, search and edit subroutines as these will be used many times. Another user-defined subroutine which I will create is one in which it refreshes the tab that is open. This will be used in other subroutines as any changes made within the database needs to be changed in the output. I will also create a subroutine that will create entry widgets and use this in the creation of the GUI. The reason for the subroutine to create the entry widgets is because there will be many entry widgets in my system so this code will be reused many times. As every time the user wants to add or delete a record entry widgets must be created and this therefore will be used many times. The process of encryption and decryption will also be user-defined subroutines as any time data that needs to be encrypted this subroutine can be called and any time data needs to be decrypted the decryption subroutine can be called.

System overview:



Data dictionaries:

Customer:

Field	Data Type	Data Format	Description	Example	Validation
CustomerID	Integer		This is the primary key for this table.	1973	It should check if the data already exists in another field.
Telehone_number	Integer		This is the phone number that belongs to the customer.	075567826912	This will be a length check as all phone numbers have the same length.
Frst_name	String		This will be the first name of the customer.	Adam	This will be checked to see if it contains only alphabetic characters.
Surname	String		This will be the surname of the customer.	Logan	This will be checked to see if it contains only alphabetic characters.
BookingID	Integer		This is a forien key linked to the Booking table.	546	This will be checked if it exists within the Booking table.
BillID	Integer		This is a forien key linked to the Bill table.	245624	This will be checked if it exists within the Bill table.
Address	String	'^\d+\s[A-z]+\s[A-z]+$'	This will be the home address of the customer.	16 Victoria Road	This will be checked to see if it fits the format mentioned in the 'Data Format' column.
Postcode	String	"^(([A-Z][A-Z]{0,1})([0-9][A-Z0-9]{0,1})) {0,}(([0-9])([A-Z]{2}))$". This will validate all UK postcodes with or without spaces but the letters will need to be upper case.	This will be the postcode of the customer.	BT78 9HJ	This will be checked to see if it fits the format mentioned in the 'Data Format' column.
Payment_type	String		This is how the customer is paying for their stay.	MaterCard	This field will be validated using a lookup check as there are only a certain amount of payment types the hotel can accept.

Card_Number	Integer		This will be the card number of the customer.	1111 1111 1111 1111	This will be checked using a type check as it will be checked if it is an integer.
Expiry_date	String (via the datetime library)	MM/YY	This is the expiry date of the customer's debit/credit card.	8/9/21	This will be checked to see if it fits the format mentioned in the 'Data Format' column. This will also be checked if the date is in the future.
CVC-code	Integer		This is the CVC-code of the customer's card.	134	This will be checked using a type check as it will be checked if it is an integer.
City	String		This is the town/city/village where the customer lives.	Carrickfergus	A presence check will be used as the user will have entered a street name and therefore lives in a town, city or village.
Age	Integer		This will be the age of the customer.	38	This field will need to have a type check form of validation to check if it is an integer
Allergies	String		This will be a list of allergies the customer has.	nuts, shellfish	This will be a lookup check that will check a list of known allergies '["milk","eggs", "nuts","fish","shellfish","fruit","Soy"]'
email	"(^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$)". This format will simply check if a character has been entered before and after the '@' symbol and before and after the '.'.		This is the email of the customer	customer1@gmail.com	This field will have a format check to check if it is the correct format for an email.
Username	String		This will be the username of the customer which they will use to log onto the system.	alogan185	This will be checked to see if it matches any other usernames in the system.
Password	String		This will be the password of the customer which they will use to log onto the system.	Password123!	This field will be checked if it is above a certain amount of characters, if it contains uppercase

					and lowercase characters and if it contains alphanumeric characters.

Booking:

Field	Data type	Data format	Description	Example	Validation
BookingID	Integer		This is the primary key of the table.	123	This will be checked to make sure that the 'BookingID' entered does not already exist.
Number_of_guests	Integer		This is the number of guests per booking	5	The 'number_of_guests' field will need to have a type check form of validation to check if it is an integer.
Check_in	String (via the datetime library)	dd/mm/yy	This is the date at which the customer 'checks in'	6/8/19	This will have a format check to check if they are in the correct format for a date.
Check_out	String (via the datetime library)	dd/mm/yy	This is the date at which the customer 'checks out'	14/8/19	This will have a format check to check if they are in the correct format for a date.
RoomID	Integer		This is a forien key linked to the Room table.	441	This will be checked if it exists within the Room table.
Holiday_Type	String		This stores whether the guest has booked an all inclusive stay or half board.	half board	This will have a lookup check. The options will be 'all inclusive', 'half-board','full-board','Room Only'
Price	Float		This is how much the customer is being charged for their booking.	199.99	This field will be validated using a type check to check if it is a floating point or not
Date_of_booking	String (via the datetime library)	dd/mm/yy	This is the date the customer booked the room.	5/5/19	This will have a format check to check if they are in the correct format for a date.

Offer_code	String		This will be a code provided by the hotel that will allow the customer to get a discount of some sort.	658HFF9P6	A lookup check will be used to check if the offer code is valid.

Bill:

Field	Data type	Data format	Description	Example	Validation
BillID	Integer		This is the primary key of the table.	2354	This will be checked to make sure that the 'BillID' entered does not already exist.
Booking_ID	Integer		This is a forien key linked to the booking table.	143643	This will be checked if it exists within the Booking table.
Main_bar_bill	Float		This is the total amount of money spent by the customer at the main bar.	80.00	This will have a type check as it is a currency and therefore has to be a float data type.
Mini_bar_bill	Float		This is the total amount of money spent by the customer using the mini bar.	9.99	This will have a type check as it is a currency and therefore has to be a float data type.
Restaurant_bill	Float		This is the total amount of money spent by the customer at the restaurant during their stay.	190.00	This will have a type check as it is a currency and therefore has to be a float data type.
Telephone_bill	Float		This is the total amount of money spent by the customer using the telephone at their stay.	30.00	This will have a type check as it is a currency and therefore has to be a float data type.
Spa_bill	Float		This is the total amount of money spent by the customer at the spa during their stay.	101.00	This will have a type check as it is a currency and therefore has to be a float data type.
Room_service	Float		This is the total amount of money spent by the customer using	86.00	This will have a type check as it is a currency and therefore has to

			room service during their stay.		be a float data type.
Deposit_paid	Boolean		This is to see if the customer has paid the deposit.	Y	This will have a type check it will be a boolean data type.
CustomerID	Integer		This is a forien key linked to the Customer table.	431643	This will be checked if it exists within the Customer table.
Gym_fee	Float		This is the fee of the gym at the hotel.	40.00	This will have a type check as it is a currency and therefore has to be a float data type.
Total_price	Float		This is the total price of the holiday.	901.30	This will have a type check as it is a currency and therefore has to be a float data type.

Room:

Field	**Data type**	**Data format**	**Description**	**Example**	**Validation**
RoomID	Integer		This is the primary key of the table.	34153	This field must be checked if it is already exists within the table.
Room_Number	Integer		This is the number of the room.	111	This will be checked to see if it is an integer.
Price_of_room	Float		This is the price of the room per night.	400.00	This will be checked to see if it is a float.
Room_TypeID	Integer		This is a forien key linked to the Room_Type table.	43532	This will be checked if it exists within the Room_Type table.
Occupied	Boolean		This value will say if the room is occupied or not.	N	This will be checked to see if it is a boolean value.
Number_of_beds	Integer		This is the number of beds the room contains.	3	This will be checked to see if it is an integer.
Room/FacilitiesID	Integer		This is a forien key linked to the Room/Facilities	155	This will be checked if it exists within the

			table.		Room/Facilities table.
Floor	Integer		This is the floor number of the room.	2	This will be checked to see if it is an integer.
View	String		This will store the type of view the room has.	Ocean	This will have a lookup check. The list that will be used is 'Ocean','City','Garden' and 'Street'.
Maintained	Boolean		This will say if the room has been maintained i.e. has it been cleaned?,is everything working?	Y	This will be checked to see if it is a boolean value.
Staff/RoomID	Integer		This is a forien key linked to the StaffID/RoomID table.	2144	This will be checked if it exists within the StaffID/RoomID table.

Room Type:

Field	**Data Type**	**Data Format**	**Description**	**Example**	**Validation**
Room TypeID	Integer		This is the primary key of the table	2135134	This field will need to be checked to see if it already exists as this is the primary key.
Type_of_suite	String		This is the suite that the room is.	Presidential suite.	This will have a presence check as something will need to be entered.
Balcony	Boolean		This will store whether or not the room contains a balcony.	Y	This will have a type check it will be a boolean data type.
Pets	Boolean		This will store whether or not pets are allowed in the room.	N	This will have a type check it will be a boolean data type.
Price	Float		This will store the price of this type of room.	495.00	This will have a type check it will be a float data type.

Staff:

Field	Data Type	Data Format	Description	Example	Validation
StaffID	Integer		This is the primary key of the table.	325	This field will be checked to see if it already exists within the table.
Salary	Float		This is the salary of the member of staff.	30,000.00	This field will be checked to see if it is a float.
Frst_name	String		This is the first name of the member of staff.	Bob	This will be checked if they contain non-alpha-numeric characters
Surname	String		This is the surname of the member of staff.	Logan	This will be checked if they contain non-alpha-numeric characters
Email	String	“(^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$)” This format will simply check if a character has been entered before and after the ‘@’ symbol and before and after the ‘.’.	This is the work email of the member of staff.	bob1@gmail.com	This field will have a format check as it will have a specific format.
Postcode	String	‘^[A-Z]{1,2}[0-9][A-Z0-9]??[0-9][A-Z]{2}$’.This will validate all UK postcodes with or without spaces but the letters will need to be upper case.	This is the postcode of the member of staff.	NJ98 9IJ	This field will have a format check as it will have a specific format.
DOB	String (via the datetime library)	dd/mm/yyyy	This is the member of staff’s date of birth.	2/9/1990	This field will have a format check as it will have a specific format.
contact_number	Integer		This is the phone number that is used to contact the member of staff.	078926374638	This field will be a length check as a phone number can only be a 12 characters long.
Username	String		This is the username of the member of staff.	bob1	This will be checked to see if it matches any other usernames in the system.

Password	String		This is the password of the member of staff.	P@55word	This field will be checked if it is above a certain amount of characters, if it contains uppercase and lowercase characters and if it contains alphanumeric characters.
Job	String		This is the job which the member of staff at the hotel has.	Receptionist	This field will have a lookup check to check if the job exists within the hotel.
Access level	Integer		This is the access level the member of staff has.	3	This field will have a presence check to see if anything has been entered as all employees should have an access level.

Facilities:

Field	Data Type	Data Format	Description	Example	Validation
FacilitiesID	Integer		This is the primary key of the table	210972	This will be checked if it already exists within the table.
Minibar	Boolean		This value will say whether or not the room has a minibar.	Y	This will be checked to see if it is a boolean value.
Wi-Fi	Boolean		This value will say whether or not the room has Wi-Fi.	N	This will be checked to see if it is a boolean value.
Shower	Boolean		This value will say whether or not the room has a shower.	Y	This will be checked to see if it is a boolean value.
Safe deposit box	Boolean		This value will say whether or not the room has a safe deposit box.	Y	This will be checked to see if it is a boolean value.
Air-conditioning	Boolean		This value will say whether or not the room has air-conditioning	N	This will be checked to see if it is a boolean value.

Staff/Room:

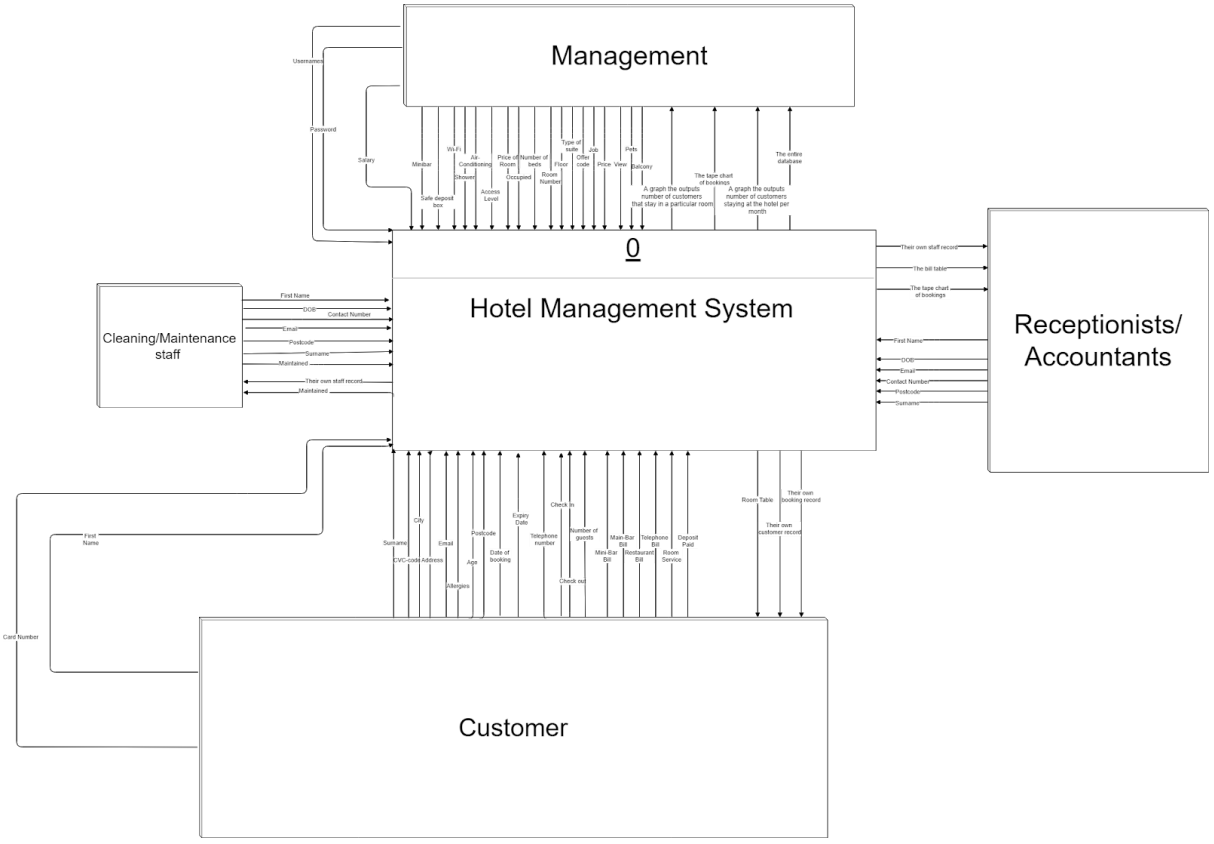
Field	Data type	Data format	Description	Example	Validation
Staff/RoomID	Integer		This is the primary key of the table.	9184376	This will be checked if it already exists within the table.
StaffID	Integer		This is a forien key linked to the staff table.	2476	This will be checked if it exists within the staff table.
RoomID	Integer		This is a forien key linked to the room table.	5427247	This will be checked if it exists within the staff table.

Room/Facilities

Field	Data type	Data format	Description	Example	Validation
Room/FacilitiesID	Integer		This is the primary key of the table.	92387983	This will be checked if it already exists within the table.
FacilitiesID	Integer		This is a forien key linked to the Facilities table.	29874198	This will be checked if it exists within the Facilities table.
RoomID	Integer		This is a forien key linked to the room table.	94729	This will be checked if it exists within the Facilities table.

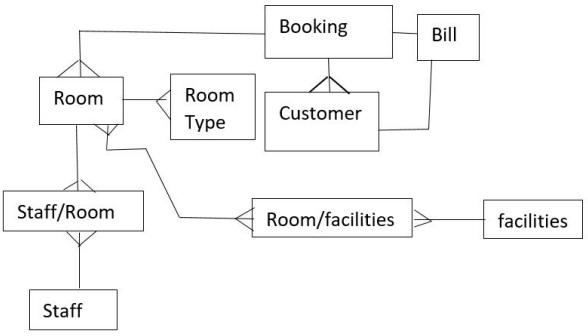
Data Flow Diagrams

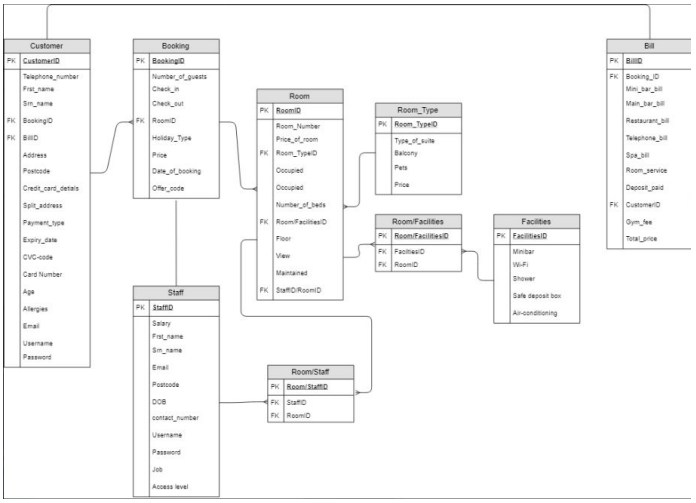
Level 0 DFD:



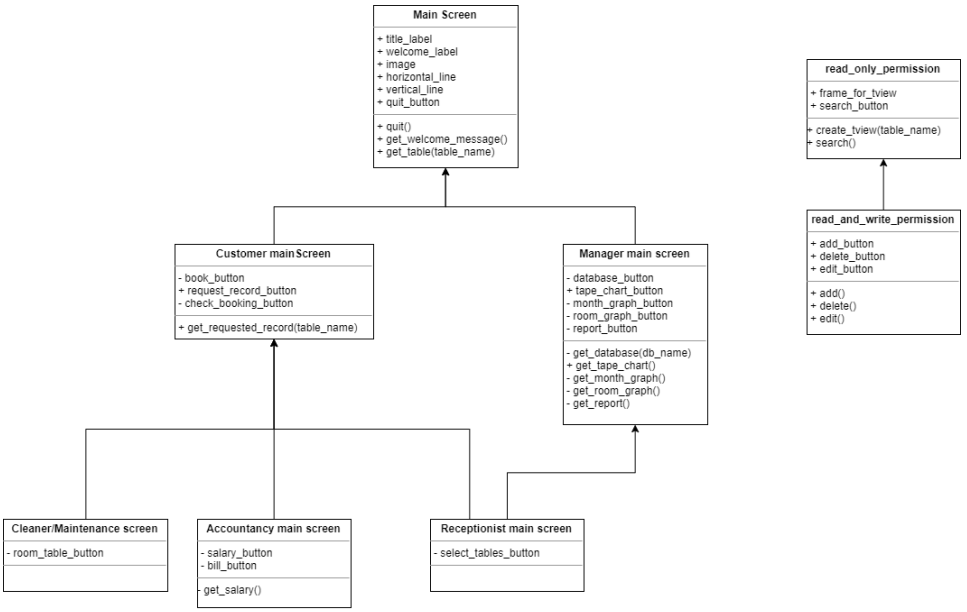
Data Structures

ER diagram:





Class diagrams:



Pseudocode

Valadiation:

```
IMPORT re
PROCEDURE RangeCheck(data, upper_boundary, lower_boundary)
        IF data < upper_boundary THEN
        IF data > lower_boundary THEN
                OUTPUT  “Valid”
                RETURN data
        ELSE
                OUTPUT “Not valid”
ELSE
        OUTPUT “Not valid”
END IF
END PROCEDURE
PROCEDURE PresenceCheck(data)
        IF data== “”  THEN
                OUTPUT “Not Valid”
        ELSE
                OUTPUT “Valid”
                RETURN data
END PROCEDURE

PROCEDURE LengthCheck(data, length_to_check)
        IF length(data) == length_to_check THEN
                OUTPUT “Valid”
                RETURN data
        ELSE
                OUTPUT “Not valid”
END PROCEDURE

PROCEDURE TypeCheck(data, data_type)
        IF type(data) == data_type
                OUTPUT “valid”
                RETURN data
        ELSE
                OUTPUT “Not valid”
END PROCEDURE

PROCEDURE LookupCheck(data, data_to_compare)
        DECLARE valid as BOOLEAN
        valid=FALSE
        FOR single_data in data_to_compare DO
```

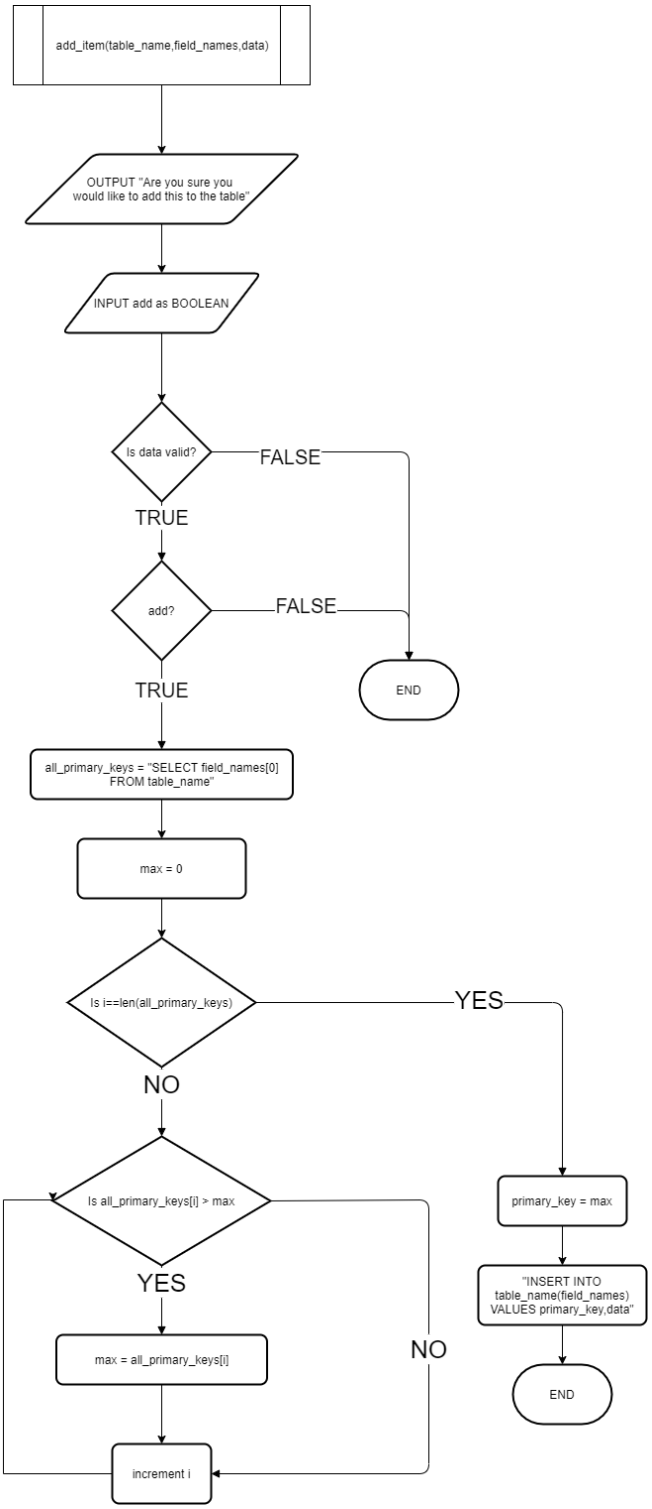
```
                IF data == data_to_compare
                        valid = TRUE
        END LOOP
        IF valid == TRUE THEN
                OUTPUT “valid”
RETURN data
        ELSE
                OUTPUT “Not valid”
END PROCEDURE

PROCEDURE FormatCheck (data,format)
        IF re.match(data,format) == TRUE
                OUTPUT “valid”
                RETURN data
        ELSE
                OUTPUT “Not valid”
END PROCEDURE
```

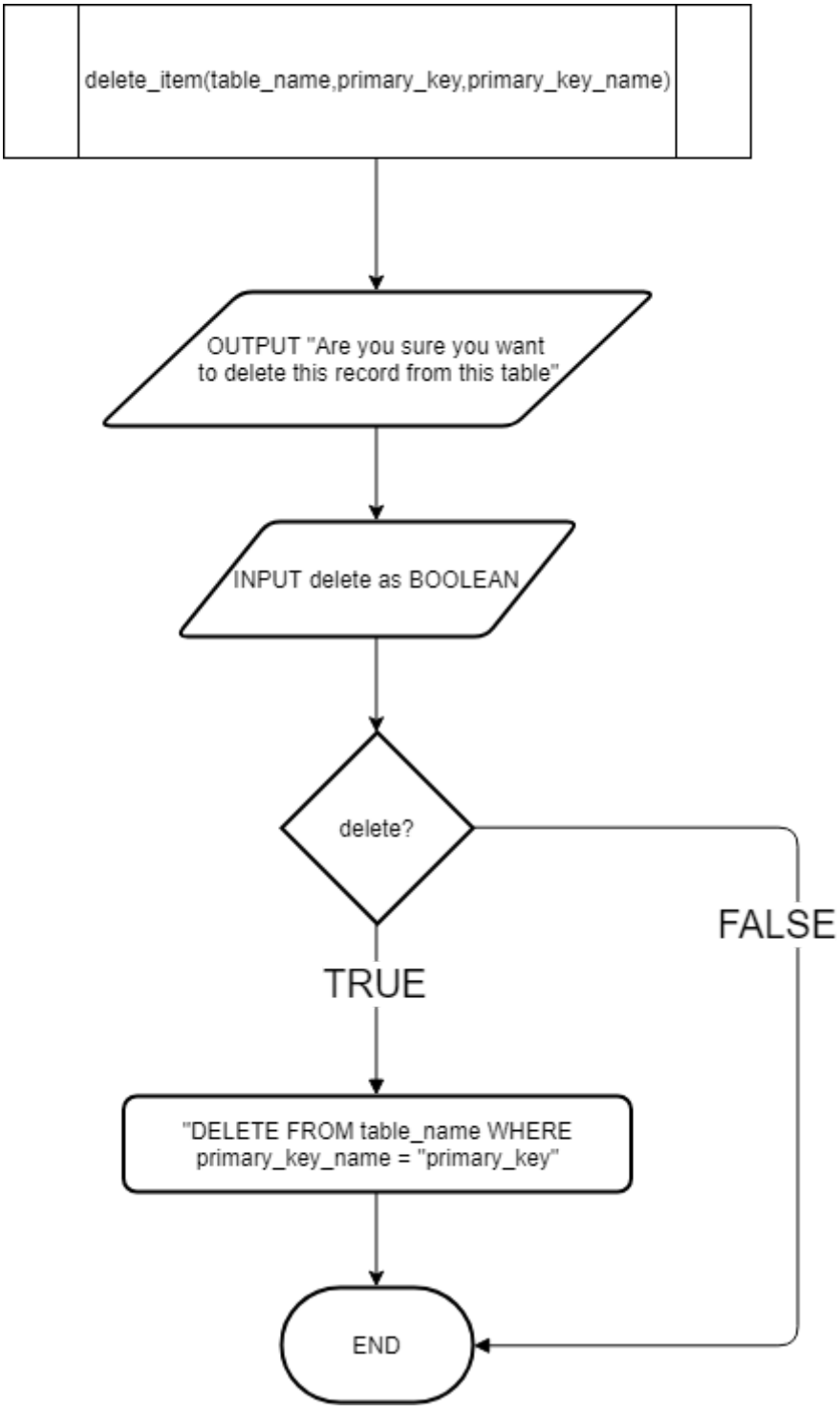
User-defined Subroutines

Add item

The below flowchart works on the assumption that the user does not enter the primary key of the record they are adding.

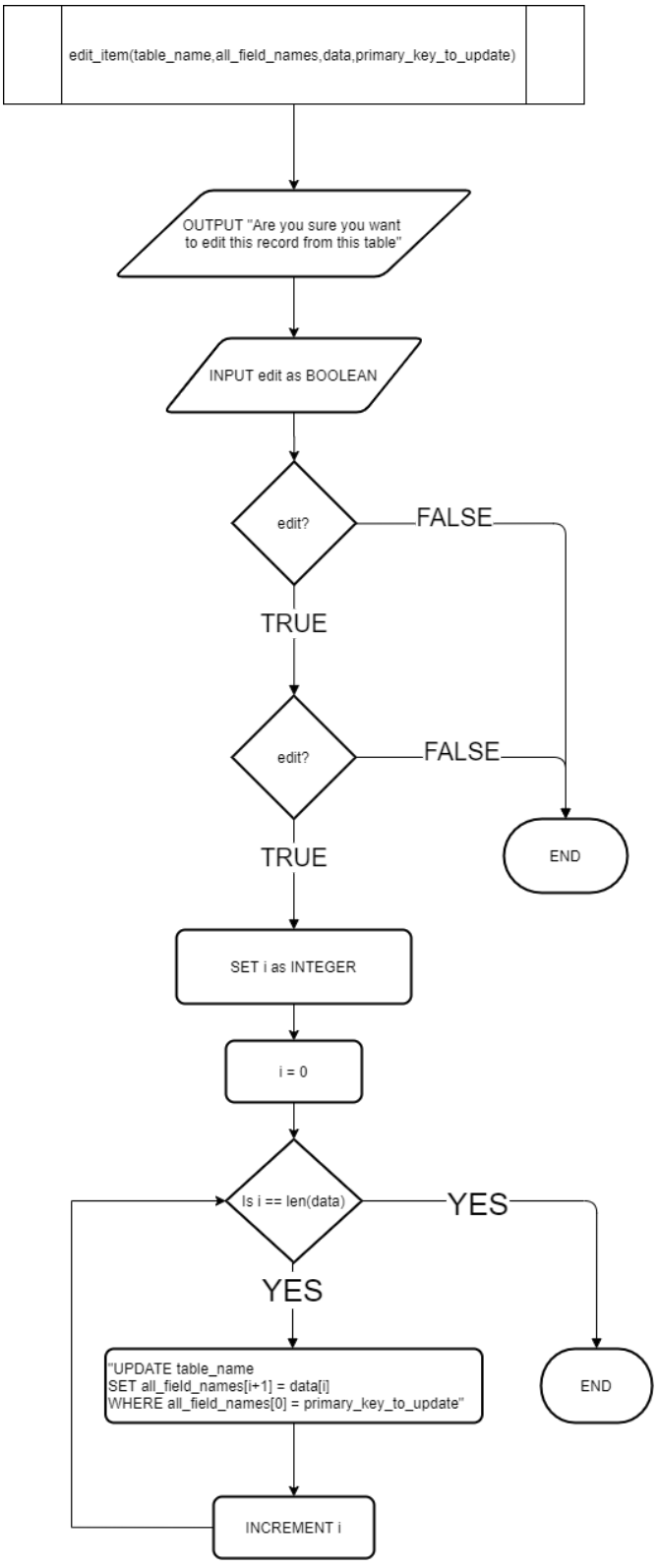


Delete flowchart

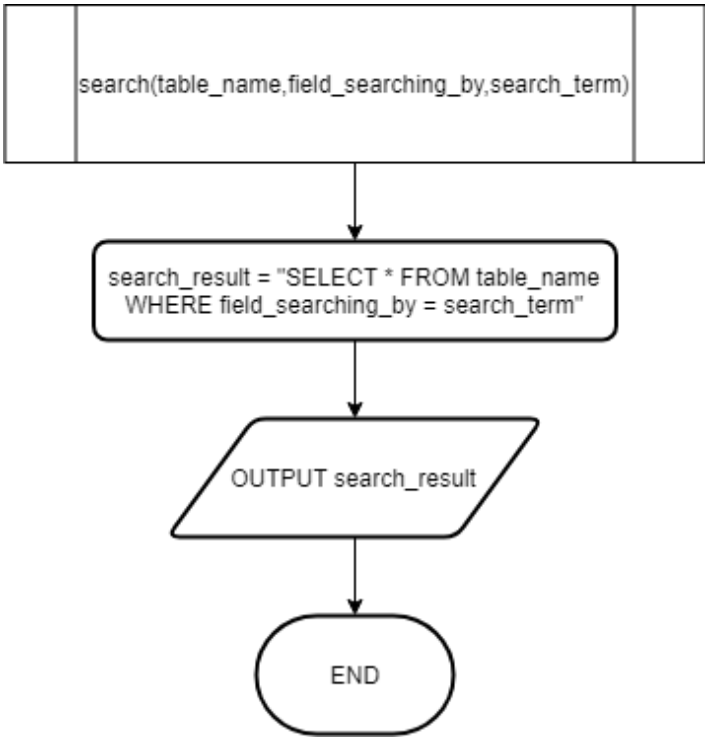


Edit flowchart

The below flowchart works under the assumptions that the primary key name is included within the ‘all_filed_names list’ and that the ‘data’ list does not contain the primary key of the record to be added.

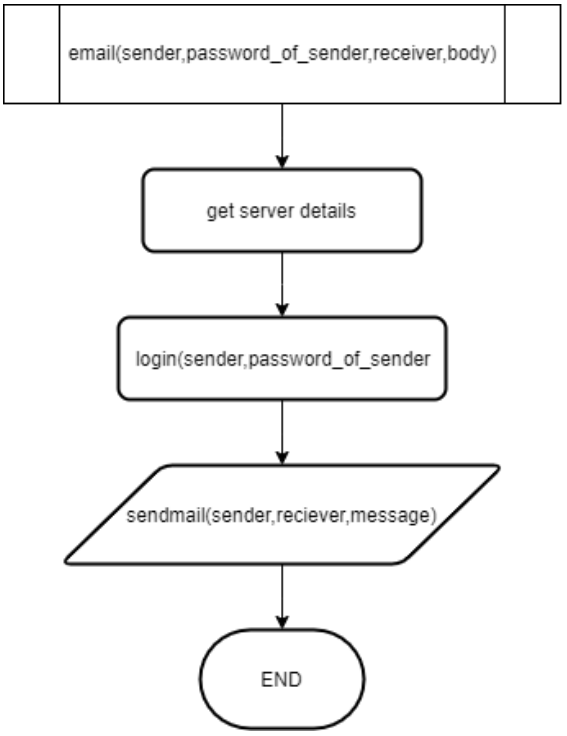


Search Flowchart



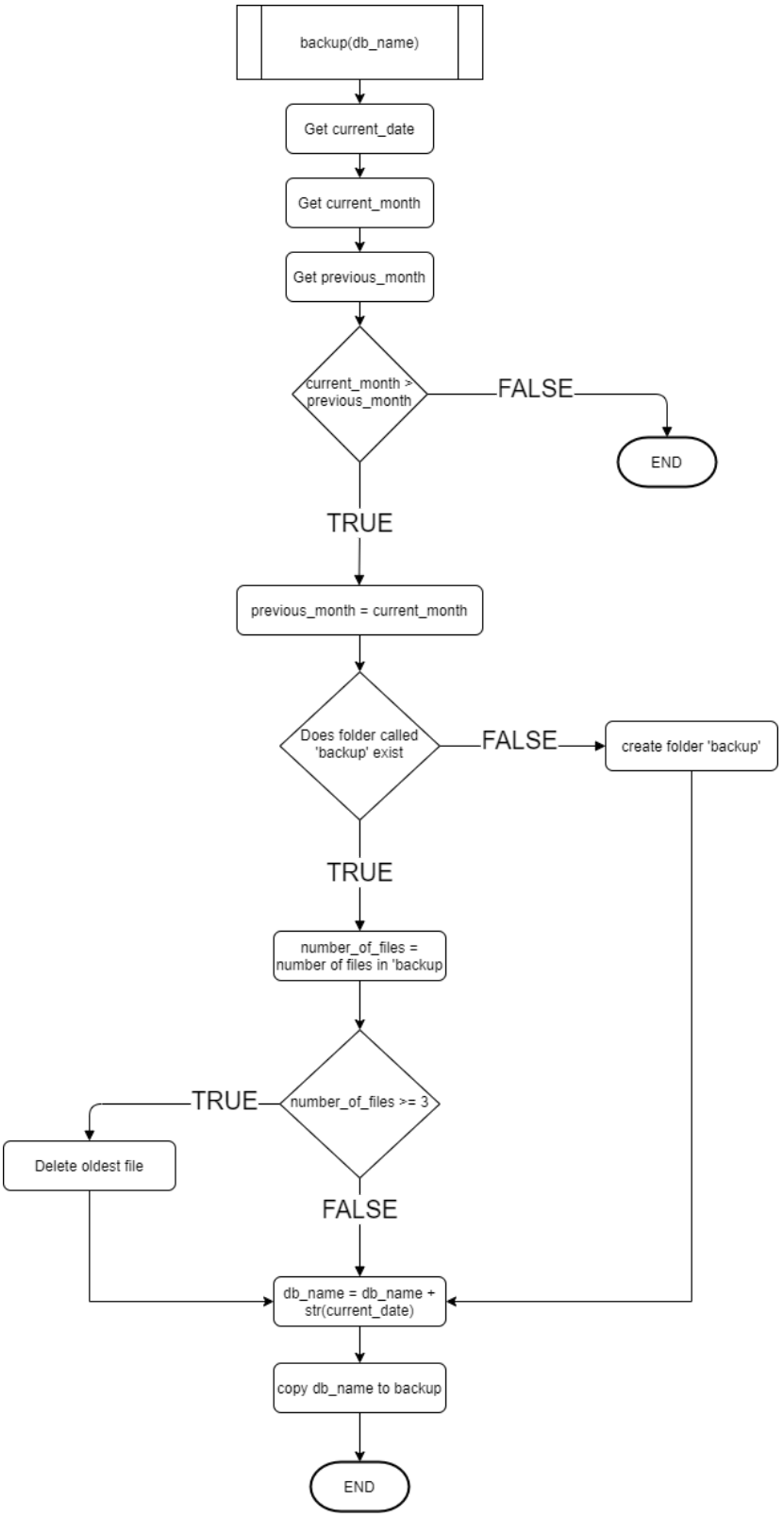
Email flowchart

The following libraries will need to be imported smtplib, email.mime.multipart and email.mime.text.

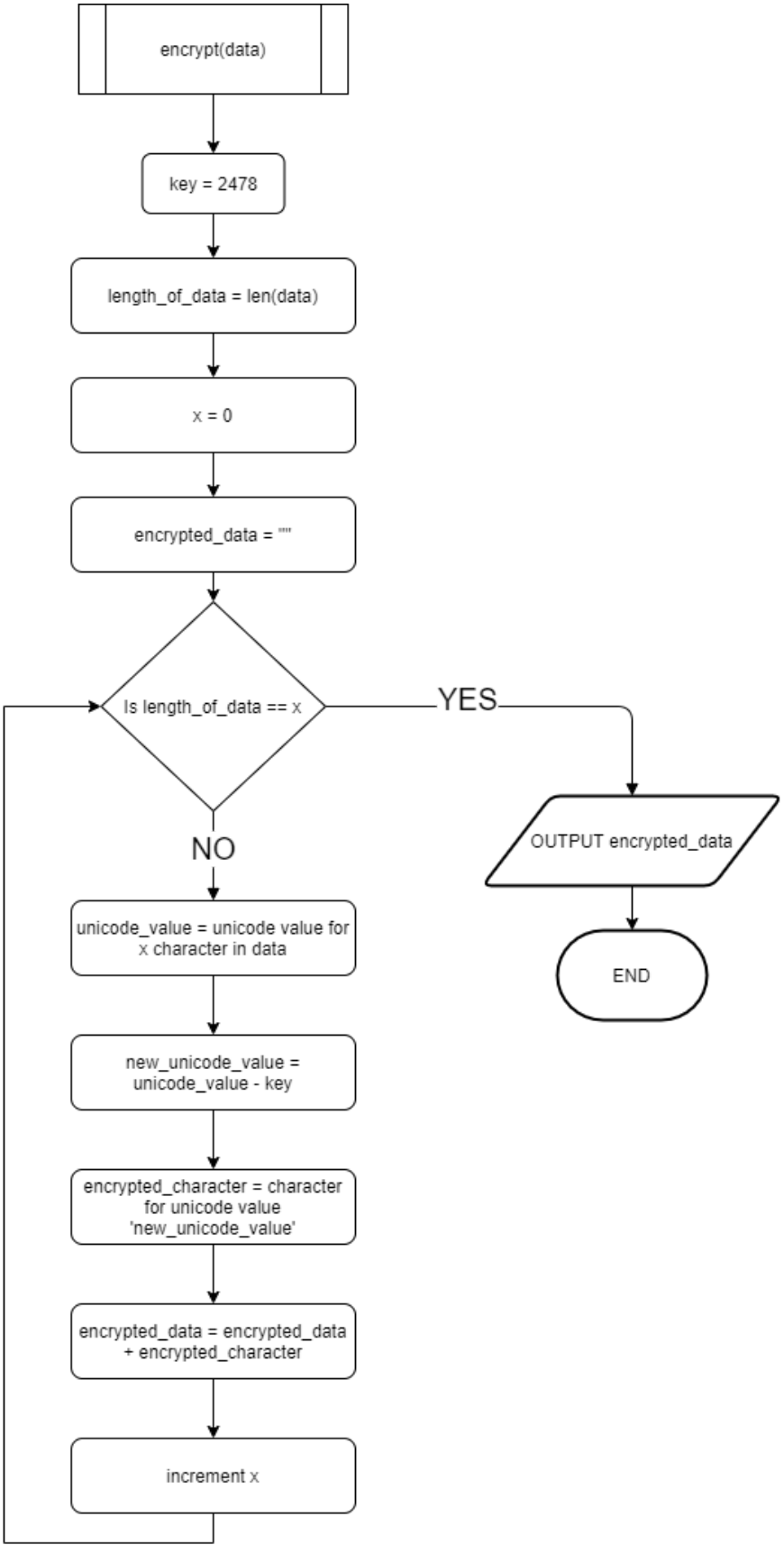


Backup

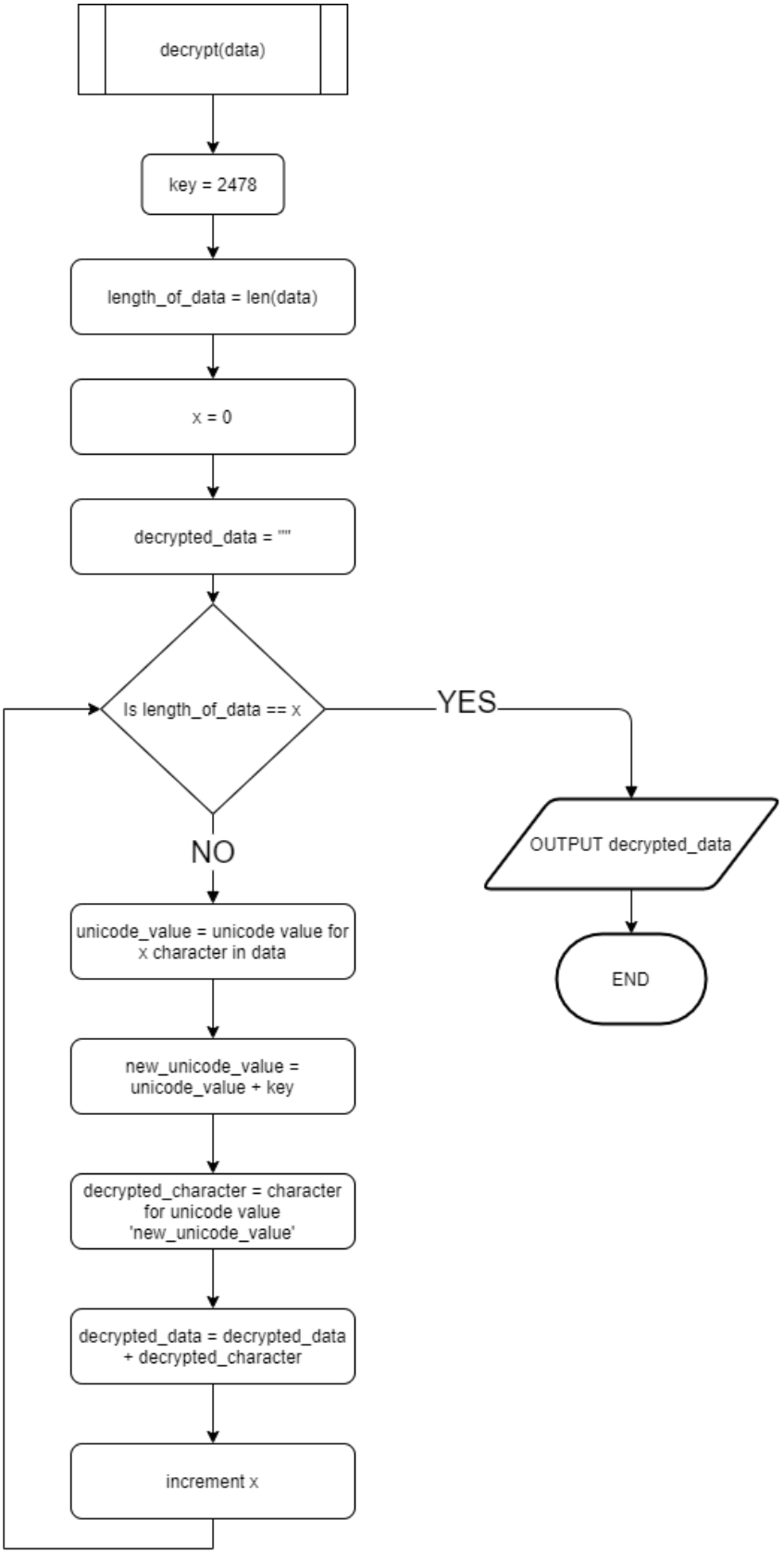
The libraries ‘datetime’, ‘os’ and ‘shutil’ will need to be imported for the following sub-routine to work. The parameter ‘db_name’ is a string and will be the name of the database that is being backed-up.



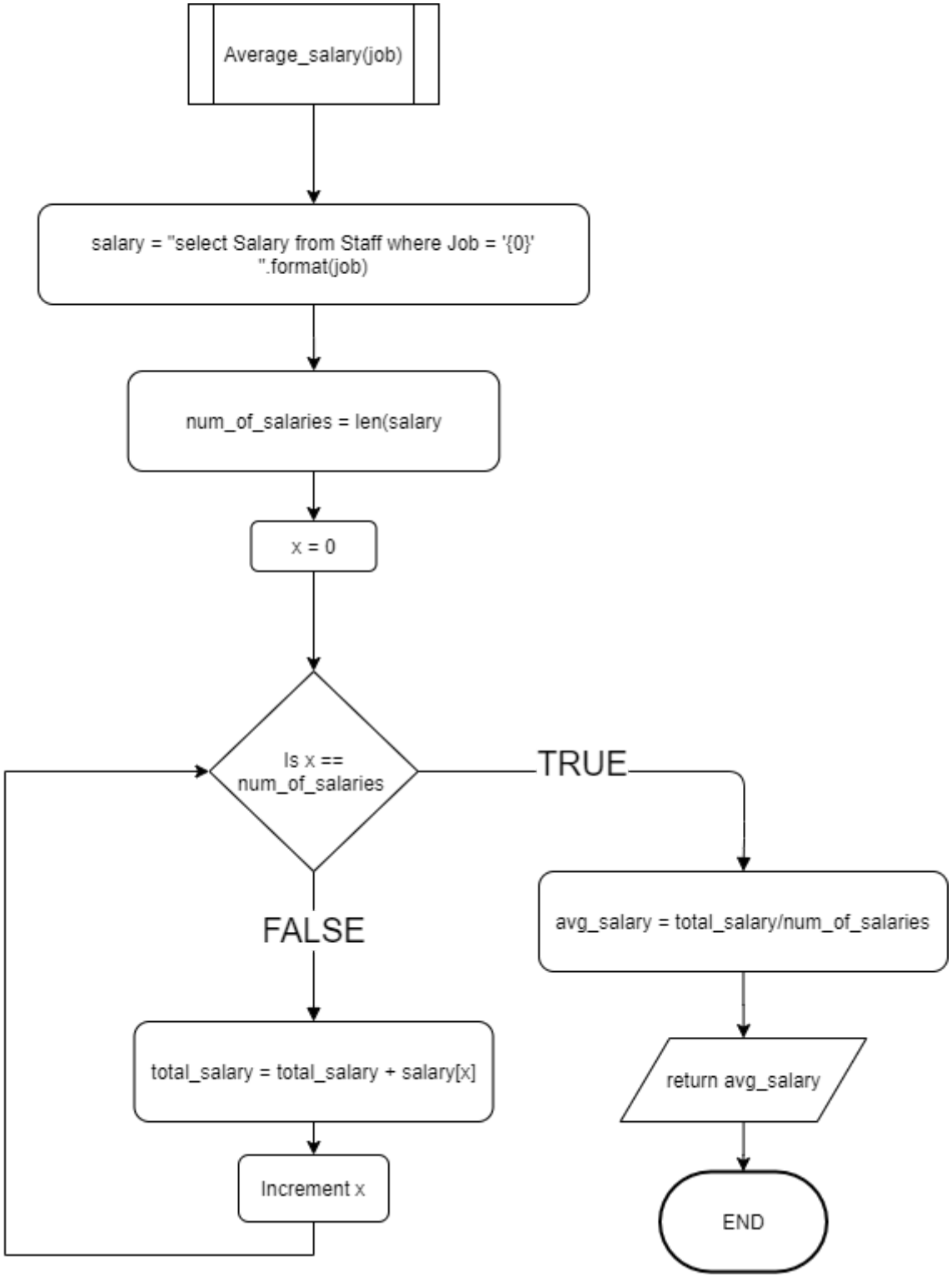
Encryption


encrypt(data)
key = 2478
length_of_data = len(data)
x = 0
encrypted_data = ""
Is length_of_data == x
YES
NO
OUTPUT encrypted_data
END
unicode_value = unicode value for x character in data
new_unicode_value = unicode_value - key
encrypted_character = character for unicode value 'new_unicode_value'
encrypted_data = encrypted_data + encrypted_character
increment x

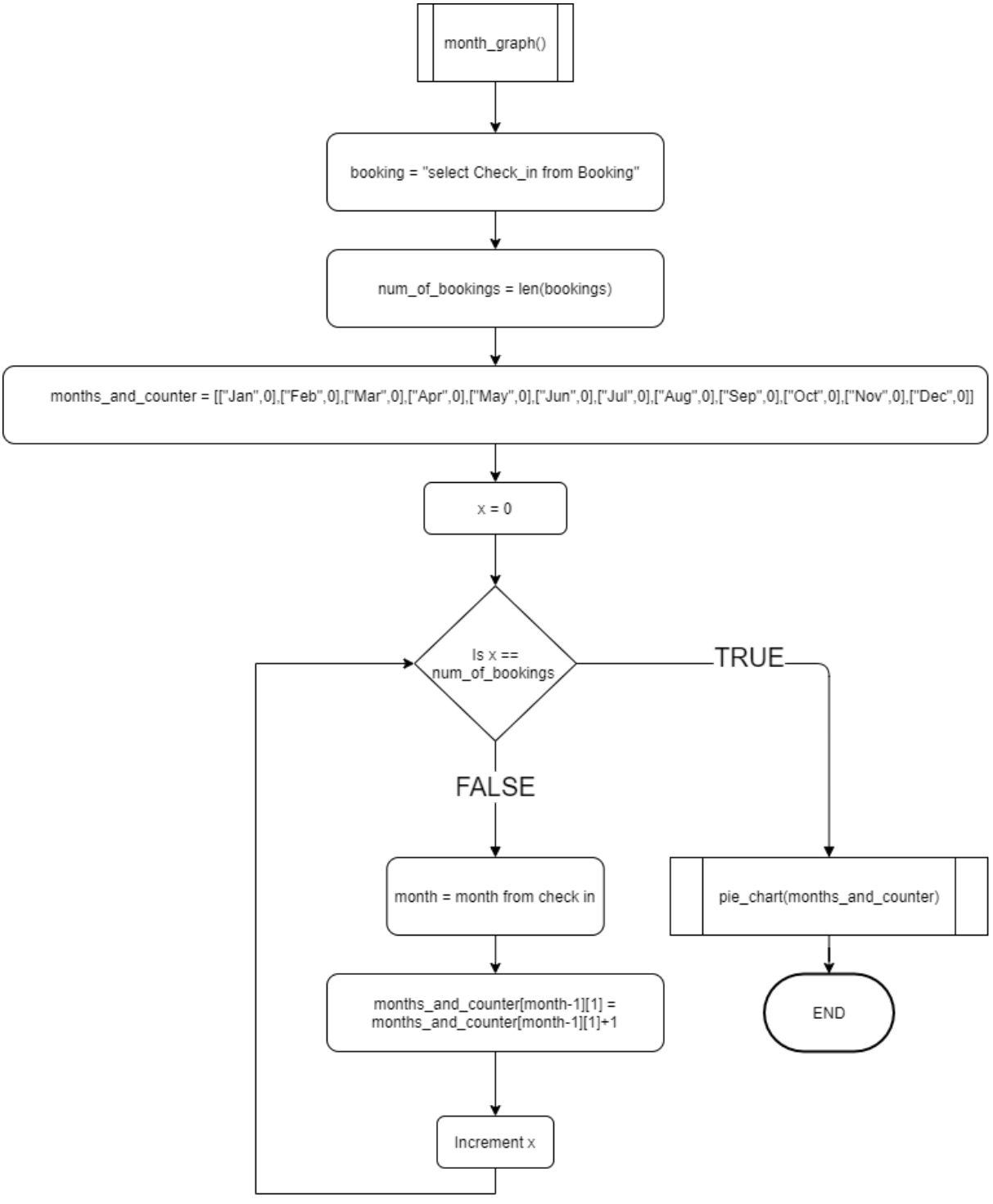
Decrypt


decrypt(data)
key = 2478
length_of_data = len(data)
x = 0
decrypted_data = ""
Is length_of_data == x
YES
NO
OUTPUT decrypted_data
END
unicode_value = unicode value for x character in data
new_unicode_value = unicode_value + key
decrypted_character = character for unicode value 'new_unicode_value'
decrypted_data = decrypted_data + decrypted_character
increment x

Average Salary



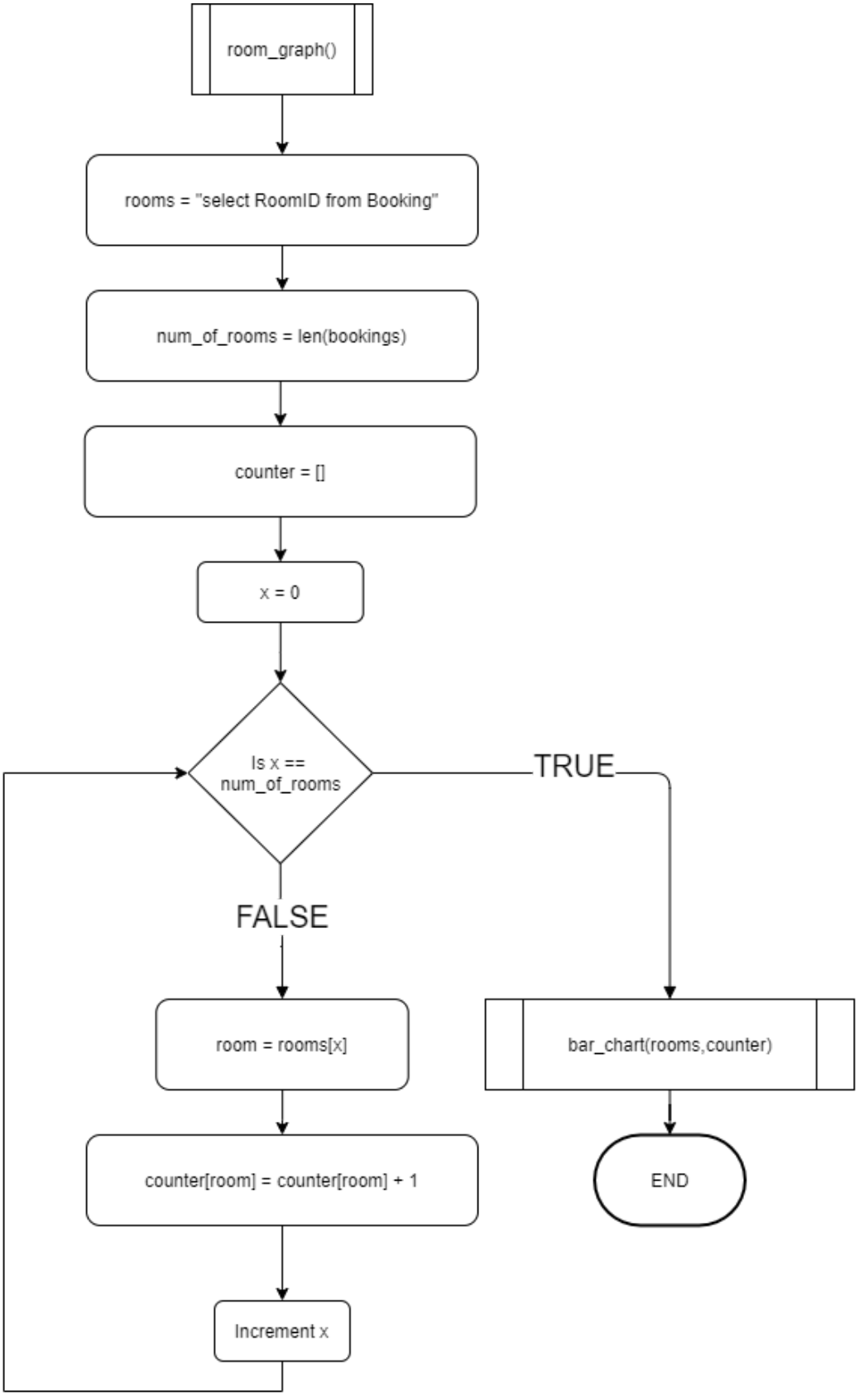
Getting data for Month graph



Pie Chart

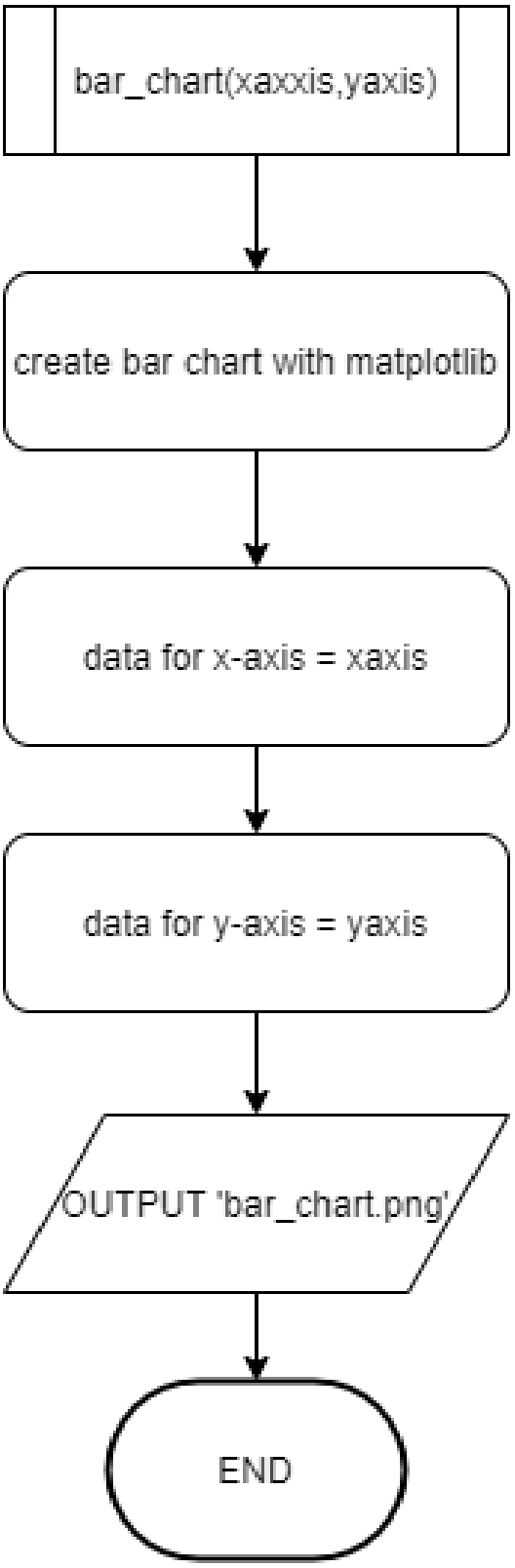
This function requires the library ‘matplotlib’ this library needs to be installed via the command prompt with the command ‘pip install matplotlib’.

Getting data for Room graph

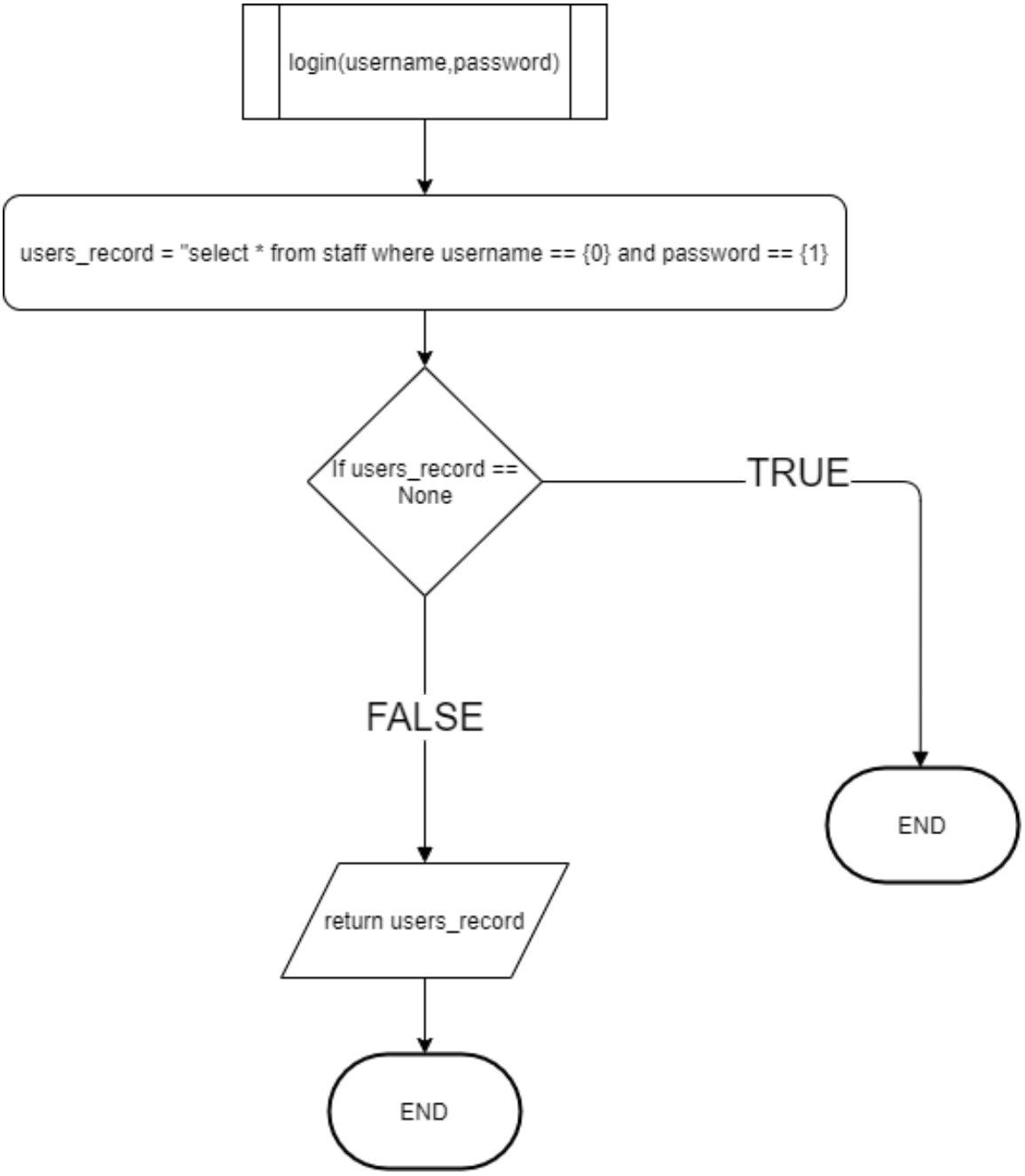


Bar Chart

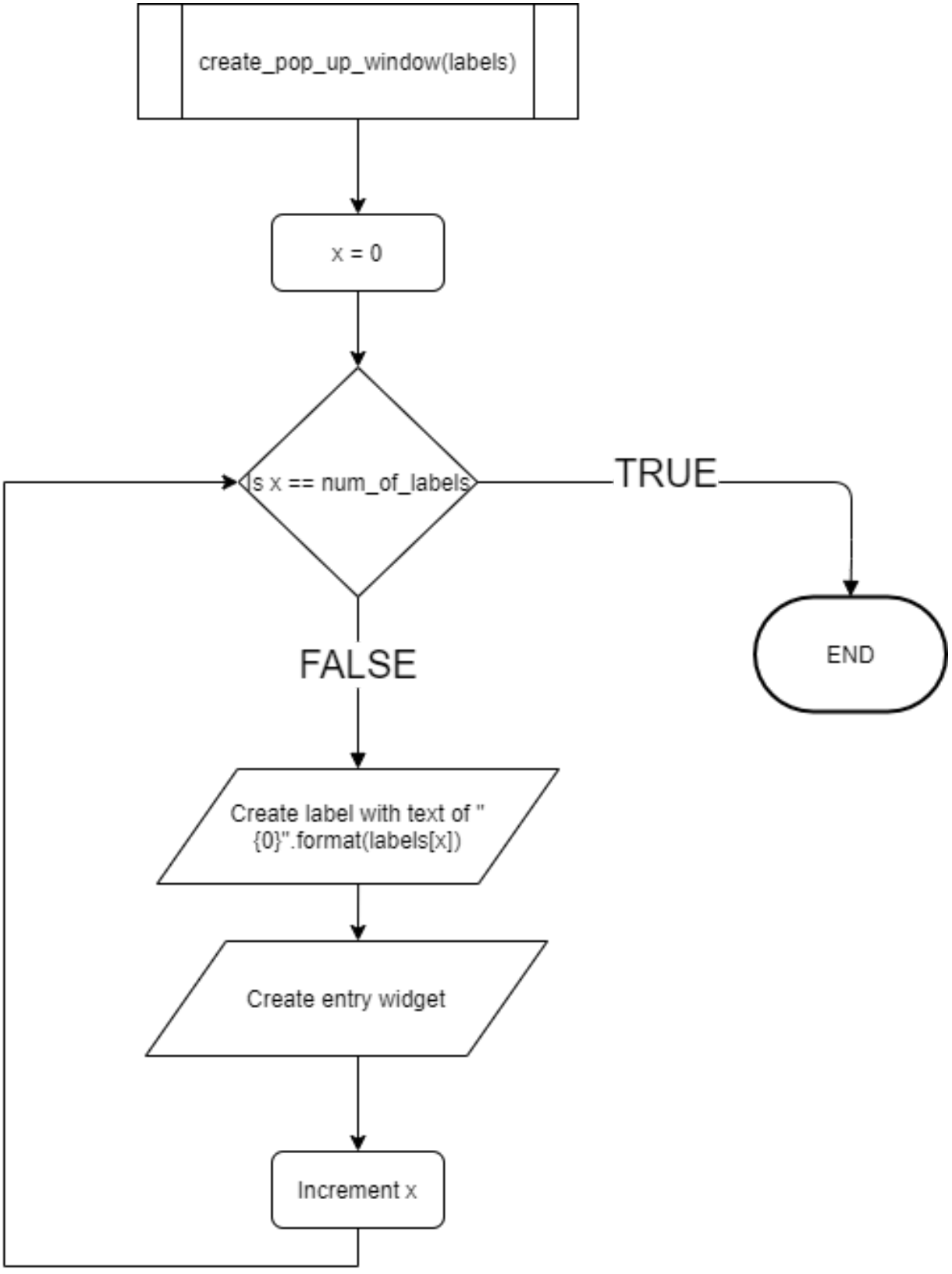
This function requires the library ‘matplotlib’ this library needs to be installed via the command prompt with the command ‘pip install matplotlib’.



Authentication

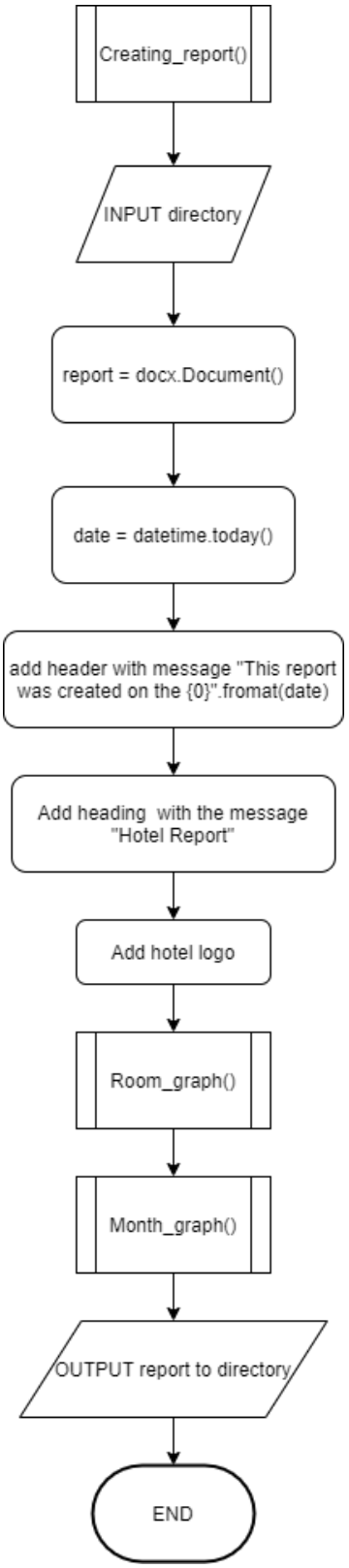


Creating ‘pop-up’ windows

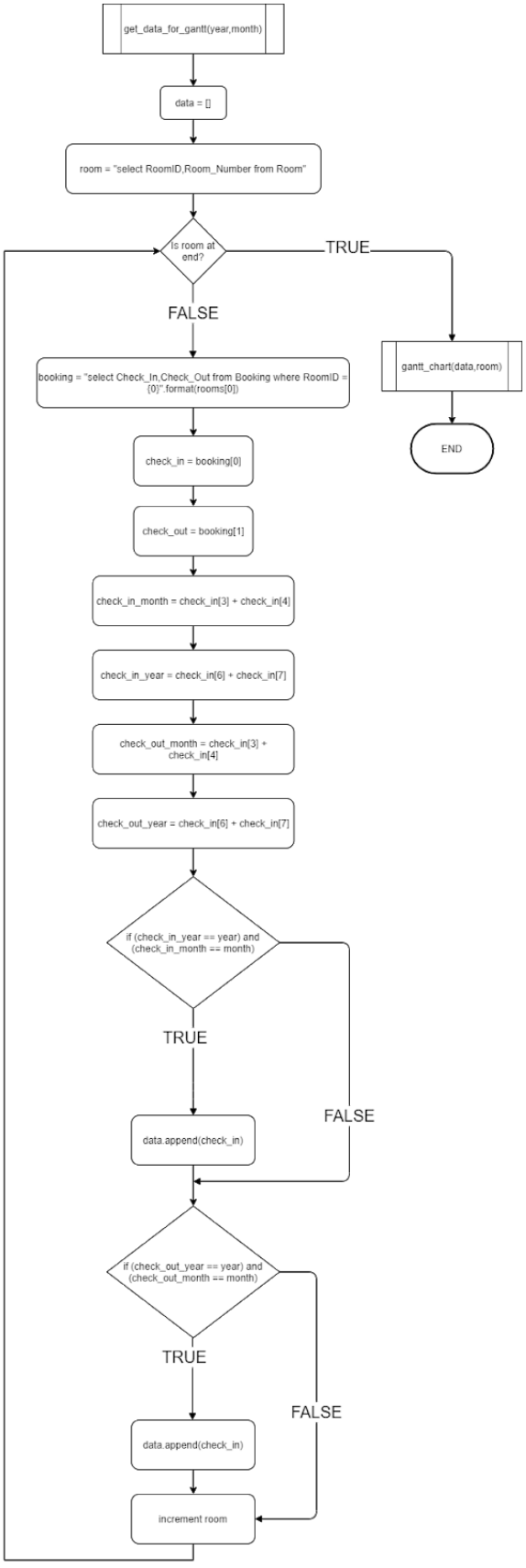


Creating Report

For this function to work ‘docx’ needs to be imported. This needs to be installed via the command prompt with the command ‘pip install python-docx’.

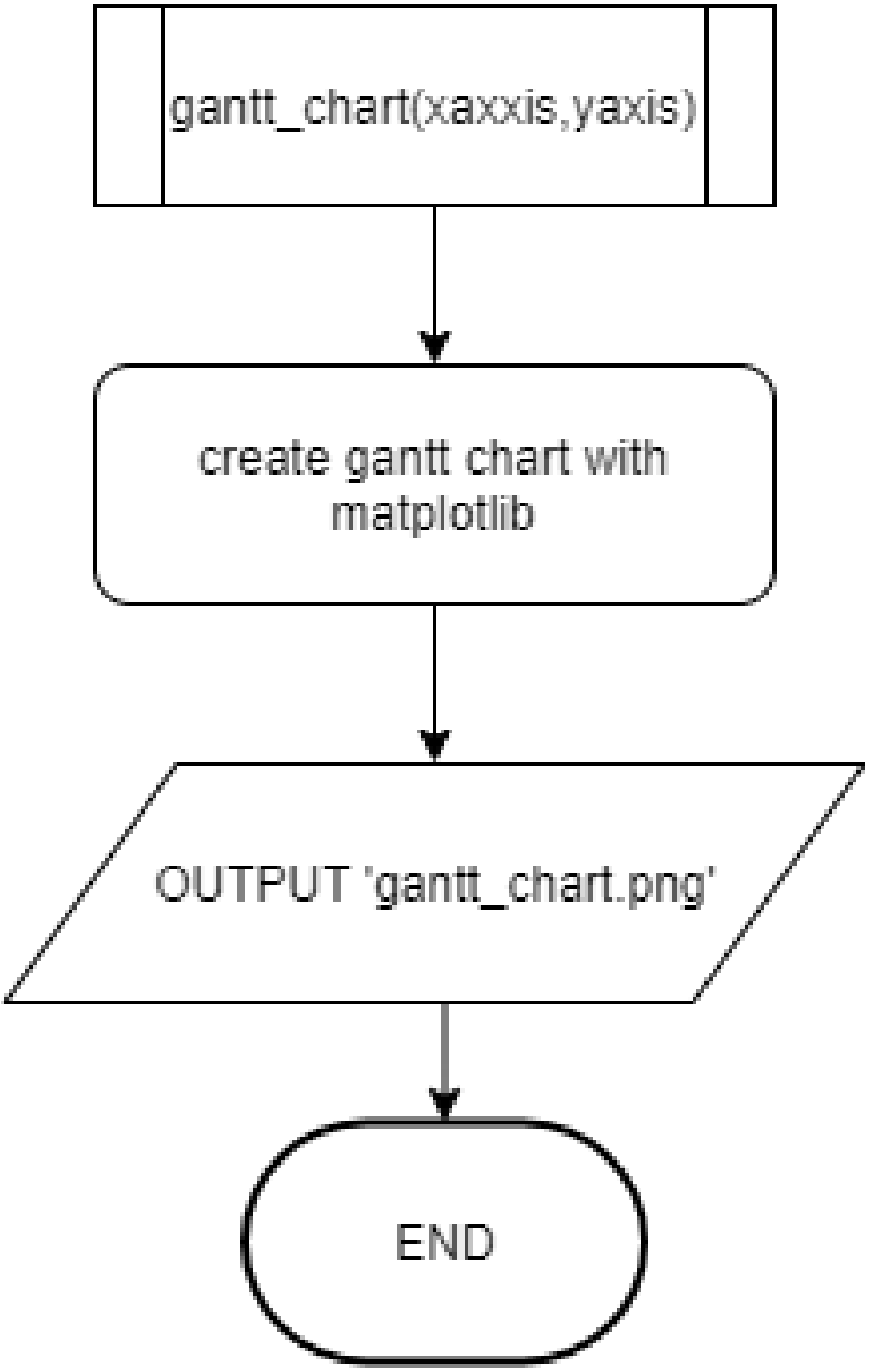


Getting data for tape chart



Tape chart

This function requires the library ‘matplotlib’ this library needs to be installed via the command prompt with the command ‘pip install matplotlib’.



Name of System

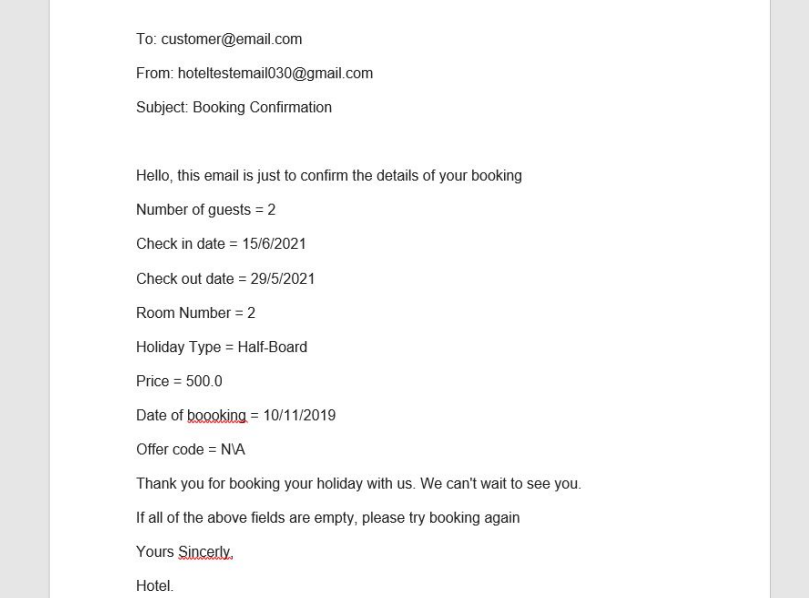
I have decided to name my system ‘Deversorium’ which is Latin for hotel. As this is an off-the-shelf solution the name of the system will be different than the name of the hotel or hotels which are using the system. Due to this fact this will be represented within my system by having the hotel’s logo in the top left corner of my system and the name of this hotel will be called ‘Gevora’.

Hard copy outputs

Email Design

Email booking details to the customer (prototype design)

Below is an example email that will be used for a confirmation email for the booking.


To: customer@email.com

From: hoteltestemail030@gmail.com

Subject: Booking Confirmation

Hello, this email is just to confirm the details of your booking

Number of guests = 2

Check in date = 15/6/2021

Check out date = 29/5/2021

Room Number = 2

Holiday Type = Half-Board

Price = 500.0

Date of boooking = 10/11/2019

Offer code = N\A

Thank you for booking your holiday with us. We can't wait to see you.

If all of the above fields are empty, please try booking again

Yours Sincerly,

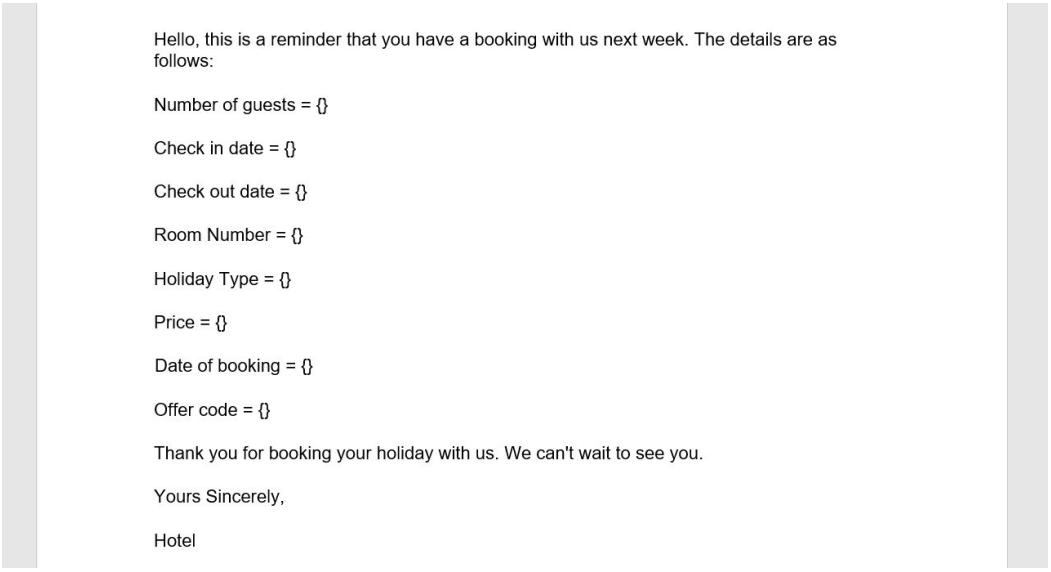
Hotel.

The data that is being displayed to the customer via the email is the entirety of the record in the booking table that relates to them excluding the roomID as this is replaced with the room number relating to that roomID.

The reason for this is to simply confirm these details with the customer and allows the customer to check the details of their booking without having to log into the system. This also allows the customer to have proof that they booked the holiday at the specified times.

Reminder email

The ‘{}’ brackets within the email design below simply represent the data from the corresponding field in the booking table.



Hello, this is a reminder that you have a booking with us next week. The details are as follows:

Number of guests = {}

Check in date = {}

Check out date = {}

Room Number = {}

Holiday Type = {}

Price = {}

Date of booking = {}

Offer code = {}

Thank you for booking your holiday with us. We can't wait to see you.

Yours Sincerely,

Hotel

Report

Below is a sample word document in which I would like my ‘report’ to look like. The font used will be ‘Ariel’ and the text size will be ‘11’. The headings will be blue.

This report was created on the {the current date}

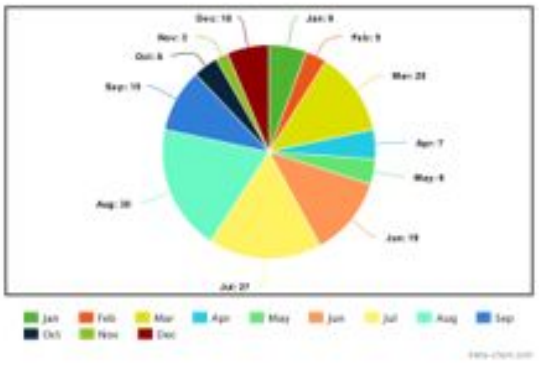
GEVORA HOTEL

Hotel Report

This reprot includes the 2 graphs that can be accessed within the main system.

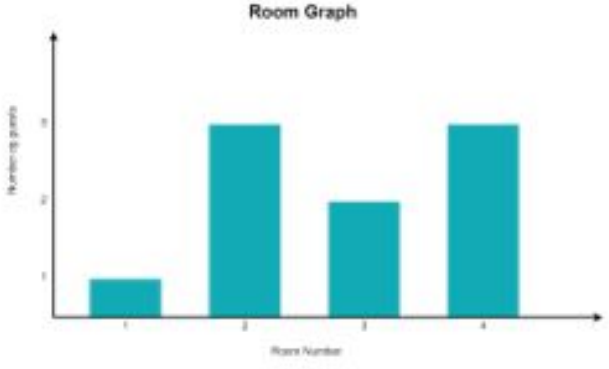
Month graph:

The graph below shows the percentage of bookings per month.



Room graph:

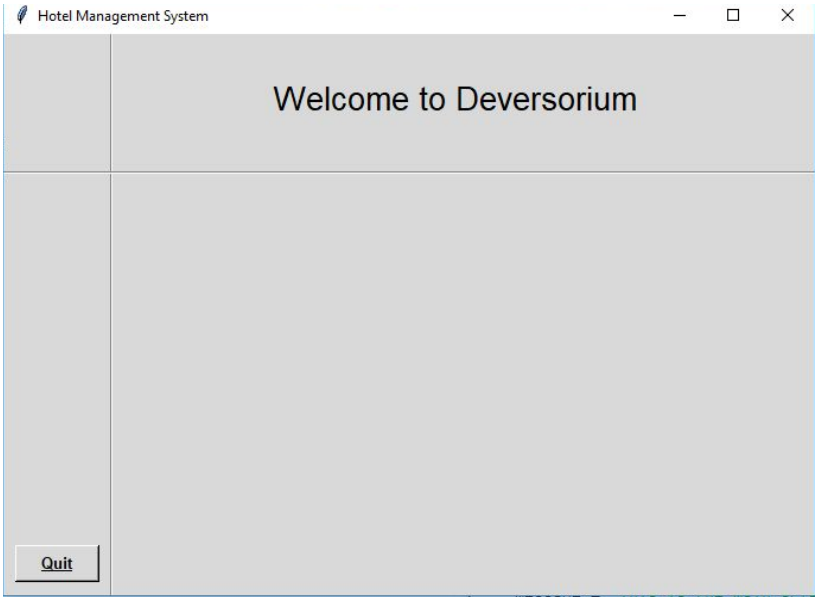
The graph below shows the number of bookings per room.



Screen Output (Prototype design)

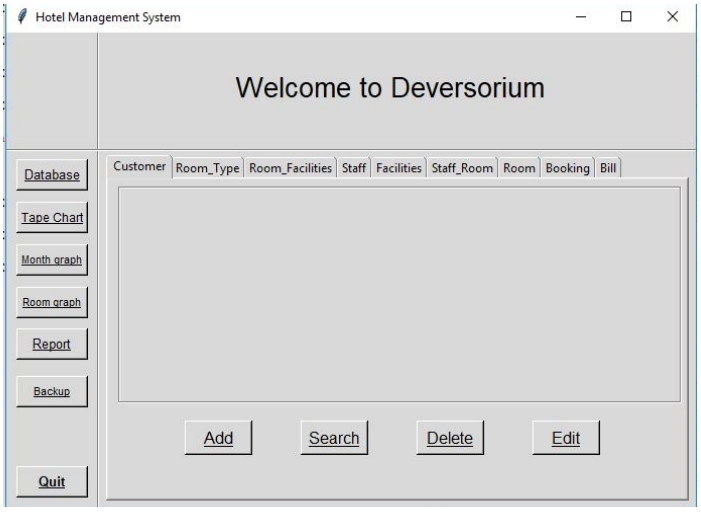
Main menu

As can be seen below the font used for the welcome message and the button is ‘Ariel’ and the background for the system is grey.. The text within all the buttons displayed on the left is both bold and underlined.



Management Screen

The buttons to the left except for the button ‘Quit’ button are the font ‘Ariel’ and underlined. The text within the tabs of the are also of font ‘Ariel’. The ‘Add’, ‘Edit’, ‘Search’ and ‘Delete’ buttons are also ‘Ariel’ and underlined. The rectangle within the notebook will be the table of the tab selected.

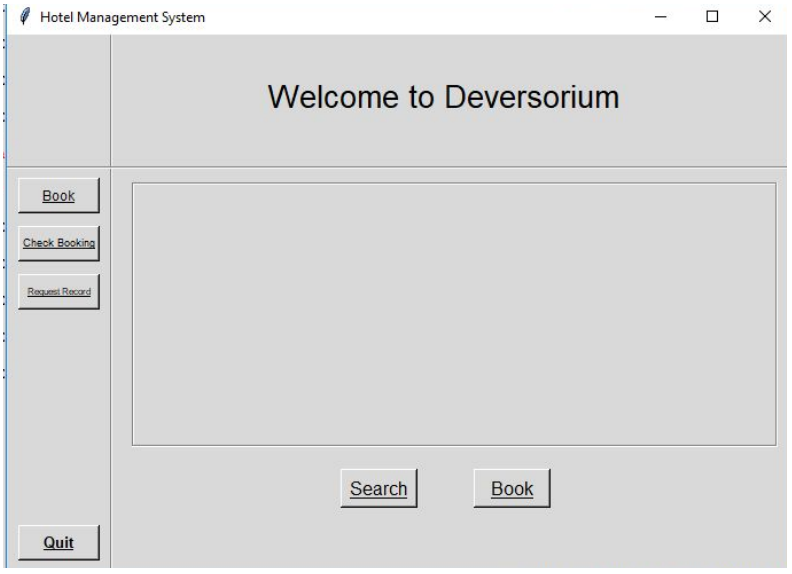


Receptionist Screen

The rectangle within the notebook will be the table of the tab selected.

Customer Screen

The rectangle within the notebook will be the table of the tab selected.



Accountants Screen

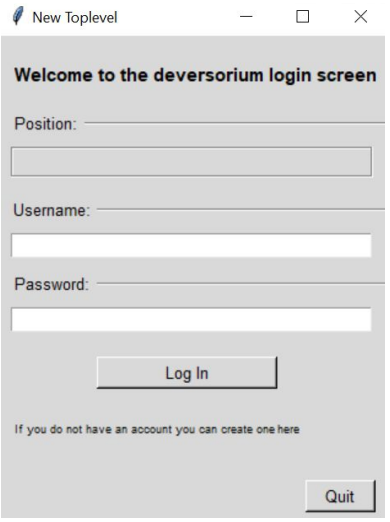
The rectangle within the notebook will be the table of the tab selected.

Cleaner Screen

The rectangle within the notebook will be the table of the tab selected.

Login Screen

All text within the login screen will be Arial and size 11. An option menu will be placed within the frame below ‘Position’.



Message boxes

Message boxes will be used when quitting the program, adding a record, deleting a record and editing a record. The creation of different message boxes are functions already within tkinter and therefore their layout is dictated by these functions. Although the text and title of the message box can be changed and I will be creating this text to display the record or quit message to make sure the user is sure of the action.

Average Salary

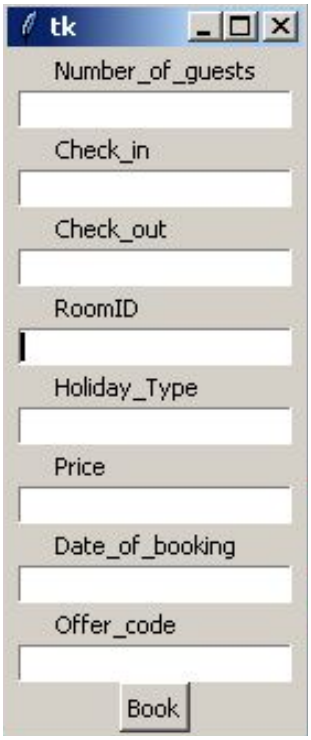
The text below will be size 14 font in ‘Ariel’. The ‘{}’ represent the calculated average salary of each position.

```
Below is the average salary for each position:
    -Manager = {}
    -Cleaner = {}
    -Accountant = {}
    -Receptionist = {}
```

Data entry:

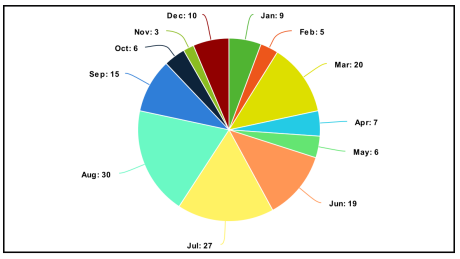
When the user needs to type data into the system a ‘pop-up’ window will be used. This ‘pop-up’ window will simply contain the entry widgets for the user to enter in the data and a button to execute the function. The reason for the ‘pop-up’ window is so the user can view the table which they are entering the data for. The font used for the labels in the ‘pop-up’ window will be ‘Dejavu Sans’ and the font used for the entry widgets is ‘Helvetica’.

Example of ‘pop-up’ window when adding to the booking table:



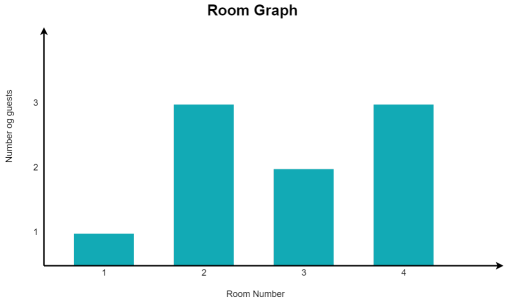
Pie chart

The font used will be the default font used by matplotlib which is DejaVu Sans. The colours used for each ‘slice’ will be automatically generated by matplotlib as well. The example pie chart below is from ‘https://www.meta-chart.com/pie#/display’ which I used random data to fill the chart.



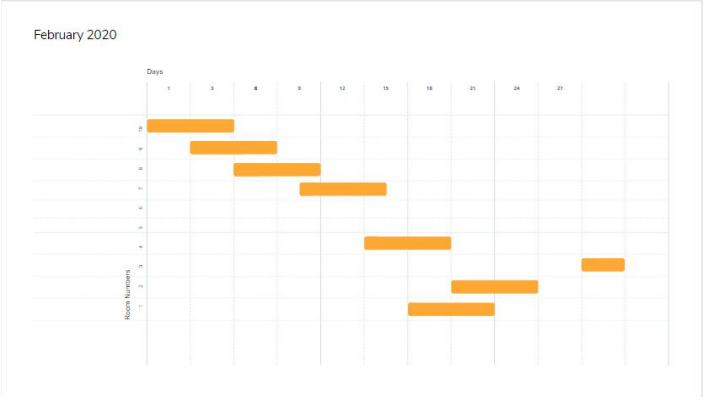
Bar chart

The font used will be the default font used by matplotlib which is DejaVu Sans. The colour of the bars will be blue as shown below. The example bar chart below is from ‘https://www.draw.io/#G1McMaBiVtz’’ which I used random data to fill the chart.



Tape Chart

The font used will be the default font used by matplotlib which is DejaVu Sans. The colour of the bars will be orange as shown below. The example tape chart below is from ‘https://app.moqups.com/8v5a0yzDoG/view/page/a0bf9cbc1?ui=0’ which I used random data to fill the chart.



Welcome message design

All welcome messages will be displayed in ‘Ariel’ size 8 font.

Manager Welcome Message

```
This is the home screen of the hotel Management system:
-To quit simply select the quit button at the bottom left.
-If you wish to return to this page at any time simply select the 'Home' button at the bottom left.
-To view the database in its entirety please hit the 'Database' button to the left. This will display the database with
 each tab being a table in the database.
    -In this screen you can add, edit, delete and search for each record
        -To delete a record just select the record and hit the edit button
        -To edit a record just select the record and hit edit
                -A pop-up window should appear and simply type the changes you would like to make
                -When editing the 'Staff' table do NOT use a comma in the SALARY field
        -To add a record hit the add button
                -A pop-up window should appear and simply type the record you wish to add
        -To search for a record select the search button
                -A pop-up window will appear and will prompt you for the 'ID' of the record you would like to search
        -There is an advanced search option allows you to search by any field
                -A pop-up window will appear and will prompt you to choose a field and to enter in a search term.
                -A new tab will open and with the search results.
-When the tape chart button is selected this will show a chart of when the rooms are occupied.
-The 'Month graph' button will display a graph of number of bookings per months.
    -This is based of how many check-ins there are per month.
-The 'Room graph' button will display the number of guests per room.
-The 'Report' button will produce a report which contain:
    -The 2 graphs described above.
    -The date which the report was created.
-More information on backups can be found when the 'Backup' button is pressed.
-To return to this screen please press 'Home'.
```

Customer Welcome Message

```
This is the home screen of the hotel Management system:
-To quit simply select the quit button at the bottom left
-If you wish to return to this page at any time simply select the 'Home' button at the bottom left.
-To book a holiday just hit the 'Book' button
    -Select a room and hit the 'book' button below the table
    -In this screen you can search for a room
        -To search for a room select the search button
                -A pop-up window will appear and will prompt you for the 'ID' of the record you would like to search
        -There is an advanced search option allows you to search by any field
                -A pop-up window will appear and will prompt you to choose a field and to enter in a search term
-To retrieve your record hit the 'Request Record' button
        -To edit a record just select the record and hit edit
                -A pop-up window should appear and simply type the changes you would like to make
To view bookings hit the 'Check Booking' button
        -To edit a record just select the record and hit edit
                -A pop-up window should appear and simply type the changes you would like to make
                -When editing a booking DO NOT CHANGE THE CUSTOMERID
                        -This links the booking with you and therefore if you change it then the booking is no
                         longer made with you
                -'RoomID' is used instead of room number and therefore if you would like to change room please
                 contact a member of staff to receive this information
                -If you would like to cancel the booking contact a member of staff
```

Receptionist Welcome Message

```
This is the home screen of the hotel Management system:
-To quit simply select the quit button at the bottom left
-If you wish to return to this page at any time simply select the 'Home' button at the bottom left.
-To view the database please hit the 'Check Tables' button to the left. This will display a Customer table,
 Room table and the Booking table (only access to the search and advanced search buttons)
    -In this screen you can add, edit, delete and search for each record
        -To delete a record just select the record and hit the edit button
        -To edit a record just select the record and hit edit
                -A pop-up window should appear and simply type the changes you would like to make
        -To add a record hit the add button
                -A pop-up window should appear and simply type the record you wish to add
        -To search for a record select the search button
                -A pop-up window will appear and will prompt you for the 'ID' of the record you would like to search
        -There is an advanced search option allows you to search by any field
                -A pop-up window will appear and will prompt you to choose a field and to enter in a search term
-The 'Tape Chart' button displays a chart showing the bookings within a particular month
        -Type in a year, select a month and hit 'Go' to display bookings for the time selected
-To retrieve your record hit the 'Request Record' button
        -To edit a record just select the record and hit edit
                -A pop-up window should appear and simply type the changes you would like to make
                -When editing the record do NOT use a comma in the SALARY field
```

Cleaner Welcome Message

```
This is the home screen of the hotel Management system:
-To quit simply select the quit button at the bottom left
-If you wish to return to this page at any time simply select the 'Home' button at the bottom left.
-To view the Room table hit the 'Room Table' button
    -In this screen you can search for a record
        -To search for a record select the search button
                -A pop-up window will appear and will prompt you for the 'ID' of the record you would like to search
        -There is an advanced search option allows you to search by any field
                -A pop-up window will appear and will prompt you to choose a field and to enter in a search term
-To retrieve your record hit the 'Request Record' button
        -To edit a record just select the record and hit edit
                -A pop-up window should appear and simply type the changes you would like to make
                -When editing the record do NOT use a comma in the SALARY field
```

Accountant Welcome Message

```
This is the home screen of the hotel Management system:
-To quit simply select the quit button at the bottom left
-If you wish to return to this page at any time simply select the 'Home' button at the bottom left.
-To view the Bill table hit the 'Bill Table' button
    -In this screen you can search for a record
        -To search for a record select the search button
                -A pop-up window will appear and will prompt you for the 'ID' of the record you would like to search
        -There is an advanced search option allows you to search by any field
                -A pop-up window will appear and will prompt you to choose a field and to enter in a search term
-The 'Salary' button displays the average salary of each position in the hotel
-To retrieve your record hit the 'Request Record' button
        -To edit a record just select the record and hit edit
                -A pop-up window should appear and simply type the changes you would like to make
                -When editing the record do NOT use a comma in the SALARY field
```