Addis Ababa Science and Technology University

College of Electrical and Mechanical Engineering

Department of Software Engineering

Individual Assignment

Students name id

1, Tilksew Misganaw CEP0107/14

2, Bereket kibret CEP0309/14

3, Yonas Birhanun CEP0121/14

Due Date: Sept 17, 2023

Contents

Introduction

Our project for the introduction to programming 2 was a hotel management system. So we tried to create a prototype For the best hotel management system.

Problem statement

On this digital world using old and slow hotel management system is inefficient.There are a lot problems we can encounter when we such methods for example
When a hotel does not utilize a digital hotel management system, it can face several challenges and problems. Here are some common issues:
1. Inefficient Reservation Management: Without a centralized digital system, managing reservations becomes complex. Double bookings or overbookings can occur, leading to dissatisfied guests and potential revenue loss.

2. Poor Guest Experience: A lack of digital tools can result in slower check-ins, longer wait times, and a less personalized experience for guests. It may be challenging to retrieve guest preferences or track their history, making it difficult to provide tailored services.

3. Limited Accessibility: Traditional paper-based systems may restrict access to crucial information. Staff members may need to physically search through files or documents, leading to delays and reduced productivity.

This are the three mani problem we tride to fix on this project .

By addressing this issues we tried to build a digital system that will solve those problems we mention above . our hotel-management system empower hotels by maximize customer satisfaction and collecting real world data and digitizing it .

Pseudo code

On this part we tried to show what our project do in the time of initiation .
Class "Customer":
Public class Attributes:
From_date (an integer)
To_date (an integer)
Name (a string)
Fname (a string)
Age (an integer)
Sex (a character)
Idno (an integer)
Address (a string)
Phone (an integer)
Class "Room":
Public class Attributes:
type (a string)
roomno (an integer)
cust (an instance of the Customer class)
day (an integer)
Function called "price" that returns a float:
If the type of the room is "Single":
Return the product of day and 500
Otherwise, if the type of the room is "Double":
Return the product of day and 1000
Otherwise, if the type of the room is "Family":
Return the product of day and 3000
Set the attribute "status" to "free" (a string)
Class "HotelMng":
private class attributes:

room (an array of rooms with a size of 18)
roomcount (an integer)
Public class Attributes:
function "check_in":
Declare choice as a character
Declare name as a string
If there is a free room available:
Display a menu to choose the type of room:
Option A: Single bedroom
Option B: Double bedroom
Option C: Family standard room
Option D: Back to the main menu
Prompt the user to choose an option and store it in the variable "choice"
Convert the choice to uppercase
Switch statement to handle the chosen option:
Case A:
If a single bed room is available:
Call Book with a parameter (single)
Else:
Display a message that all single bed rooms are taken
Break
Case B:
If a double bed room is available:
Call Book with a parameter (Double)
Else:
Display a message that all double bed rooms are taken
Break
Case C:
If a family standard room is available:
Call Book with a parameter (Family)
Else:
Display a message that all family standard rooms are taken
Break
Case D:
Break

Default:
Display a message to enter a valid choice
Go back to the first step of choosing an option
Break
Else:
Display a message that all rooms are reserved
Function "check_out":
Declare roomn as an integer
Declare answer as a character
If there is at list 1 person in the hotel:
Prompt the user to enter a room number (0 to exit) and store it in "roomn"
If the room number is valid and the room is taken:
Show the information of the room
Prompt the user if the information is correct (Y/N) and store the answer in "answer"
Convert the answer to uppercase
If the answer is 'Y':
Change the status of the room to "needClean"
Call Save() to save the changes in file
Display a message that the room has been checked out successfully
Else if the answer is 'N':
Display a message that the available information is only by room number
Prompt the user to try again by changing the room number
Go back to the step of entering a room number
Else:
Display a message to enter only 'Y' or 'N'
Go back to the step of prompting the user if the information is correct

Else if the room number is invalid:
Display a message to enter a valid room number
Go back to the step of entering a room number

Else if the room is free or needs cleaning:

Display a message that the room is empty and prompt the user to enter a valid room number
Go back to the step of entering a room number
Else:
Display a message that there is no person in the room
Function book(roomtype)
    Declare answer as a character
    Get all information about the customer
    Store the information to room[roomcount]
Call show(room[roomcount]) to display all necessary informations
Ask the user if that information is correct
    Input answer
Covert answer to uppercase
If answer is 'Y' then
        Call Save()
        Output "Room checked in successfully”
    Else If answer is 'N' then
        Delete room[roomcount]
        Clear screen
        Output "Please enter the information again"
        Call check_in()
    Else
        Output "Please enter only (Y/N)"
        Go to third
    End If
End function
Function getstatus()
Declare roomnum as an integer
Open "bookedstatus.txt" for reading as file
While reading roomnum from file
    Store that room information in room[roomnum-1]
Close file
End function
Function save()
Open "bookedstatus.txt" for writing as file
For I = 0 up to 18

```
If room[I].status is different from “free”
Write it in the file
        Close file
End function
Function available()
Display all available rooms
End function
function showPersonalInfo():
if there are any taken rooms:
            repeat:
Get first name and last name from the user
                    if name is equal to "X":
                        exit the repeat loop
                    for each room in the room list:
                        if the room is taken and the customer's
first name and last name match:
                                display the room information
Else
                        display "There is no person found with
this name “
                        display "Please try another name"
                        go back to the repeat loop
Else:
            Display "Sorry, there are no persons in our rooms"
Function showAllReservedRoom():
clear the screen
for each room in the room list:
            if the room is taken:
                display the room information
End function
function managerSide():
loop forever:
            display "Rooms that have recently been taken and
need cleaning"
            for each room in the room list:
                if the room status is equal to “needClean”:
                    display the room number
```

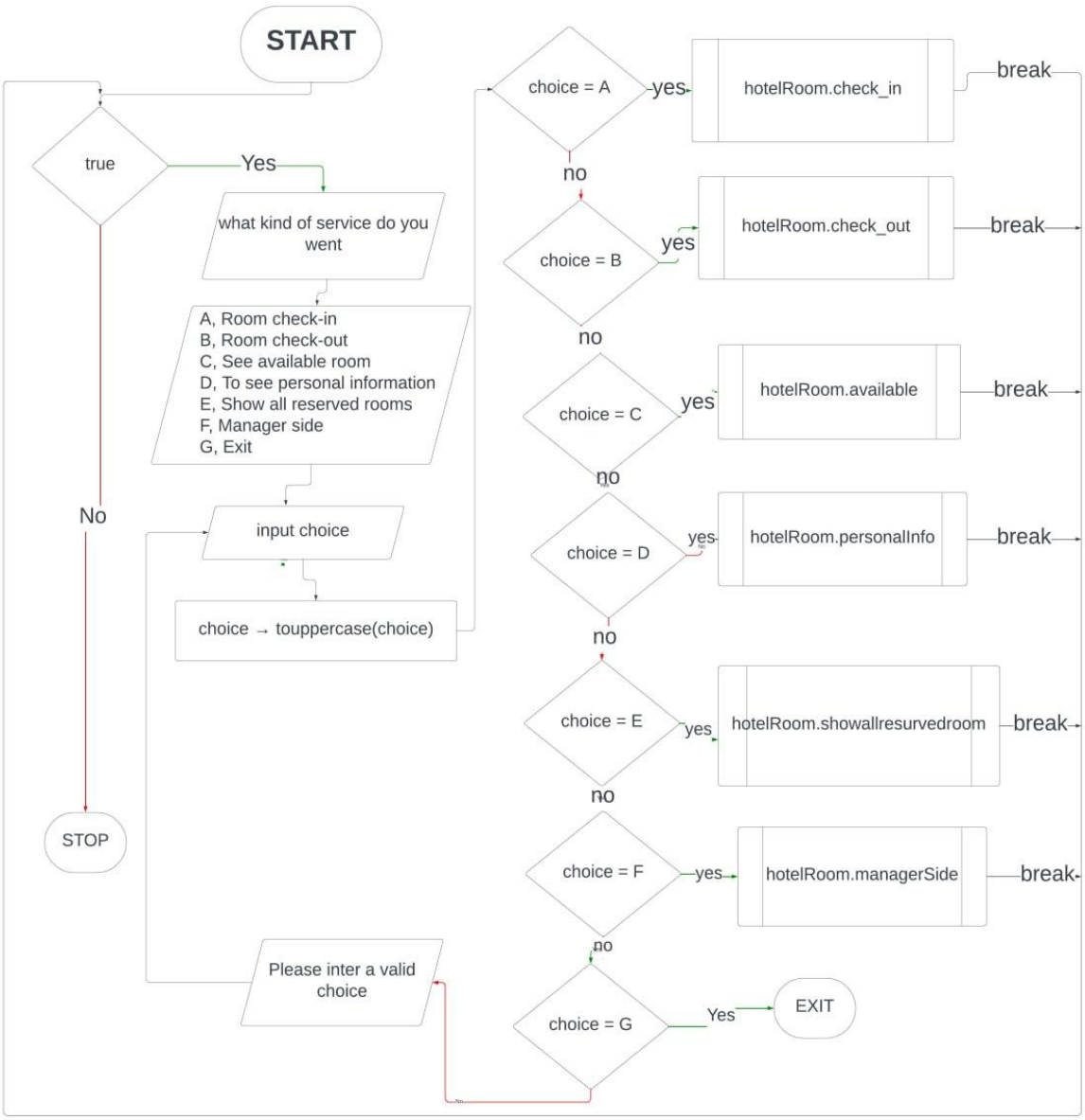
display "Add a room number to the cleaning list (0 to exit): "
input roomno
if the room by room number roomno is marked as needing cleaning:
mark the room as free
save the changes
display "the Room number successfully added to the list of available rooms"
else:
display "Please enter a valid room number"
Function main():
Call the choose() function
End of main function
Function choose():
Create an instance of hotelmng called hotelroom
Call the getstatus() method of hotelroom
While true:
Clear the screen
"WELCOME TO OUR HOTEL"
Display "What kind of service do you want?"
Display "A. Room check-in"
Display "B. Room check-out"
Display "C. See available rooms"
Display "D. To see personal information"
Display "E. Show all reserved rooms"
Display "F. Manager side"
Display "G. Exit"
Input the choice as a character
Switch choice:
Case 'A'
Clear the screen
Call the check_in() method of hotelroom
Pause
Case 'B'
Clear the screen
Call the check_out() method of hotelroom
Pause

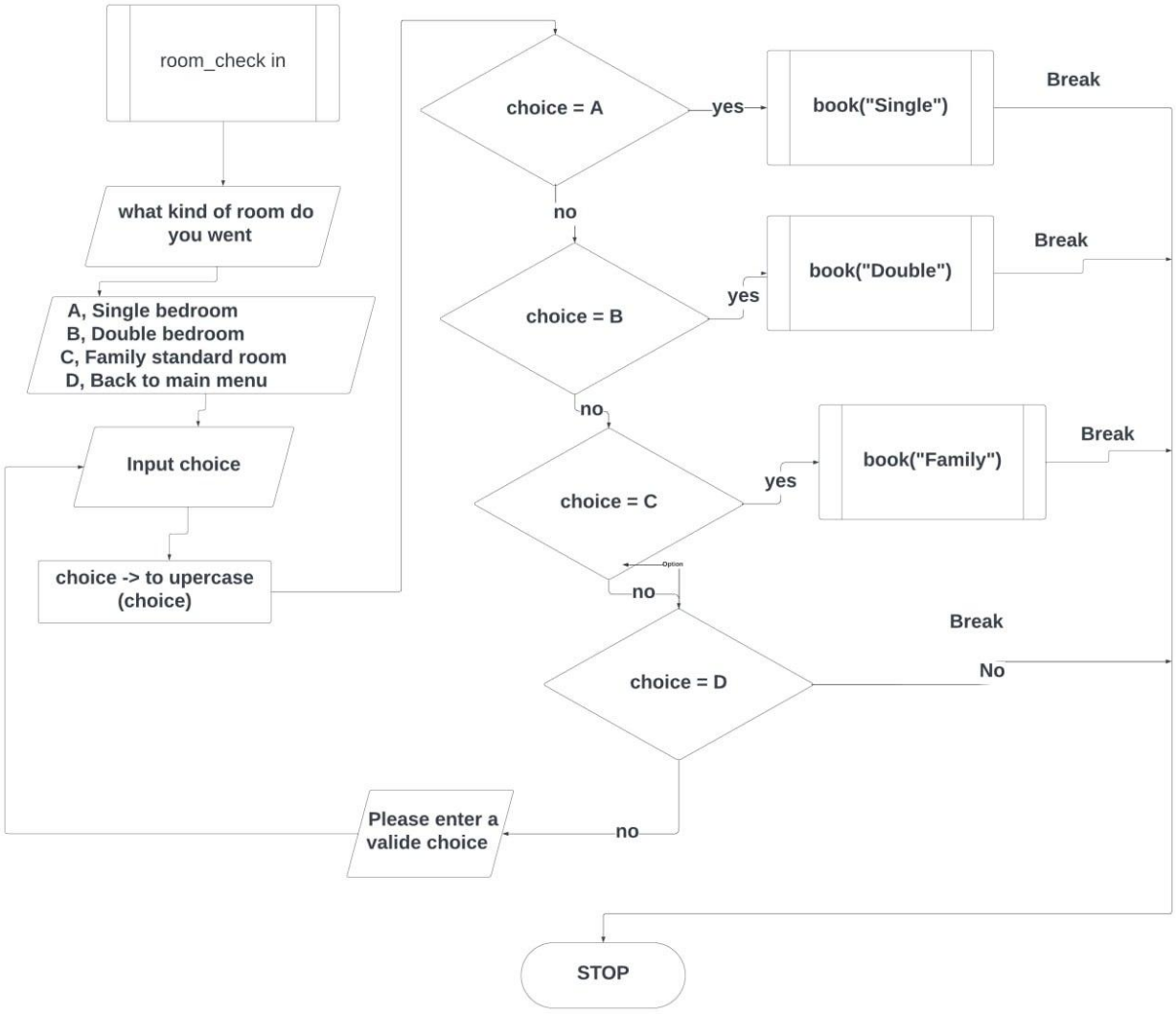
Case 'C'
Clear the screen
Call the available() method of hotelroom
Pause
Case 'D'
Clear the screen
Call the showpersonalinfo() method of hotelroom
Pause
Case 'E'
Clear the screen
Call the showallreservedroom() method of hotelroom
Case 'F'
Clear the screen
Call the managerSide() method of hotelroom
Case 'G'
Exit the program
Otherwise
Display "Please enter a valid choice"
End switch
End while
End of function choose.

The above discrption show us what are the ways or process take place when we start using our system.

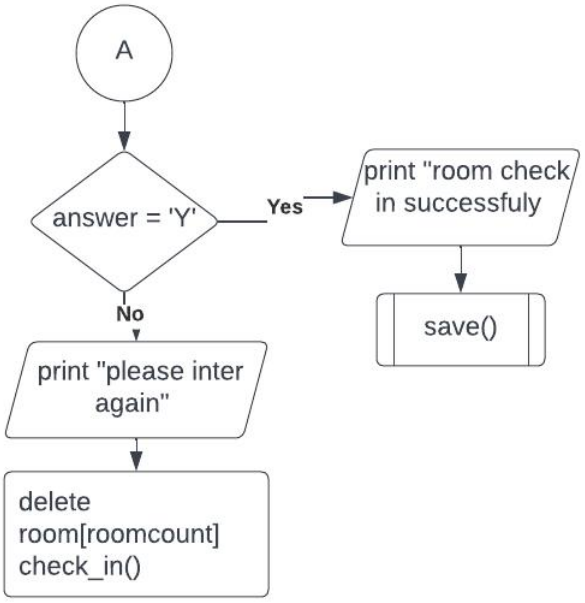
Flow chart

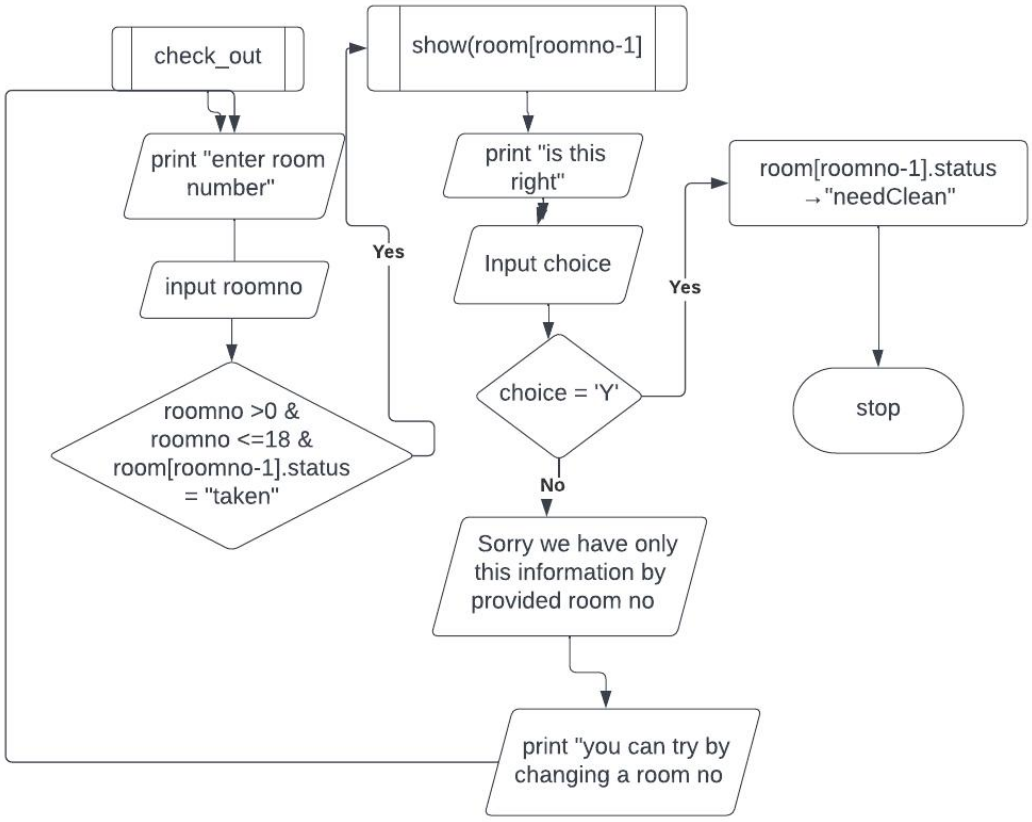
And for more description we tried to build diagram representation of the system with simpel steps.


START
true
Yes
No
what kind of service do you went
A, Room check-in
B, Room check-out
C, See available room
D, To see personal information
E, Show all reserved rooms
F, Manager side
G, Exit
input choice
choice → touppercase(choice)
STOP
choice = A
yes
hotelRoom.check_in
break
no
choice = B
yes
hotelRoom.check_out
break
no
choice = C
yes
hotelRoom.available
break
no
choice = D
yes
hotelRoom.personalInfo
break
no
choice = E
yes
hotelRoom.showallresurvedroom
break
no
choice = F
yes
hotelRoom.managerSide
break
no
choice = G
Yes
EXIT
Please inter a valid choice


room_check in
what kind of room do you went
A, Single bedroom
B, Double bedroom
C, Family standard room
D, Back to main menu
Input choice
choice -> to upercase (choice)
choice = A
yes
book("Single")
Break
no
choice = B
yes
book("Double")
Break
no
choice = C
yes
book("Family")
Break
Option
no
choice = D
Break
No
no
Please enter a valide choice
STOP


book(roomtype)
print "enter person name"
input room[roomcount].cust.name
print "enter last name"
input room[roomcount].cust.fname
print "enter Id no"
input room[roomcount].cust.idno
print "enter sex"
input room[roomcount].cust.sex
print "enter age"
input room[roomcount].cust.age
print "enter address"
input room[roomcount].cust.address
print "enter phone no"
input room[roomcount].cust.phoneno
print "enter from date"
input room[roomcount].cust.from_date
print "enter to date"
input room[roomcount].cust.to_date
rm[roomcount].type=roomtype
rm[roomcount].status="taken"
rm[roomcount].roomno = roomcount+1
rm[roomcount].day = rm[roomcount].cust.to_date - rm[roomcount].cust.from_date
show(room[roomcount
print "is this right"
input answer
answer -> to upercase(answer)
A


A
answer = 'Y'
Yes
print "room check in successfuly
save()
No
print "please inter again"
delete room[roomcount] check_in()



Summary

On this part we try to summary the benefits of using our system. This benefits can helps businesses to grow and strengthen.

1. Streamlined Operations: A digital hotel management system automates and integrates various processes, such as reservations, check-ins, check-outs, housekeeping, and billing. This streamlines operations, reduces manual errors, and enhances overall efficiency.

2. Improved Guest Experience: Digital systems enable faster and more convenient check-ins, personalized guest services, and efficient communication. Guests can easily make online reservations, access their booking details, and provide feedback, leading to a more satisfying and tailored experience.

3. Improved Communication and Collaboration: Digital systems facilitate communication and collaboration among hotel staff across different departments. This leads to better coordination, faster response times, and improved teamwork, resulting in a more efficient and cohesive operation

Conclusion

We have tide to show what are are the capability of our system and benefits so that you can use our system and enjoy the benefits to help and grow.