PROBLEM DEFINITION

A hotel reservation system is software used in the hotel industry to manage room inventory, rates, and bookings. It may be housed within the hotel's property management system (PMS) or may be standalone software connected to the PMS. The CRS acts as the central hub of the reservation ecosystem, which may also include a website booking engine, channel manager, and revenue management system. When integrated, these components work together to help hotels manage room availability, distribution, reservations, and revenue.

This is a project that is developed by *Avishek Karmakar & Raj Sekhar Saha.*

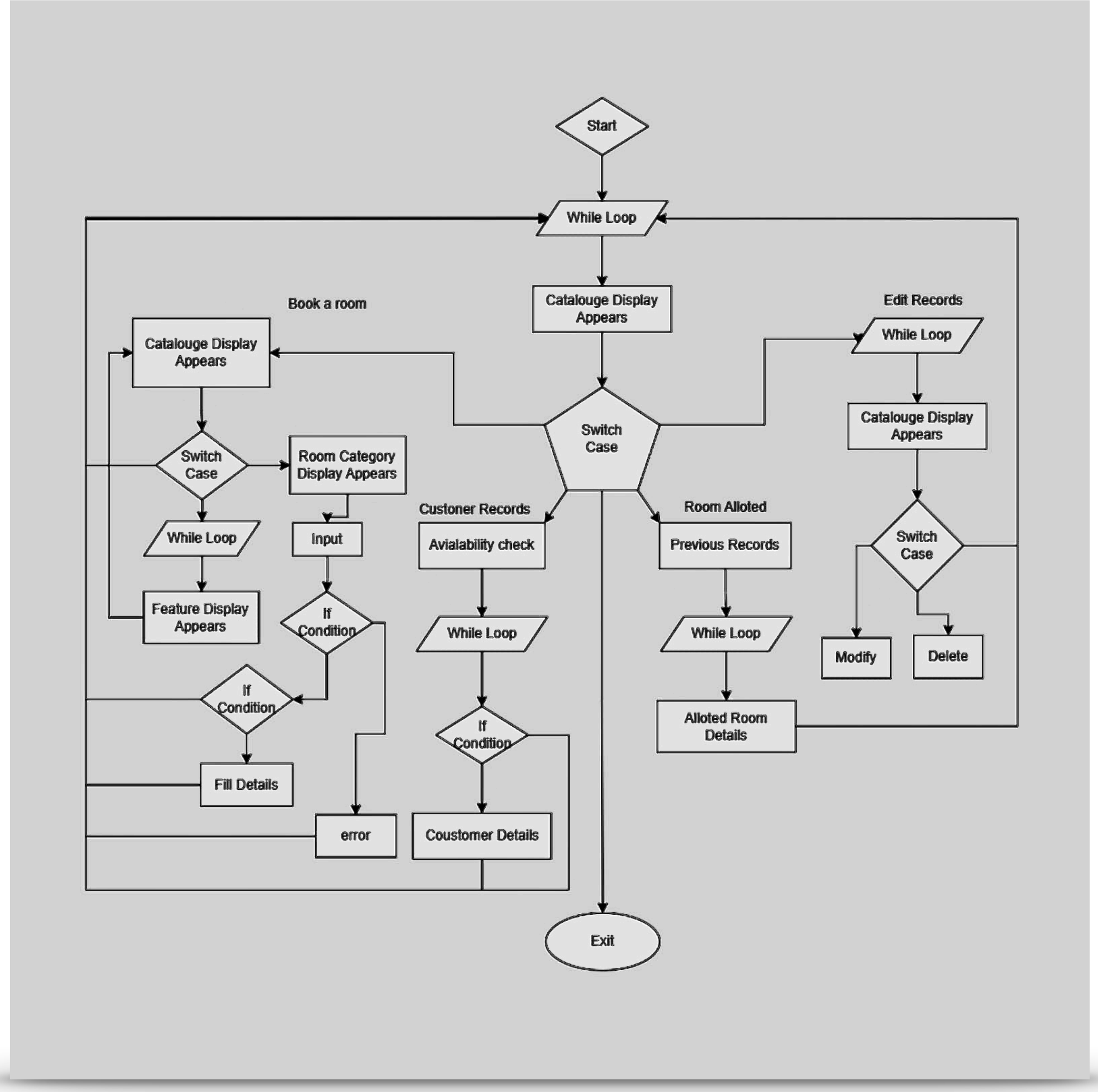
NAME OF THE PROJECT

The name of our project is ***Hotel Reservation System***.

AIM

- Efficiency of booking, especially for repeat customers.
- Ease of information gathering in support of selecting a hotel.
- Accurate, reliable reservations.

FLOWCHART


Start
While Loop
Catalouge Display Appears
Book a room
Edit Records
Catalouge Display Appears
While Loop
Catalouge Display Appears
Switch Case
Switch Case
Room Category Display Appears
Custoner Records
Room Alloted
While Loop
Input
Avialability check
Previous Records
Switch Case
Feature Display Appears
If Condition
While Loop
While Loop
Modify
Delete
If Condition
If Condition
Alloted Room Details
Fill Details
error
Coustomer Details
Exit

LIMITATONS

Although it has some good useful features, it has some limitations too. Here are some flaw in our project

I. Billing can be added.
II. Check-in & Check-out can be added.
III. Could be use some library for implementing GUI(Graphical User Interface).
IV. Sometime having an issue or error in “Room Allotted” options.
V. This program is not support cross-platform(only Windows).
VI. System requirement TURBO C++

FEATURES

There are lots of options when it comes to choosing a hotel reservation system. To find the best solution for your hotel operations, look for the following key features:

- An **open, cloud-based platform** that is easy to implement and maintain, is accessible from any device with an internet connection and can scale with your business.
- **Integration with your hotel's PMS** to provide automatic updates to guest information rather than having to manually transfer data from one platform to another.
- **Two-way connectivity** with a channel manager and revenue management software to optimize distribution, pricing, and inventory.
- A simple, **user-friendly, and commission-free booking engine** that can be customized to match your website design.
- A **built-in rate comparison widget** gives visitors the confidence that booking directly is the best option.
- A **mobile-friendly** booking experience. In 2019, mobile bookings accounted for 43% of online bookings in the U.S., 41% in the U.K., and 79% in China, according to Skift – and those numbers are only climbing.
- **Built-in reporting and analytics** with powerful data insights to help you understand purchase behavior and preferences.
- **Language and currency functionality** to allow international guests to complete a reservation in their preferred language and currency to simplify the booking process.
- A **secure system that adheres to privacy regulations** and ensures guest information and payment details are processed and stored in a secure manner.
- **Customer support** staff who specialize in hospitality will be there when you need them and provide robust onboarding, training, and support services to improve the guest experience.

ALGORITHM

```
STEP 1: START
STEP 2: CREATE A CLASS HOTEL
STEP 3: ADDING SOME VARIABLES room_no, name, address, phone
STEP 4: CREATE A PUBLIC CLASS IN HOTEL
STEP 5: ADDING SOME FUNCTION VARIABLES add_room(),about(),main_menu(),add(),display(),rooms(),edit(),check(int),modify(int),delete_rec(int)
STEP 6: ADD FUNCTION OF hotel::main_menu()
STEP 7: while(choice!=5) then
STEP 8: PRINT "1. Book a room","2. Customer Records","3. Rooms Alloted""4. Edit Records","5. Exit"
STEP 9: INPUT choice
STEP 10: switch(choice) then
STEP 11: case 1: then add(),case 2: then display(),case 3: then rooms(),case 4: then edit(),case 5: then system("exit"),default: then PRINT "Wrong choice !","press enter to continue...!!"
STEP 12: ADD FUNCTION OF hotel::add()
STEP 13: PRINT "1. About","2. Select a room","3. Back"
STEP 14: INPUT choice
STEP 15: switch (choice) then
STEP 16: case 1: then about(), case 2: then add_room(), case 3: then break, default: then PRINT "Wrong choice !","press enter to continue...!!"
STEP 17: ADD FUNCTION OF hotel::display()
STEP 18: INPUT r
STEP 19: while(!fin.eof() then
STEP 20: fin.read((char*)this,sizeof(hotel))
STEP 21: if (room_no==r) then
STEP 22: PRINT " Room no: "," Name: ",<<" Address: "," Phone no: "
STEP 23: if(flag==0)
STEP 24: PRINT " Sorry Room no. not found or vacant....!!" and " Press any key to continue....!!"
STEP 25: then close the file fin.close()
STEP 26: ADD FUNCTION OF hotel::rooms()
STEP 27: initlize "record.txt" by ifstream fin("record.txt")
STEP 28: PRINT "List Of Rooms Allotted" and "Room No.\tName\t\tAddress\t\tPhone No.\n"
STEP 29: while(!fin.eof())
STEP 30: fin.read((char*)this,sizeof(hotel))
STEP 31: PRINT "<<room_no<<"\t\t"<<name<<"\t\t"<<address<<"\t\t"<<phone"
STEP 32: After complete PRINT "Press any key to continue"
STEP 33: close the file fin.close()
STEP 34: ADD FUNCTION of hotel::edit()
STEP 35: take 2 variables choice and r
STEP 36: while(choice!=3)
STEP 37: PRINT "1. Modify Customer Record","2. Delete Customer Record","3. Back"
STEP 38: INPUT choice
STEP 39: switch(choice)
STEP 40: case 1: PRINT "Enter room number " INPUT r modify(r) brake, case 2: PRINT "Enter room no: " INPUT r Delete(r) break, case 3: brake , default PRINT "Wrong choice....!!"
STEP 41: PRINT "press any key to continue....!!
STEP 42: ADD FUNCTION of hotel::add_room()
STEP 43: take 2 variables r and flag
STEP 44: call the record.txt file
STEP 45: PRINT "Total no. of Rooms - 50\n" ,"\n ORDINARY Rooms from : [ 1 - 30 ]\n", "\n LUXURY Rooms from  : [ 31 - 45 ]\n" and "\n ROYAL Rooms from    : [ 46 - 50 ]\n"
STEP 46: take INPUT r
STEP 47: if(r<=50)
STEP 48: check the input with flag variable
STEP 49: PRINT "Sorry..!!!Room is already booked"
STEP 50: else
STEP 51: room_no=r
STEP 52: Take customer name address and phone number
STEP 53: put that data into the record.txt file
STEP 54: then show the user successfull message
STEP 55: else
STEP 56: if user enter more than 50 then it show error
STEP 57: after the function done it PRINT "Press any key to continue"
STEP 58: then close the record.txt file by fout.close()
STEP 59: ADD FUNCTION of hotel::about()
STEP 60: take a variable choice
STEP 61: while(choice!=1)
STEP 62: print about features about rooms
STEP 63: INPUT choice to back
STEP 64: switch(choice)
STEP 65: case 1: run add() function and break, default: break
STEP 66: ADD FUNCTION of hotel::check(int r)
STEP 67: open record.txt file
STEP 68: while(!fin.eof())
STEP 69: READ all the value of that text file
STEP 70: if(room_no==r)
STEP 71: flag=1 and brake
STEP 72: record.txt file close
STEP 73: then return in flag
STEP 74: ADD FUNCTION of hotel::modify(int r)
STEP 75: take long var pos and flag =0
STEP 76: call the record.txt file
STEP 77: while(!file.eof())
STEP 78: pos=file.tellg()
STEP 79: READ the all the char value of the record.txt file
STEP 80: Then all details can be edit like name address and phone number
STEP 81: then all details of the record.txt file can be change
STEP 82: then show the successfull message
STEP 83: if(flag==0)
STEP 84: PRINT "Sorry Room no. not found or vacant...!!"
STEP 85: file close record.txt
STEP 86: ADD FUNCTION of hotel::delete_rec(int r)
STEP 87: flag = 0
STEP 88: take ch char variable
STEP 89: open "record.txt"
STEP 90: CREATE "temp.dat"
STEP 91: while(!fin.eof())
STEP 92: read all char value in the record.txt file
STEP 93: if(room_no==r)
STEP 94: PRINT "NAME" , "ADDRESS" and "PHONE"
STEP 95: Take input to delete or not
STEP 96: if(ch=='n')
STEP 97: read all char value and check if room is vacant or not
STEP 98: else
STEP 99: read all char value of the text file
STEP 100: close all the files
STEP 101: if(flag==0)
STEP 102: PRINT "Sorry room no. not found or vacant...!!"
STEP 103: else
STEP 104: remove record.txt and rename temp.dat into record.txt
STEP 105: int main()
STEP 106: take hotel var h
STEP 107: PRINT "Developed By:","Raj Sekhar Saha & Avishek Karmakar","Press any key to continue....!!"
STEP 108: h.main_menu()
STEP 109: return 0
STEP 110: END
```

REFERENCES

1. https://code-projects.org/simple-hotel-management-system-in-c-with-source-code/

2. https://stackoverflow.com/

3. https://stackoverflow.com/questions/17131407/object-oriented-design-for-hotel-reservation-system

4. https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm

5. https://www.cppbuzz.com/projects/c++/c++-project-on-hotel-management

6. https://github.com/

7. https://www.youtube.com

8. https://www.geeksforgeeks.org

LITERATURE STUDY

- To get the basic idea of our project.
- To gain some knowledge about how the reservation process will works.
- To get knowledge of some features like (Book room, View customer records, Edit records, Delete records).
- To know about the basic functions of libraries.
- To know how to store customer data in a text file.
- To make the program simple so that everyone can easily understand.
- When we facing some issue or error, we go through some websites for understanding the solution of our problems(eg. stackoverflow, tutorials point, geeksforgeeks, YouTube etc.)

GREETINGS

Your words of encouragement, guidance, and advice kept us sane while chasing deadlines. We just wanted to appreciate you and express how glad we are to work under your leadership. We are genuinely proud to be led by such you. Thank you for everything.

PROJECT MEMBERS

- ***Avishek Karmakar***

  Roll No: 22CS022004
  Dept. : B.tech(CSE)

- ***Raj Sekhar Saha***

  Roll No: 22CS022012
  Dept. : B.tech(CSE)

SUBMITTED TO,

Radhakrishna Jana Sir

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Teacher Signature.

CONCLUSION

The project idea is to design of a Command Line Interface hotel reservation system where the receptionist can easily & efficiently book a hotel, check the records. Also to delete or edit records of customers.