# bs

# COMPUTER

**SCIENCE PROJECT**

**ON**

**HOTEL**

**MANAGEMENT SYSTEM**

# CERTIFICATE

Certified to be the bona fide work done by Abhinav Priyadarshi of

class XII – B in Computer Science during the year 2023-24 at

LAXMAN PUBLIC SCHOOL

This Project is absolutely genuine and does not indulge in plagiarism of any kind. The reference taken in making this project has been declared at the end of the project.

Internal signature external signature

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my computer teacher, Abhay Sir, for his vital support, guidance, and encouragement without which this project would not have been completed.

I would also like to thank my parents and those who have supported me during the making of this project.

INDEX

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **CHAPTERS** | **PAGE NO.** |
| **1** | INTRODUCTION | 1 |
| **2** | TECHNOLOGY STACK | 3 |
| **3** | DATABASE DESIGN | 4 |
| **4** | FEATURES | 5 |
| **5** | USER INTERFACE | 8 |
| **6** | FUNCTIONALITY DEMONSTRATION | 10 |
| **7** | SOURCE CODE | 13 |
| **8** | FLOWCHART | 30 |
| **9** | CONCLUSION | 33 |
| **10** | BIBLIOGRAPHY | 34 |

1

§ INTRODUCTION

The Hotel Management System (HMS) stands as a pivotal solution in the hospitality industry, designed to revolutionize and automate various operational facets of a hotel. This system aims to not only optimize internal processes but also elevate the overall customer experience. Employing MySQL for robust database management and Python for a dynamic and scale-able back-end, this project seeks to create an efficient, user-friendly, and comprehensive solution for hotel management.

In this CUI application, we have tried to keep the language highly user friendly so that the user is properly directed through the steps of booking their hotel rooms. At the same time, we have also tried to maintain all the possibilities which may help the user to book his/her hotel efficiently.

1.2|Objective:

The HMS provides facilities like:

* **USER ACCOUNTS:** This feature helps users to create their individual accounts so that their data is secured and bookings are easily accessible.
* **ACCOUNT SETTINGS:** This feature helps the users to change their account details like username, password & email id anytime they wish to. (Note: For changing password, the last password will also be required for confirmation so as to prevent unauthorized access to user accounts)
* **CARD VALIDITY CHECKER:** This feature allows the system to check the validity of a card number based on some internationally accepted predefined structures for credit and debit card numbers.Thus, payments are more secured.
* **INVOICE GENERATOR:** This feature provides the user with adetailed invoice at the end of every transaction. This invoice can also be accessed later by the user through their accounts.
* **BOOKING CANCELLATION:** This feature allows the user to cancel a booking and get the refund for a booking they had done previously, before the booking period is over.

§ TECHNOLOGY STACK:

2

The development of the Hotel Management System (HMS) involved the utilization of a well-thought-out technology stack that combines various programming languages and technologies to create a robust and efficient system. The chosen stack ensures that the HMS not only meets the functional requirements but also offers a user-friendly interface and smooth performance.

**3.1 Programming Languages**

Python: Python was selected as the primary programming language for developing the HMS due to its simplicity, readability, and versatility. Its extensive libraries and frameworks enable rapid development and efficient implementation of various functionalities.

**3.2 Backend Technologies**

MySQL Database: MySQL, a widely used open-source relational database management system, was employed as the backend database for the HMS. Its strong data management capabilities and support for complex queries make it suitable for storing and retrieving book details, member information, and lending records.

Appropriate module for Python were use such that it was integrated to facilitate seamless communication between the Python code and the MySQL database. This abstraction layer simplifies database operations and enhances data integrity.

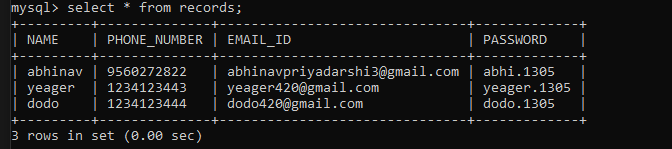
**3.3 Frontend Technologies**

Command-Line Interface (CLI): The HMS employs a command-line interface as its frontend. While graphical user interfaces (GUIs) are more common, the CLI offers a simple and efficient way for librarians to interact with the system. It ensures minimal distractions and focuses on essential functionalities.

§ DATABASE DESIGN:

3

The database design phase is critical for the system's success. Each table will be meticulously crafted to store information about rooms, guests, reservations, invoices, and other relevant entities. Relationships between tables will be established to maintain data integrity and ensure efficient data retrieval. Normalization techniques will be applied to eliminate data redundancy, reduce anomalies, and enhance overall database efficiency.





§ How HMS helps:

4

The Hotel Management System (HMS) presents various advantages that contribute to the efficiency, accuracy, and overall improvement of hotel operations. Here are some key advantages:

**1. Efficient Reservation Management:**

- Streamlined Process: The system automates the reservation process, allowing guests to check room availability, make reservations, and modify or cancel bookings seamlessly.

- Real-time Updates: The system provides real-time updates on room availability, reducing the chances of overbooking and ensuring optimal utilization of resources.

**2. Automated Room Allocation:**

- Optimal Resource Utilization: The system intelligently assigns available rooms based on user preferences and real-time availability, optimizing room allocation and enhancing guest satisfaction.

- Time Savings: Automation reduces manual effort in room allocation, enabling staff to focus on other guest services.

**3. Enhanced Guest Check-in/Check-out Experience:**

- Quick Processing: The system streamlines the check-in and check-out process, minimizing wait times for guests and improving overall service efficiency.

- Real-time Room Status: Updates room status in real-time, ensuring accurate information on room availability and readiness.

**4. Accurate Billing and Invoicing:**

- Transparent Billing: The system generates accurate bills considering room rates, additional services, and the duration of stay, ensuring transparency and reducing billing errors.

- Efficient Invoicing: Automation of the billing process saves time and reduces the likelihood of discrepancies in guest invoices.

**5. Effective Inventory Management:**

- Resource Optimization: The system tracks and manages hotel resources such as toiletries, linens, and other supplies, preventing stockouts and ensuring a consistent level of service.

- Cost Savings: Efficient inventory management minimizes waste and reduces unnecessary expenses related to overstocking or emergency purchases.

**6. Insightful Reporting:**

- Data-driven Decision Making: The reporting feature provides insightful analytics on occupancy rates, revenue, and other key performance indicators, enabling hotel management to make informed and data-driven decisions.

- Performance Evaluation: Regular reports aid in evaluating the hotel's performance, identifying trends, and formulating strategies for improvement.

**7. Security and User Authentication:**

- Secure Access: The user authentication system ensures secure access to the system, protecting sensitive guest and hotel information.

- Role-based Access Control: Different user roles (admin, receptionist, housekeeping) have specific access levels, enhancing security and privacy.

**8. Scalability and Future Enhancements:**

- Scale-able Architecture: The system is designed with scalability in mind, accommodating a growing number of users, rooms, and data over time.

- Future Integration: The system can be easily integrated with online booking platforms, mobile applications, and external systems for payment processing, ensuring adaptability to future needs.

1. **Cost Savings and Resource Optimization:**

- Operational Efficiency: Automation of various tasks reduces manual workload, leading to operational cost savings and improved resource allocation.

- Reduced Errors: Automation minimizes the risk of human errors in tasks such as billing and inventory management, contributing to cost savings and improved accuracy.

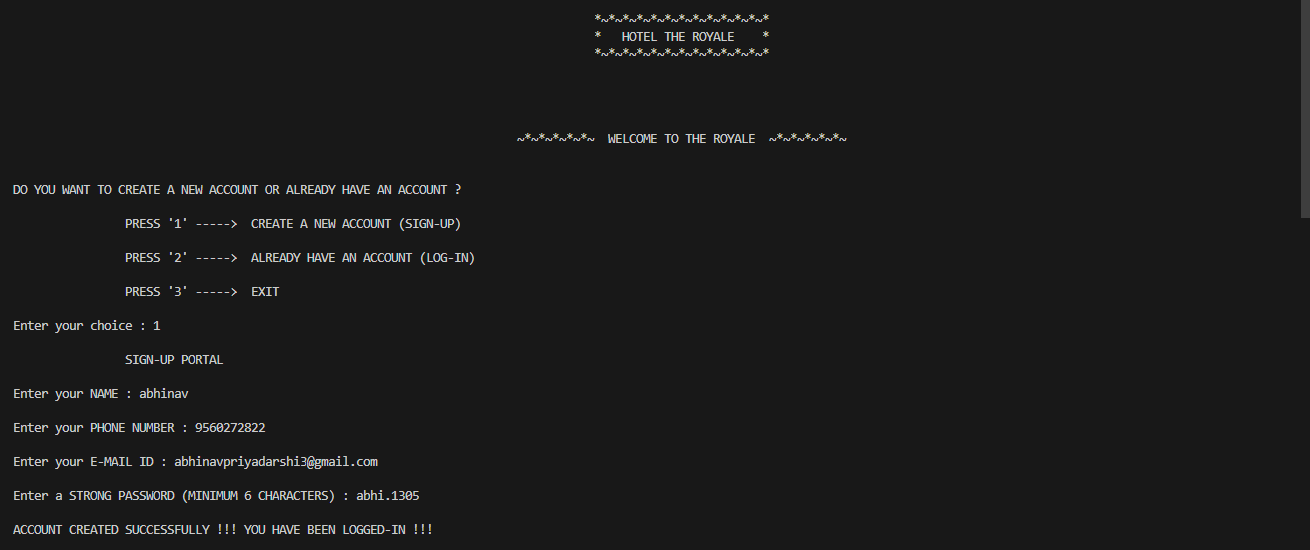
In summary, the Hotel Management System offers a range of advantages that collectively contribute to the efficiency, accuracy, and enhanced guest experience within a hotel. The system's features are designed to address common challenges in hotel management, making it a valuable asset for modern hospitality establishments.

§ User Interface:

5

The user interface (UI) of the Hotel Management System (HMS) has been thoughtfully designed to provide a user-friendly and efficient experience for both the administration and the guests. The UI focuses on simplicity, easy navigation, and clear visual representation of data. Below are screenshots of different screens within the HMS:

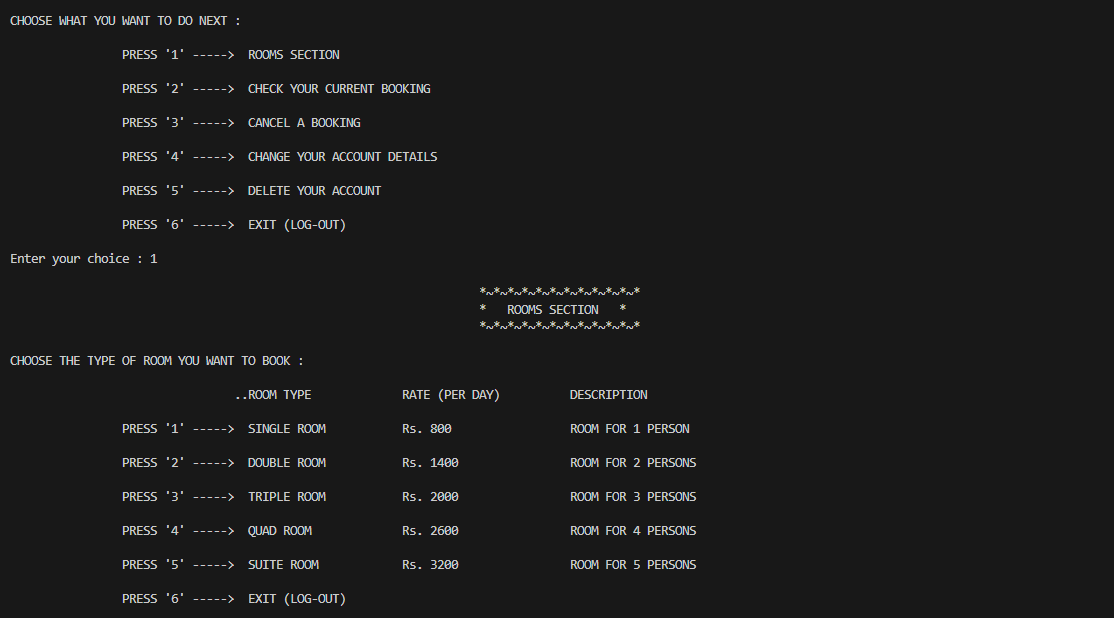
**1 -** Upon launching the HMS, users are greeted with a well structured greeting message and a menu.



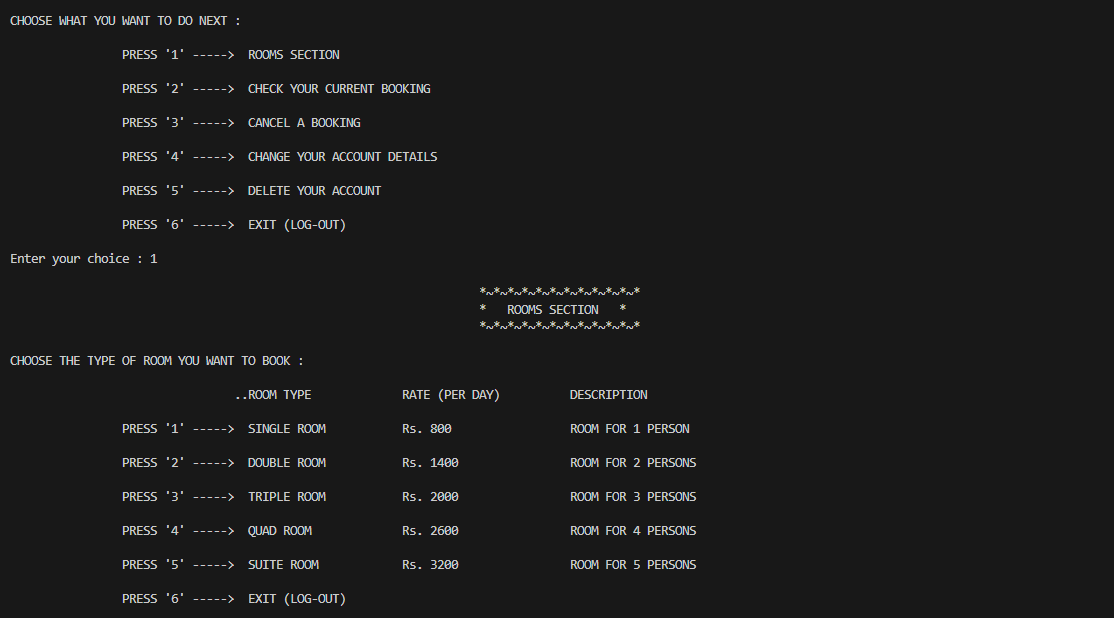
**2 -** menu above allows user sign-in, log-in or exit the menu.

Once signed in below menu pops… displaying further options

for the signed in user.

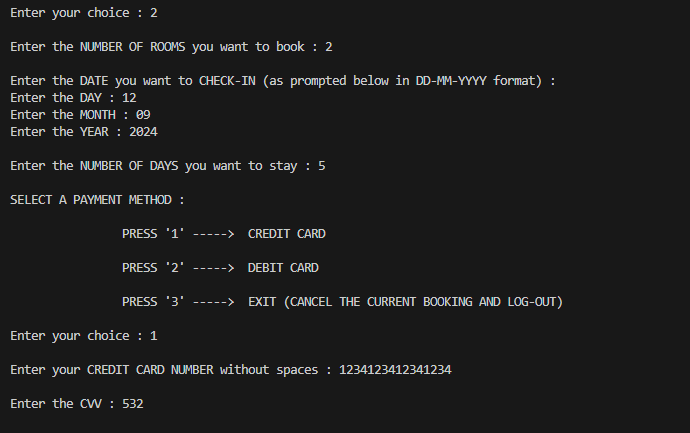


**3 - ROOMS SECTION**

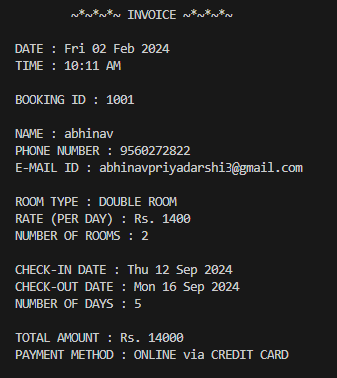


Presents a list of rooms/suites available for the guests, by The Royale.

**4 -** Selecting for their desired type of room, user is asked with continuous prompts about booking details and concludes with a payment method.



**5 -**  Once user has filled in for their desired rooms to check in an invoice is generated. This invoice includes guest’s personal details, and details regarding their booking.



Thus HMS is extremely user friendly and has a counter prompt for wrong inputs, making HMS accountable and interactive.

§ Functionality Demonstration:

6

In this section, we will walk through the key features of the Library Management System, demonstrating how users interact with the system and perform essential tasks.

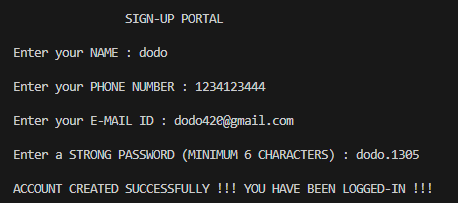
**1 -** Signing-In:

From the menu select the option corresponding to “Create a new account” (1)

A sign up portal will prompt up asking for your details, such as

1. Your name
2. Phone number
3. Email id

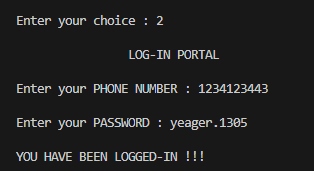
And finally you’ll need to create a password for your account.



**2 -** Logging-In:

On selecting login option (2), a login portal will prompt up asking for your:

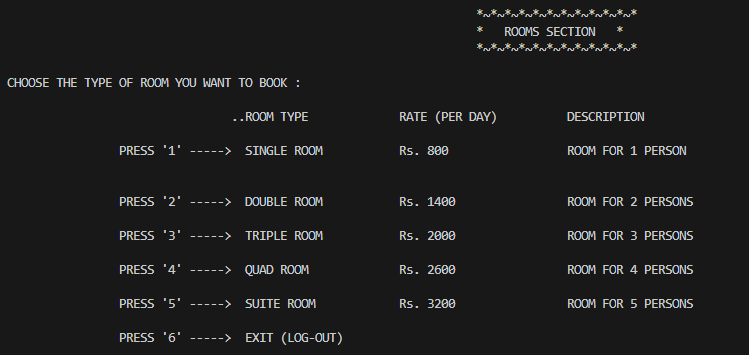
1. Phone number
2. Password

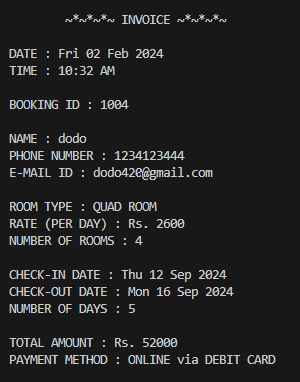
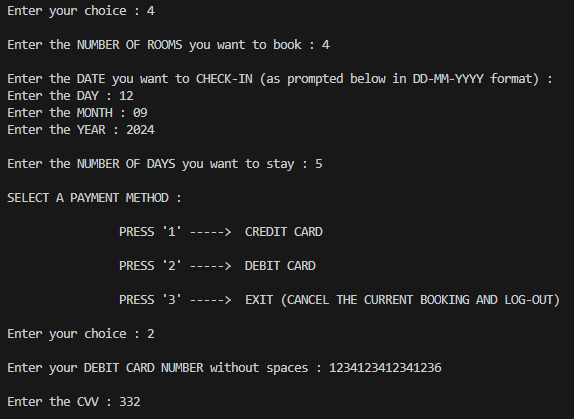


**3 -** Rooms Section:

After user gets logged in they can choose for rooms to book in. This can be done by selecting the option corresponding to so (1)

If done correctly, a Rooms Section will prompt up asking the user to select from the list provided. Upon completing the selection, user agrees upon a payment method and hence booking is concluded.

User receives an invoice including a BOOKING ID which is of great importance.



**4 -**  Checking your current Booking:

This feature requires you to enter your phone number and you booking id so that it gives you a well generated message containing all the details of your recent booking corresponding to the booking id.

**5 -** Canceling a Booking:

Above feature helps the user to cancel their booking just by asking two details, their phone number and booking

**6 -** Change your account details:

Upon selecting option 4 after logging-in, user can change their details by selecting the particular detail they want to edit.

§ SOURCE CODE:

7

# HOTEL MANAGEMENT SYSTEM

import mysql.connector

import datetime

import sys

import csv

db = mysql.connector.connect(host = "localhost", user = "root", password = "1305")

cr = db.cursor()

cr.execute("CREATE DATABASE IF NOT EXISTS HOTEL\_MANAGEMENT")

cr.execute("USE HOTEL\_MANAGEMENT")

cr.execute('''CREATE TABLE IF NOT EXISTS RECORDS(NAME char(50), PHONE\_NUMBER char(10), EMAIL\_ID char(50),

PASSWORD char(50))''')

acc = c = st = na = 0

print('''

\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*

\* HOTEL THE ROYALE \*

\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*

~\*~\*~\*~\*~\*~ WELCOME TO THE ROYALE ~\*~\*~\*~\*~\*~ \n\n ''')

print("DO YOU WANT TO CREATE A NEW ACCOUNT OR ALREADY HAVE AN ACCOUNT ?")

while acc != 1 and acc != 2 and acc != 3 :

print('''

PRESS '1' -----> CREATE A NEW ACCOUNT (SIGN-UP)

PRESS '2' -----> ALREADY HAVE AN ACCOUNT (LOG-IN)

PRESS '3' -----> EXIT \n ''')

acc = int(input("Enter your choice : "))

if acc != 1 and acc != 2 and acc != 3 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!!")

def ph\_valid() :

c = 0

while c != 10 :

c = 0

ph = int(input("Enter your PHONE NUMBER : "))

st = ph

while st > 0 :

c = c + 1

st = (int)(st / 10)

if c != 10 :

print("PLEASE ENTER A 10-DIGIT PHONE NUMBER !!! \n ")

return ph

def sign\_up() :

global ph1

print("\n SIGN-UP PORTAL \n ")

n1 = input("Enter your NAME : ")

print()

ph1 = ph\_valid()

na = 0

cr.execute("SELECT PHONE\_NUMBER FROM RECORDS")

st = cr.fetchall()

c = cr.rowcount

if c > 0 :

c = 0

while c == 0 :

for i in st :

if i[0] == (str)(ph1) :

print("PHONE NUMBER ALREADY REGISTERED !!! \n ")

print(" PRESS '1' -----> REGISTER WITH A DIFFERENT PHONE NUMBER \n ")

print(" PRESS '2' -----> LOG-IN (IF YOU ALREADY HAVE AN ACCOUNT) \n ")

print(" PRESS '3' -----> EXIT \n ")

na = 0

while na != 1 and na != 2 and na != 3 :

na = int(input("Enter your choice : "))

if na != 1 and na != 2 and na != 3 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!! \n ")

if na == 1 :

print()

ph1 = ph\_valid()

break

if na == 2 :

log\_in()

if na == 3 :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!!")

sys.exit(0)

c = 1

if na != 2 :

print()

c = 0

while c != 2 :

c = 0

em1 = input("Enter your E-MAIL ID : ")

for i in em1 :

if i == "@" or i == "." :

c = c + 1

if c < 2 :

print("PLEASE ENTER A VALID E-MAIL ID !!! \n ")

print()

c = 0

while c < 6 :

pswd1 = input("Enter a STRONG PASSWORD (MINIMUM 6 CHARACTERS) : ")

c = len(pswd1)

if c < 6 :

print("PLEASE ENTER A STRONG PASSWORD (MINIMUM 6 CHARACTERS) !!! \n ")

cr.execute("INSERT INTO RECORDS VALUES('" + n1 + "','" + (str)(ph1) + "','" + em1 + "','" + pswd1 + "')")

db.commit()

print("\nACCOUNT CREATED SUCCESSFULLY !!! YOU HAVE BEEN LOGGED-IN !!!")

def log\_in() :

global ph1

print("\n LOG-IN PORTAL \n ")

ph1 = ph\_valid()

na = 0

cr.execute("SELECT PHONE\_NUMBER FROM RECORDS")

st = cr.fetchall()

c = cr.rowcount

if c == 0 :

print("NO SUCH PHONE NUMBER FOUND !!! PLEASE CREATE A NEW ACCOUNT !!! \n ")

print(" PRESS '1' -----> CREATE A NEW ACCOUNT (SIGN-UP) \n ")

print(" PRESS '2' -----> EXIT \n ")

na = 0

while na != 1 and na != 2 :

na = int(input("Enter your choice : "))

if na != 1 and na != 2 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!! \n ")

if na == 1 :

sign\_up()

if na == 2 :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!!")

sys.exit(0)

if na != 1 :

if c > 0 :

c = 0

while c < 3 :

for i in st :

if i[0] == (str)(ph1) :

c = 4

break

c = c + 1

if c < 3 :

print("NO SUCH PHONE NUMBER FOUND !!! PLEASE ENTER PHONE NUMBER AGAIN !!! \n ")

ph1 = ph\_valid()

if c == 3 :

print("NO SUCH PHONE NUMBER FOUND !!! NO MORE TRIES LEFT !!! \n ")

print(" PRESS '1' -----> CREATE A NEW ACCOUNT (SIGN-UP) \n ")

print(" PRESS '2' -----> EXIT \n ")

na = 0

while na != 1 and na != 2 :

na = int(input("Enter your choice : "))

if na != 1 and na != 2 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!! \n ")

if na == 1 :

sign\_up()

if na == 2 :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!!")

sys.exit(0)

if na != 1 :

print()

c = 0

while c < 3 :

pswd2 = input("Enter your PASSWORD : ")

cr.execute("SELECT PASSWORD FROM RECORDS WHERE PHONE\_NUMBER = '" + (str)(ph1) + "'")

st = cr.fetchall()

if st[0][0] == pswd2 :

c = 4

c = c + 1

if c < 3 :

print("PASSWORD NOT MATCHING !!! PLEASE ENTER PASSWORD AGAIN !!! \n ")

if c == 3 :

print("PASSWORD NOT MATCHING !!! NO MORE TRIES LEFT !!! \n ")

print(" PRESS '1' -----> CREATE A NEW ACCOUNT (SIGN-UP) \n ")

print(" PRESS '2' -----> EXIT \n ")

na = 0

while na != 1 and na != 2 :

na = int(input("Enter your choice : "))

if na != 1 and na != 2 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!! \n ")

if na == 1 :

sign\_up()

if na == 2 :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!!")

sys.exit(0)

print("\nYOU HAVE BEEN LOGGED-IN !!!")

if acc == 1 :

sign\_up()

if acc == 2 :

log\_in()

if acc == 3 :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!!")

sys.exit(0)

def find\_date(x, y) :

p = y.day - 1

q = y.month

r = y.year

while x != 0 :

if q in [1, 3, 5, 7, 8, 10, 12] :

while p < 31 and x != 0 :

p = p + 1

x = x - 1

if p == 31 and q == 12 and x != 0 :

r = r + 1

p = 0

q = 1

elif p == 31 and x != 0 :

p = 0

q = q + 1

elif q in [4, 6, 9, 11] :

while p < 30 and x != 0 :

p = p + 1

x = x - 1

if p == 30 and x != 0 :

p = 0

q = q + 1

elif q == 2 :

if (r % 4 == 0 and r % 100 != 0) or r % 400 == 0 :

while p < 29 and x != 0 :

p = p + 1

x = x - 1

if p == 29 and x != 0 :

p = 0

q = q + 1

else :

while p < 28 and x != 0 :

p = p + 1

x = x - 1

if p == 28 and x != 0 :

p = 0

q = q + 1

return(datetime.date(day = p, month = q, year = r))

def card\_valid(cval) :

k = 0

m = 0

cval = (str)(cval)

if len(cval)!=16 :

return False

else :

return True

def date\_make(dmk) :

k = dmk.split("-")

if len(k) == 1 :

dmk = dmk.split(":")

return (datetime.time(hour = (int)(dmk[0]), minute = (int)(dmk[1])))

return (datetime.date(year = (int)(k[0]), month = (int)(k[1]), day = (int)(k[2])))

def bill(bkid) :

a = {}

with open("ROOMS\_SECTION\_DATA.csv", "a+", newline = '') as fh :

fh.seek(0)

bpr = csv.DictReader(fh)

for i in bpr :

if i["BK\_ID"] == (str)(bkid) :

a = i

break

print("\n ~\*~\*~\*~ INVOICE ~\*~\*~\*~ \n")

print("DATE :", date\_make(a["BK\_DATE"]).strftime("%a %d %b %Y"))

print("TIME :", date\_make(a["BK\_TIME"]).strftime("%I:%M %p"))

print("\nBOOKING ID :", a["BK\_ID"])

print("\nNAME :", a["NAME"])

print("PHONE NUMBER :", a["PH\_NO"])

print("E-MAIL ID :", a["EMAIL\_ID"])

print("\nROOM TYPE :", a["RM\_TYPE"])

print("RATE (PER DAY) :", a["RATE\_PD"])

print("NUMBER OF ROOMS :", a["NO\_RM"])

print("\nCHECK-IN DATE :", date\_make(a["CHKIN\_DT"]).strftime("%a %d %b %Y"))

print("CHECK-OUT DATE :", date\_make(a["CHKOUT\_DT"]).strftime("%a %d %b %Y"))

print("NUMBER OF DAYS :", a["NO\_DAYS"])

print("\nTOTAL AMOUNT :", a["TOTAL\_AMT"])

print("PAYMENT METHOD :", a["PAY\_MTD"])

print()

def cancel(bkid) :

md = []

with open("ROOMS\_SECTION\_DATA.csv", "a+", newline = '') as fh :

fh.seek(0)

bpr = csv.DictReader(fh)

for i in bpr :

if i["BK\_ID"] != (str)(bkid) :

md.append(i)

with open("ROOMS\_SECTION\_DATA.csv", "w", newline = '') as fh :

bcan = csv.DictWriter(fh, fieldnames = ["BK\_ID", "NAME", "PH\_NO", "EMAIL\_ID", "PASSWORD", "RM\_TYPE","RATE\_PD",

"NO\_RM", "CHKIN\_DT", "CHKOUT\_DT", "NO\_DAYS", "TOTAL\_AMT", "PAY\_MTD",

"BK\_DATE", "BK\_TIME"])

bcan.writeheader()

bcan.writerows(md)

while True :

print("\nCHOOSE WHAT YOU WANT TO DO NEXT : ")

acc = 0

while acc != 1 and acc != 2 and acc != 3 and acc != 4 and acc != 5 and acc != 6 :

print('''

PRESS '1' -----> ROOMS SECTION

PRESS '2' -----> CHECK YOUR CURRENT BOOKING

PRESS '3' -----> CANCEL A BOOKING

PRESS '4' -----> CHANGE YOUR ACCOUNT DETAILS

PRESS '5' -----> DELETE YOUR ACCOUNT

PRESS '6' -----> EXIT (LOG-OUT) \n ''')

acc = int(input("Enter your choice : "))

if acc != 1 and acc != 2 and acc != 3 and acc != 4 and acc != 5 and acc != 6 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!!")

def rooms() :

rt = ["SINGLE ROOM", "DOUBLE ROOM", "TRIPLE ROOM", "QUAD ROOM", "DELUXE ROOM" ]

rate = ["Rs. 800", "Rs. 1400", "Rs. 2000", "Rs. 2600", "Rs. 3200"]

print('''

\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*

\* ROOMS SECTION \*

\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\* \n ''')

print("CHOOSE THE TYPE OF ROOM YOU WANT TO BOOK : ")

c = 0

while c != 1 and c != 2 and c != 3 and c != 4 and c != 5 and c != 6 :

print('''

\t\t..ROOM TYPE\t\tRATE (PER DAY)\t\tDESCRIPTION

PRESS '1' -----> SINGLE ROOM\t\tRs. 800\t\t\tROOM FOR 1 PERSON

PRESS '2' -----> DOUBLE ROOM\t\tRs. 1400\t\tROOM FOR 2 PERSONS

PRESS '3' -----> TRIPLE ROOM\t\tRs. 2000\t\tROOM FOR 3 PERSONS

PRESS '4' -----> QUAD ROOM\t\tRs. 2600\t\tROOM FOR 4 PERSONS

PRESS '5' -----> SUITE ROOM\t\tRs. 3200\t\tROOM FOR 5 PERSONS

PRESS '6' -----> EXIT (LOG-OUT) \n ''')

c = int(input("Enter your choice : "))

if c != 1 and c != 2 and c != 3 and c != 4 and c != 5 and c != 6 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!!")

if c == 6 :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!! YOU HAVE BEEN LOGGED-OUT !!!")

sys.exit(0)

nr = 0

while nr < 1 or nr > 10 :

nr = int(input("\nEnter the NUMBER OF ROOMS you want to book : "))

if nr < 1 or nr > 10 :

print("PLEASE ENTER A VALID NUMBER OF ROOMS !!! YOU CAN BOOK A MAXIMUM OF 10 ROOMS AT A TIME !!!")

date\_now = datetime.datetime.now().date()

while True :

na = 0

print("\nEnter the DATE you want to CHECK-IN (as prompted below in DD-MM-YYYY format) : ")

dd = int(input("Enter the DAY : "))

mm = int(input("Enter the MONTH : "))

yyyy = int(input("Enter the YEAR : "))

if yyyy < date\_now.year or yyyy > (date\_now.year + 1) :

print("\nINVALID DATE !!! PLEASE CHECK THE RANGE OF YEAR (RANGE OF YEAR - ", date\_now.year, " to ",

(date\_now.year + 1), ") !!!", sep = "")

na = 1

elif mm in [1, 3, 5, 7, 8, 10, 12] :

if dd < 1 or dd > 31 :

print("\nINVALID DATE !!! PLEASE CHECK THE RANGE OF DAY (RANGE OF DAY FOR THAT MONTH - 1 to 31) !!!")

na = 1

elif mm in [4, 6, 9, 11] :

if dd < 1 or dd > 30 :

print("\nINVALID DATE !!! PLEASE CHECK THE RANGE OF DAY (RANGE OF DAY FOR THAT MONTH - 1 to 30) !!!")

na = 1

elif mm == 2 :

if (yyyy % 4 == 0 and yyyy % 100 != 0) or yyyy % 400 == 0 :

if dd < 1 or dd > 29 :

print()

print("INVALID DATE !!! PLEASE CHECK THE RANGE OF DAY (RANGE OF DAY FOR THAT MONTH - 1 to 29) !!!")

na = 1

elif dd < 1 or dd > 28 :

print("\nINVALID DATE !!! PLEASE CHECK THE RANGE OF DAY (RANGE OF DAY FOR THAT MONTH - 1 to 28) !!!")

na = 1

else :

print("\nINVALID DATE !!! PLEASE CHECK THE RANGE OF MONTH (RANGE OF MONTH - 1 to 12) !!!")

na = 1

if na == 0 :

cindt = datetime.date(day = dd, month = mm, year = yyyy)

if cindt <= date\_now :

print("\nINVALID DATE !!! PLEASE ENTER A VALID CHECK-IN DATE STARTING FROM TOMORROW !!!")

else :

break

nd = 0

while nd < 1 or nd > 30 :

nd = int(input("\nEnter the NUMBER OF DAYS you want to stay : " ))

if nd < 1 or nd > 30 :

print("PLEASE ENTER A VALID NUMBER OF DAYS !!! YOU CAN BOOK A ROOM FOR MAXIMUM 30 DAYS AT A TIME !!!")

coutdt = find\_date(nd, cindt)

print("\nSELECT A PAYMENT METHOD : ")

pm = 0

while pm != 1 and pm != 2 and pm != 3 :

print('''

PRESS '1' -----> CREDIT CARD

PRESS '2' -----> DEBIT CARD

PRESS '3' -----> EXIT (CANCEL THE CURRENT BOOKING AND LOG-OUT) \n ''')

pm = int(input("Enter your choice : "))

if pm != 1 and pm != 2 and pm != 3 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!!")

if pm == 1 :

pm = "ONLINE via CREDIT CARD"

cdno = int(input("\nEnter your CREDIT CARD NUMBER without spaces : "))

while card\_valid(cdno) == False :

print("INVALID CREDIT CARD NUMBER !!! PLEASE ENTER CREDIT CARD NUMBER AGAIN !!!")

cdno = int(input("\nEnter your CREDIT CARD NUMBER : "))

cvv = int(input("\nEnter the CVV : "))

while len((str)(cvv)) != 3 and len((str)(cvv)) != 4 :

print("INVALID CVV !!! PLEASE ENTER CVV AGAIN !!!")

cvv = int(input("\nEnter the CVV : "))

elif pm == 2 :

pm = "ONLINE via DEBIT CARD"

cdno = int(input("\nEnter your DEBIT CARD NUMBER without spaces : "))

while card\_valid(cdno) == False :

print("INVALID DEBIT CARD NUMBER !!! PLEASE ENTER DEBIT CARD NUMBER AGAIN !!!")

cdno = int(input("\nEnter your DEBIT CARD NUMBER : "))

cvv = int(input("\nEnter the CVV : "))

while len((str)(cvv)) != 3 and len((str)(cvv)) != 4 :

print("INVALID CVV !!! PLEASE ENTER CVV AGAIN !!!")

cvv = int(input("\nEnter the CVV : "))

else :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!! YOU HAVE BEEN LOGGED-OUT !!! ")

sys.exit(0)

cr.execute("SELECT NAME, EMAIL\_ID, PASSWORD FROM RECORDS WHERE PHONE\_NUMBER = '" + (str)(ph1) + "'")

for i in cr :

nm = i[0]

emid = i[1]

pswd = i[2]

j = 1001

z = 0

with open("ROOMS\_SECTION\_DATA.csv", "a+", newline = '') as rm\_dt :

rm\_dt.seek(0)

hx = csv.DictReader(rm\_dt)

for i in hx :

z = 1

j = j + 1

user\_dt = {"BK\_ID" : j, "NAME" : nm, "PH\_NO" : ph1, "EMAIL\_ID" : emid, "PASSWORD" : pswd, "RM\_TYPE" : rt[c - 1],

"RATE\_PD" : rate[c - 1], "NO\_RM" : nr, "CHKIN\_DT" : cindt, "CHKOUT\_DT" : coutdt, "NO\_DAYS" : nd,

"TOTAL\_AMT" : "Rs. " + ((str)((int)(rate[c - 1][4 : :]) \* nr \* nd)), "PAY\_MTD" : pm,

"BK\_DATE" : datetime.datetime.now().date(), "BK\_TIME" : datetime.datetime.now().time()}

if z == 0 :

wrt = csv.DictWriter(rm\_dt, fieldnames = ["BK\_ID", "NAME", "PH\_NO", "EMAIL\_ID", "PASSWORD", "RM\_TYPE",

"RATE\_PD", "NO\_RM", "CHKIN\_DT", "CHKOUT\_DT", "NO\_DAYS",

"TOTAL\_AMT", "PAY\_MTD", "BK\_DATE", "BK\_TIME"])

wrt.writeheader()

wrt.writerow(user\_dt)

if z == 1 :

wrt = csv.DictWriter(rm\_dt, fieldnames = ["BK\_ID", "NAME", "PH\_NO", "EMAIL\_ID", "PASSWORD", "RM\_TYPE",

"RATE\_PD", "NO\_RM", "CHKIN\_DT", "CHKOUT\_DT", "NO\_DAYS",

"TOTAL\_AMT", "PAY\_MTD", "BK\_DATE", "BK\_TIME"])

wrt.writerow(user\_dt)

print()

bill( j )

print("\nDO YOU WANT TO CONTINUE WITH THIS BOOKING ?")

cnf = 0

while cnf != 1 and cnf != 2 :

print('''

PRESS '1' -----> YES

PRESS '2' -----> NO (CANCEL THE CURRENT BOOKING AND LOG-OUT) \n ''')

cnf = int(input("Enter your choice : "))

if cnf != 1 and cnf != 2 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!!")

if cnf == 1 :

if nr == 1 :

print("\nTRANSACTION SUCCESSFUL !!! YOUR ROOM HAS BEEN BOOKED !!!")

else :

print("\nTRANSACTION SUCCESSFUL !!! YOUR ROOMS HAVE BEEN BOOKED !!!")

if cnf == 2 :

cancel( j )

print("\nYOUR BOOKING HAS BEEN CANCELLED !!!")

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!! YOU HAVE BEEN LOGGED-OUT !!! ")

sys.exit(0)

def check\_booking() :

booking\_id = input("\nEnter the BOOKING ID : ")

e = 1

while True :

a = {}

b = 2

with open("ROOMS\_SECTION\_DATA.csv", "a+", newline = '') as fh :

fh.seek(0)

bpr = csv.DictReader(fh)

for i in bpr :

if i["BK\_ID"] == (str)(booking\_id) :

a = i

break

if a == {} :

b = 0

elif datetime.datetime.now().date() > date\_make(a["CHKOUT\_DT"]) :

b = 1

if b == 0 :

if e == 3 :

print("INVALID BOOKING ID !!! NO MORE TRIES LEFT !!!")

break

print("INVALID BOOKING ID !!! PLEASE ENTER BOOKING ID AGAIN !!!")

booking\_id = input("\nEnter the BOOKING ID : ")

if b == 1 :

print("\nTHIS BOOKING IS NO LONGER AVAILABLE !!!")

break

if b == 2 :

print("\n\nYOUR CURRENT BOOKING IS THIS : ")

bill(booking\_id)

break

e = e + 1

def cancel\_booking() :

booking\_id = input("\nEnter the BOOKING ID : ")

e = 1

while True :

a = {}

b = 2

with open("ROOMS\_SECTION\_DATA.csv", "a+", newline = '') as fh :

fh.seek(0)

bpr = csv.DictReader(fh)

for i in bpr :

if i["BK\_ID"] == (str)(booking\_id) :

a = i

break

if a == {} :

b = 0

elif datetime.datetime.now().date() >= date\_make(a["CHKIN\_DT"]) :

b = 1

if b == 0 :

if e == 3 :

print("INVALID BOOKING ID !!! NO MORE TRIES LEFT !!!")

break

print("INVALID BOOKING ID !!! PLEASE ENTER BOOKING ID AGAIN !!!")

booking\_id = input("\nEnter the BOOKING ID : ")

if b == 1 :

print("\nTHIS BOOKING CAN NO LONGER BE CANCELLED !!!")

break

if b == 2 :

print("\nYOUR BOOKING HAS BEEN CANCELLED !!! YOUR REFUND IS BEING PROCESSED !!!")

cancel(booking\_id)

break

e = e + 1

def account\_details() :

print("\nCHOOSE WHAT YOU WANT TO CHANGE : ")

cnf = 0

while cnf != 1 and cnf != 2 and cnf != 3 and cnf != 4 :

print('''

PRESS '1' -----> CHANGE NAME

PRESS '2' -----> CHANGE E-MAIL ID

PRESS '3' -----> CHANGE PASSWORD

PRESS '4' -----> EXIT (LOG-OUT) \n ''')

cnf = int(input("Enter your choice : "))

if cnf != 1 and cnf != 2 and cnf != 3 and cnf != 4 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!!")

if cnf == 1 :

newnm = input("\nEnter the new NAME : ")

cr.execute("UPDATE RECORDS SET NAME = '" + newnm + "' WHERE PHONE\_NUMBER = '" + (str)(ph1) + "'")

cr.execute("COMMIT")

print("\nYOUR NAME HAS BEEN CHANGED SUCCESSFULLY !!!")

elif cnf == 2 :

c = 0

while c != 2 :

c = 0

newemid = input("\nEnter the new E-MAIL ID : ")

for i in newemid :

if i == "@" or i == "." :

c = c + 1

if c < 2 :

print("PLEASE ENTER A VALID E-MAIL ID !!!")

cr.execute("UPDATE RECORDS SET EMAIL\_ID = '" + newemid + "' WHERE PHONE\_NUMBER = '" + (str)(ph1) + "'")

cr.execute("COMMIT")

print("\nYOUR E-MAIL ID HAS BEEN CHANGED SUCCESSFULLY !!!")

elif cnf == 3 :

while True :

curpswd = input("\nEnter the CURRENT PASSWORD : ")

cr.execute("SELECT PASSWORD FROM RECORDS WHERE PHONE\_NUMBER = '" + (str)(ph1) + "'")

for i in cr :

pswd = i[0]

if pswd != curpswd :

print("CURRENT PASSWORD NOT MATCHING !!! PLEASE ENTER CURRENT PASSWORD AGAIN !!!")

else :

break

c = 0

while c < 6 :

newpswd = input("\nEnter the NEW PASSWORD (MINIMUM 6 CHARACTERS) : ")

c = len(newpswd)

if c < 6 :

print("PLEASE ENTER A STRONG PASSWORD (MINIMUM 6 CHARACTERS) !!!")

cr.execute("UPDATE RECORDS SET PASSWORD = '" + newpswd + "' WHERE PHONE\_NUMBER = '" + (str)(ph1) + "'")

cr.execute("COMMIT")

print("\nYOUR PASSWORD HAS BEEN CHANGED SUCCESSFULLY !!!")

else :

print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!! YOU HAVE BEEN LOGGED-OUT !!!")

sys.exit(0)

def delete\_account() :

print("\nARE YOU SURE YOU WANT TO DELETE YOUR ACCOUNT ?")

cnf = 0

while cnf != 1 and cnf != 2 :

print('''

PRESS '1' -----> YES

PRESS '2' -----> NO \n ''')

cnf = int(input("Enter your choice : "))

if cnf != 1 and cnf != 2 :

print("INVALID INPUT !!! PLEASE ENTER CHOICE AGAIN !!!")

if cnf == 1 :

cr.execute("DELETE FROM RECORDS WHERE PHONE\_NUMBER = '" + (str)(ph1) + "'")

cr.execute("COMMIT")

print("\nYOUR ACCOUNT HAS BEEN DELETED !!!")

if acc == 1 :

rooms()

if acc == 2 :

check\_booking()

if acc == 3 :

cancel\_booking()

if acc == 4 :

account\_details()

if acc == 5 :

delete\_account()

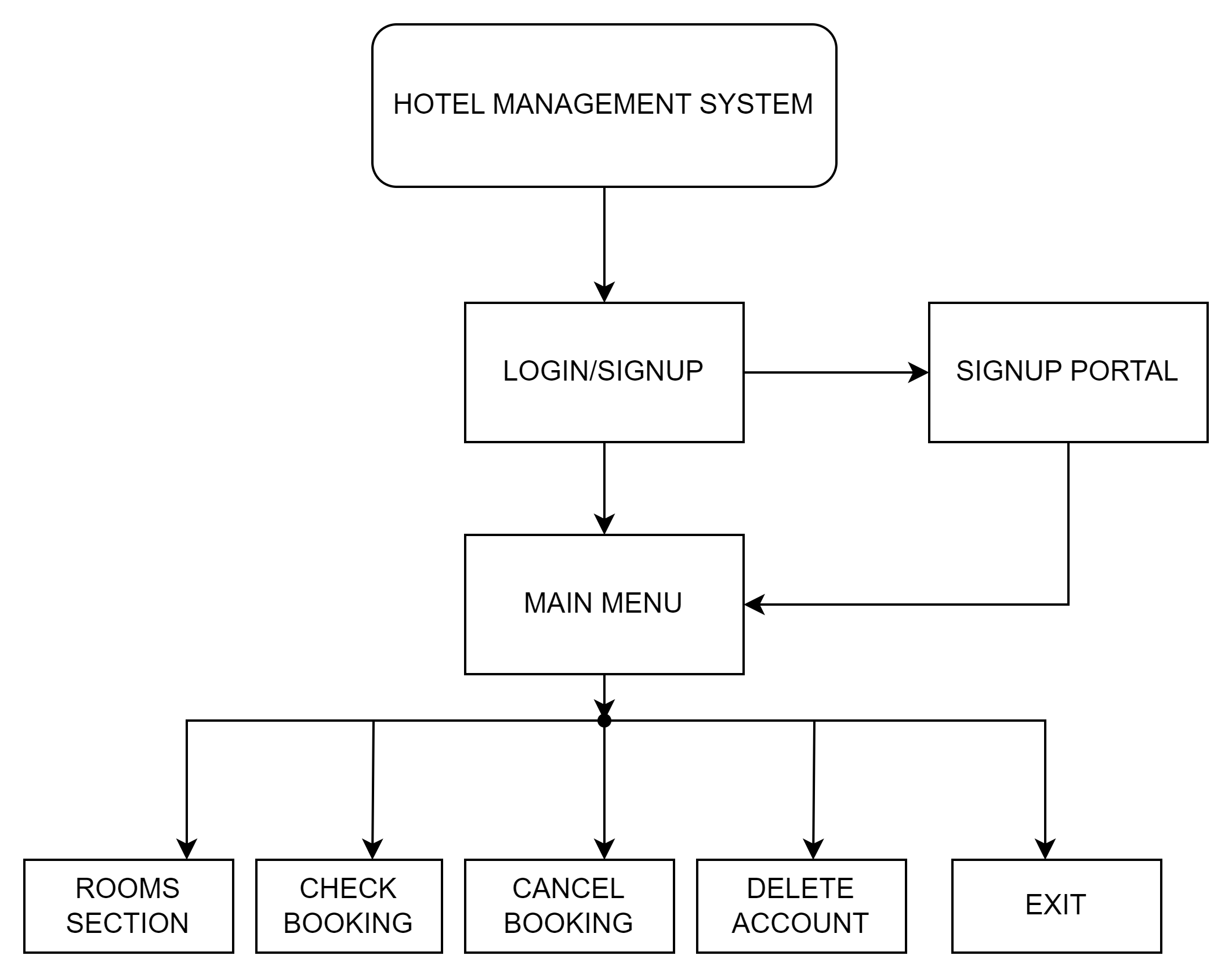
break

if acc == 6 :

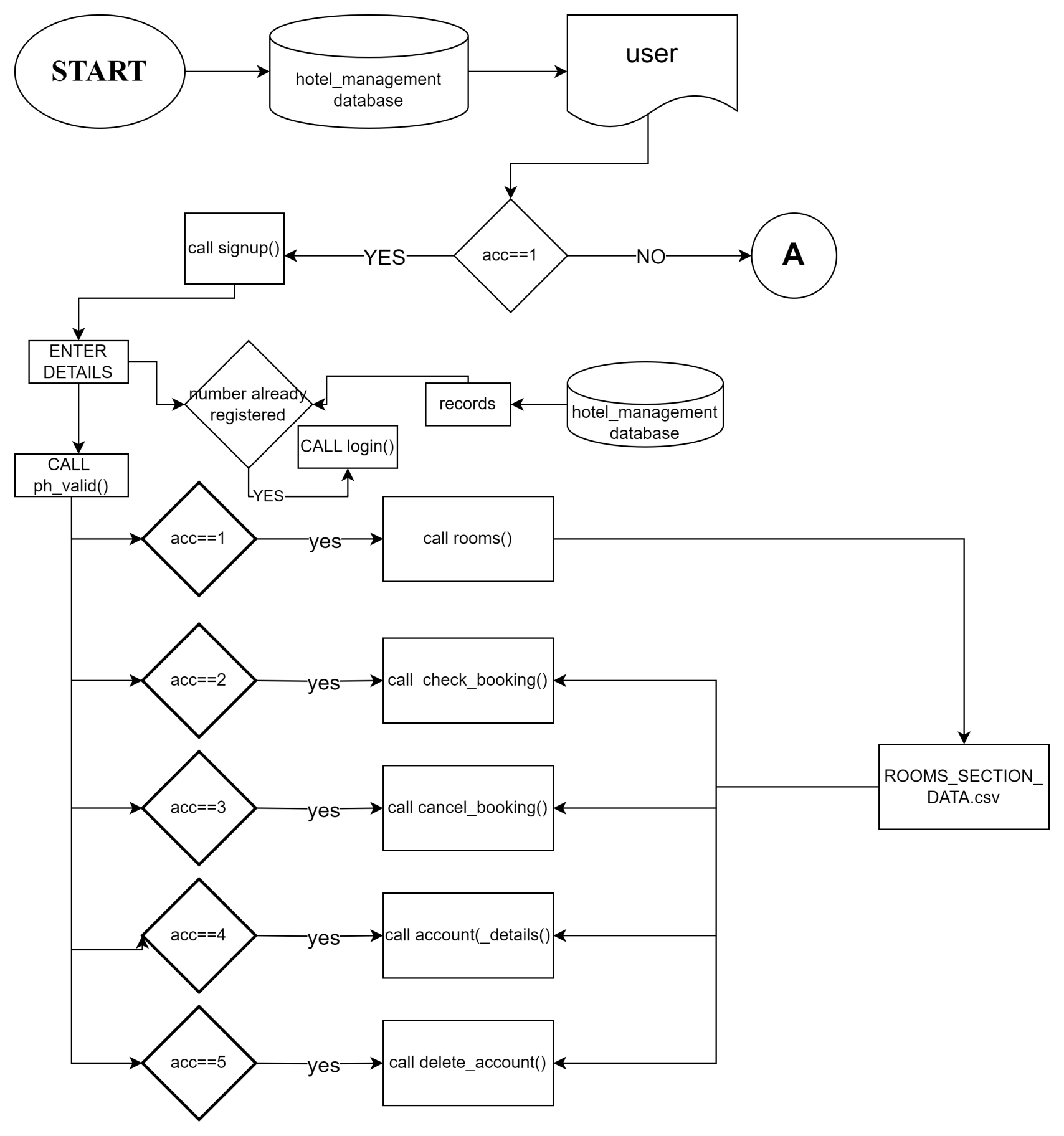
print("\nTHANK YOU FOR VISITING HOTEL THE ROYALE !!! YOU HAVE BEEN LOGGED-OUT !!!")

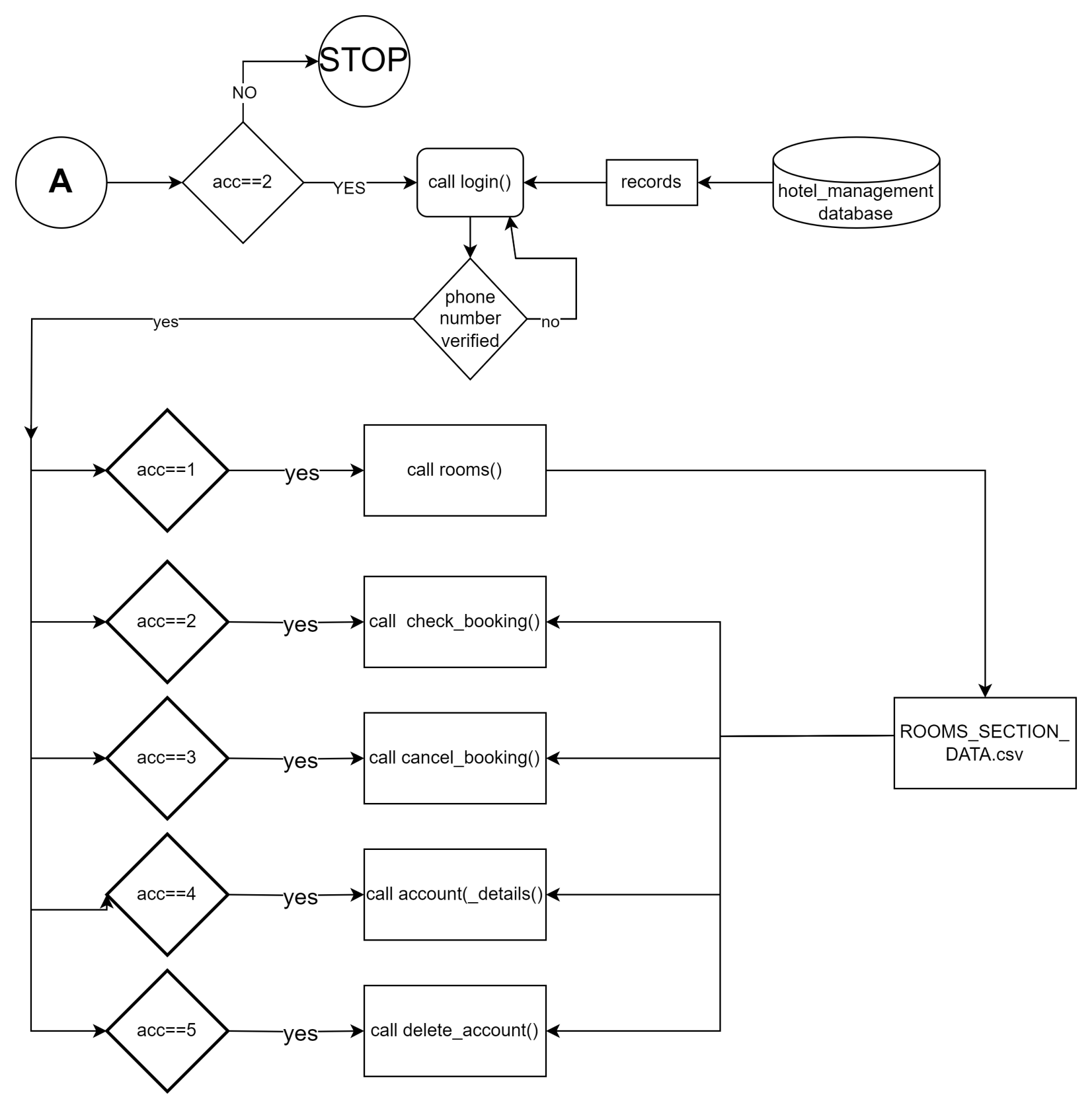
sys.exit(0)

§ **Flowcharts**



8





§ CONCLUSION

9

In conclusion, the development and implementation of the Hotel Management System represent a significant leap forward in optimizing and modernizing hotel operations. This comprehensive solution, driven by a robust combination of MySQL for database management and Python for the backend, brings forth a plethora of advantages that elevate the efficiency, accuracy, and overall guest experience within the hospitality industry.

By automating key processes such as reservation management, room allocation, guest check-in/check-out, billing, and reporting, the Hotel Management System not only streamlines daily operations but also empowers hotel staff to focus on delivering exceptional service. The system's user-friendly interface, coupled with secure user authentication and real-time updates, ensures a seamless experience for both guests and hotel management.

The advantages of the system extend beyond operational efficiency. Accurate billing and invoicing, insightful reporting, and effective inventory management contribute to cost savings and improved resource allocation. The incorporation of a feedback and reviews system fosters continuous improvement, enabling hotels to adapt to evolving guest expectations and enhance their overall reputation.

Looking ahead, the system's scalability and readiness for future enhancements, such as integration with online booking platforms and mobile applications, position it as a forward-thinking solution that can evolve alongside the dynamic landscape of the hospitality industry.

As the project concludes, it marks not just the development of a technological solution but the creation of a tool that can positively impact the way hotels operate. The Hotel Management System stands as a testament to the power of technology in enhancing efficiency, improving guest satisfaction, and contributing to the overall success of hospitality establishments. This project is not merely a conclusion but a stepping stone towards more efficient, customer-centric, and technologically advanced hotel management practices.

§ BIBLIOGRAPHY

10

During the development of this project, I consulted various resources that greatly aided me in understanding and implementing the Library Management System. These resources include:

* "Computer Science with Python" by Sumita Arora, Class 12 CBSE Curriculum.
* "Computer Science for Class 12" by NCERT.
* "Python Programming for Beginners" by Mary Brown.
* W3Schools: An online resource for web development and programming tutorials (https://www.w3schools.com/).
* GeeksforGeeks: A website providing programming solutions and tutorials (https://www.geeksforgeeks.org/).
* Stack Overflow: An online community for programmers to seek and share solutions (https://stackoverflow.com/).
* GitHub: A platform for collaborative coding and version control (https://github.com/).

These resources, along with the textbooks prescribed by the CBSE board, played a crucial role in shaping the project's concepts, design, and implementation