Capgemini

[redacted]

High Level Design & Low Level Design

Index

1. Introduction

The Nature Valley Homestay Billing application is a system that allows the resort to automate the billing process & maintain a bill book of guests in a systematic order in a database removing the traditional way of using bookkeeping. Now with the introduction of this system it simplifies the process as now every customer record is accessible at a single click right from their names to their permanent address, Id proof, the type of room, no. of guests & extra beds. The system helps in calculating the bill when the customer checks out and updating the database.

1.1 Intended Audience: -

The target audience set for this project can be identified as resort/hotel owners who are looking to automate the process of billing & maintaining the records of customer bookings.

1.2 Project Purpose: -

The Nature Valley Homestay billing application is a project that helps us understand the basic concepts of functions, file handling, data structure and Object Oriented Programming. The application will take customer details entered by the receptionist. It will search for rooms & validate the details. If the room is available and details are validated the customer is checked in. We maintain a database of current customers and bookings by displaying, updating and removing by the main menu of this program. On check out if the customer has paid the bill we delete it from the current customers database & update the rooms and bookings database.

1.3 Key Project Objectives: -

a. Allow the Receptionist to enter customer details.

b. Search for rooms & validate details.

c. Check In the customer.

d. Display/Search for customers.

e. Check Out the customer.

f. Display the bill.

g. Delete the customer & vacate the room.

1.4 Project scope : -

This project aims to create the development of an Nature Valley Homestay billing application,Which takes the customer details such as ID, Name, no. of guests, validates the info and checks in the customer & adds it to the customer, booking & rooms database. It can let the user search/view the customers using the customer id. When the customer checks out it can calculate the bill & display it, delete the customer from the database & vacate the room. When the customer pays the bill it will update the booking database.

2. Design Overview: -

- **Billing Application comprises of the following modules in customer database:**

Name of the Module	Check In
Handled by	
Description	The customer gets added in the database

Name of the Module	Check Out
Handled by	
Description	The customer gets deleted from the database

Name of the Module	Search Customer
Handled by	
Description	The customer details are returned from the database

Name of the Module	Customer Report Module
Handled by	
Description	A report of all customers in the resort is displayed.

- **Billing Application comprises of the following modules in Check In:**

Name of the Module	Search for available rooms
Handled by	
Description	It returns if a room is available or not

Name of the Module	Validate the no. of guests & no. of days
Handled by	
Description	It validates the customer details.

Name of the Module	Add the customer
Handled by	
Description	It adds the customer to the database

Name of the Module	Generate customer id
Handled by	
Description	It generates a customer id on check in.

- **Bowling Game comprises of the following modules in show reports:**

Name of the Module	Customer Report Module
Handled by	
Description	It will display all the current customers in the hotel

Name of the Module	Bill Book Module
Handled by	
Description	It will display the bill book of the resort

Name of the Module	Bookings Report
Handled by	
Description	It will display the details of all the customers ever checked in to the hotel.

- **Billing Application comprises of the following modules in Check Out:**

Name of the Module	Display Bill
Handled by	
Description	It returns if a room is available or not

Name of the Module	Check Bill payment
Handled by	
Description	It validates the customer details.

Name of the Module	Delete Customer
Handled by	
Description	It will delete the customer from the database

2.1 Design Objectives:

1. Add all rooms to the rooms database.
2. Start the billing application.
3. Update the database for each check-in and check-out.
4. Displays all the details of the rooms/customers.
5. Modify/Update the customer records.
6. Calculate & display the bill on check-out.

2.2 Design Alternative: -

We have used a Vector data structure to store data i.e.. customer ID, Name, Id proof ,room type, no. of guests, check-in date, no. of days.

2.3 User Interface Paradigms: -

The Bowling game provides an option to Bowlers by generating the score automatically in each frame while playing and keeping the records of players also.

2.4 Validation: -

- The type of room the customer wants should be available.
- The no. of days of stay should be less than 15 and more than 1.

3. SYSTEM ARCHITECTURE: -

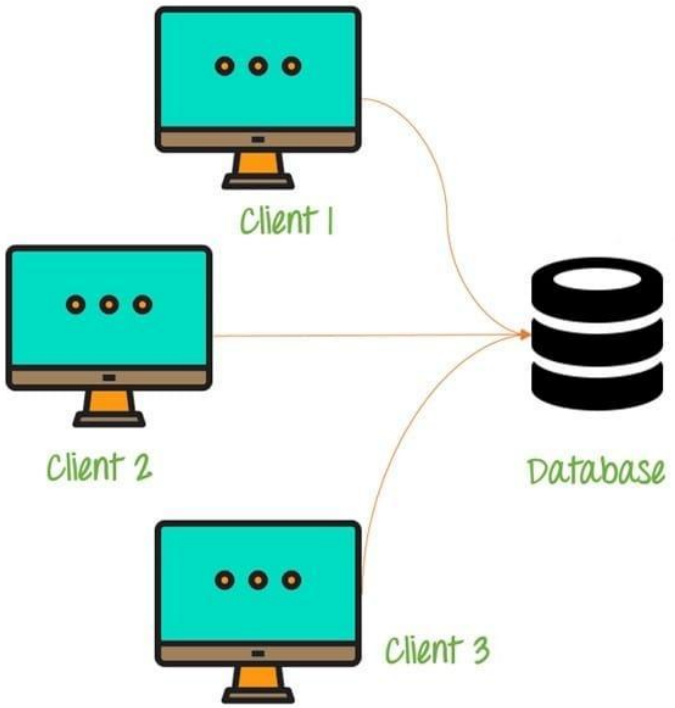
3.1. Database Architecture

The architecture used in this system comprises the database architecture. It is a representation of the database management system design, wherein you can design, develop, implement and maintain the database. This architecture allows dividing the database into different components that can be independently modified, changed, replaced and altered as required for the system.

The database architecture is divided into three tiers namely:-

- 1 - Tier Architecture
- 2 - Tier Architecture
- 3 - Tier Architecture

Our system is based on the Tier 1 model of the database architecture. In this type of model the database is directly available to the user, the user can directly access the database and all of its contents. Which enables the user to directly interact and execute operations.



Some of the characteristics of Database Architecture are:

Self-Describing Nature of a Database System :

- One of the most fundamental characteristics of the database approach is that the database system contains not only the database itself but also an entire definition or description of the database structure and constraints also known as metadata of the database.

Isolation between Data, Programs and Data Abstraction:

- In a traditional file processing system, the structure of database knowledge files is embedded within the application programs, so any

changes to the structure of a file may require changing all programs that access that file.

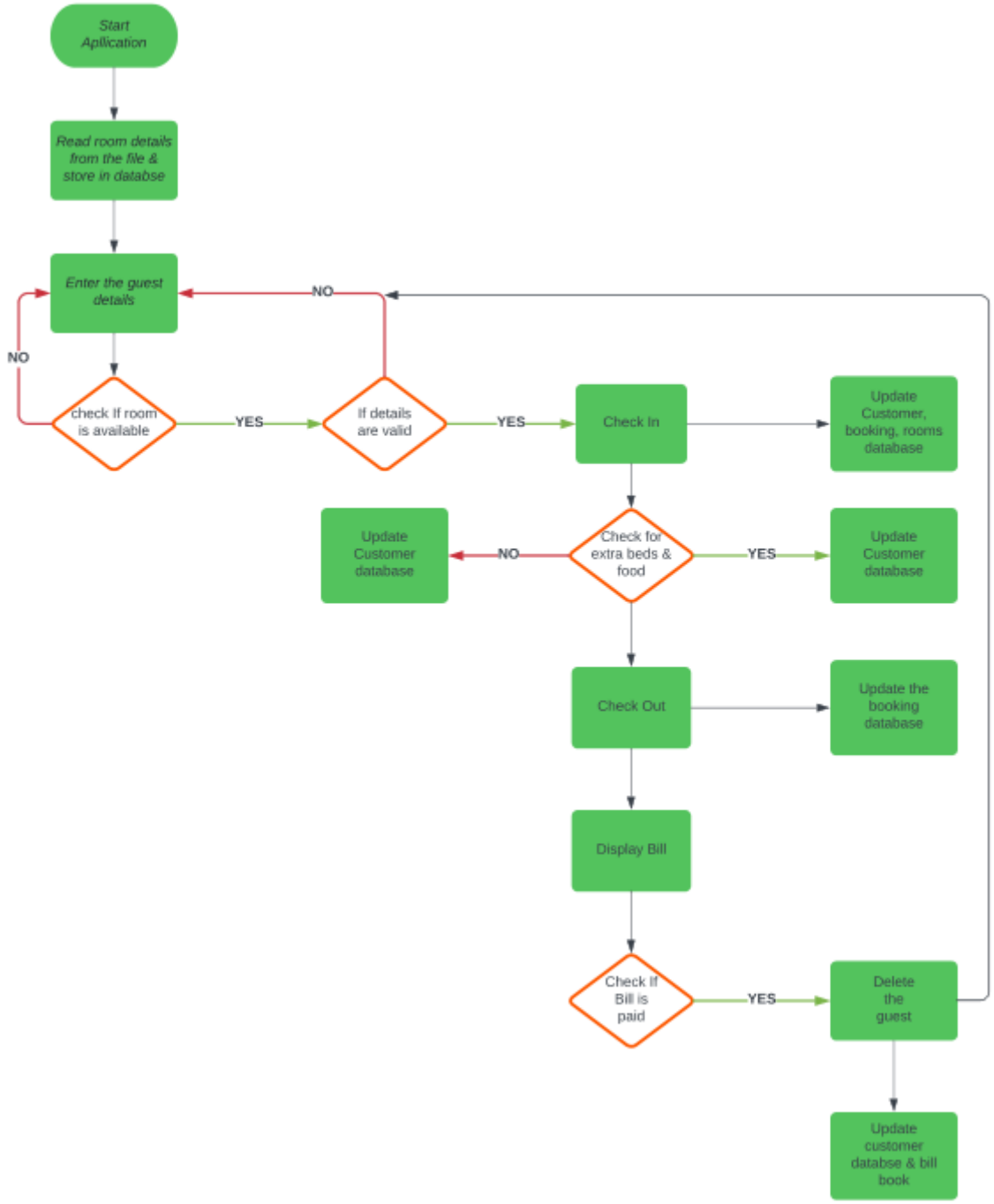
Support for Multiple Views of the Data :

- A database sometimes has many users, each of whom may require a special perspective or view of the database.

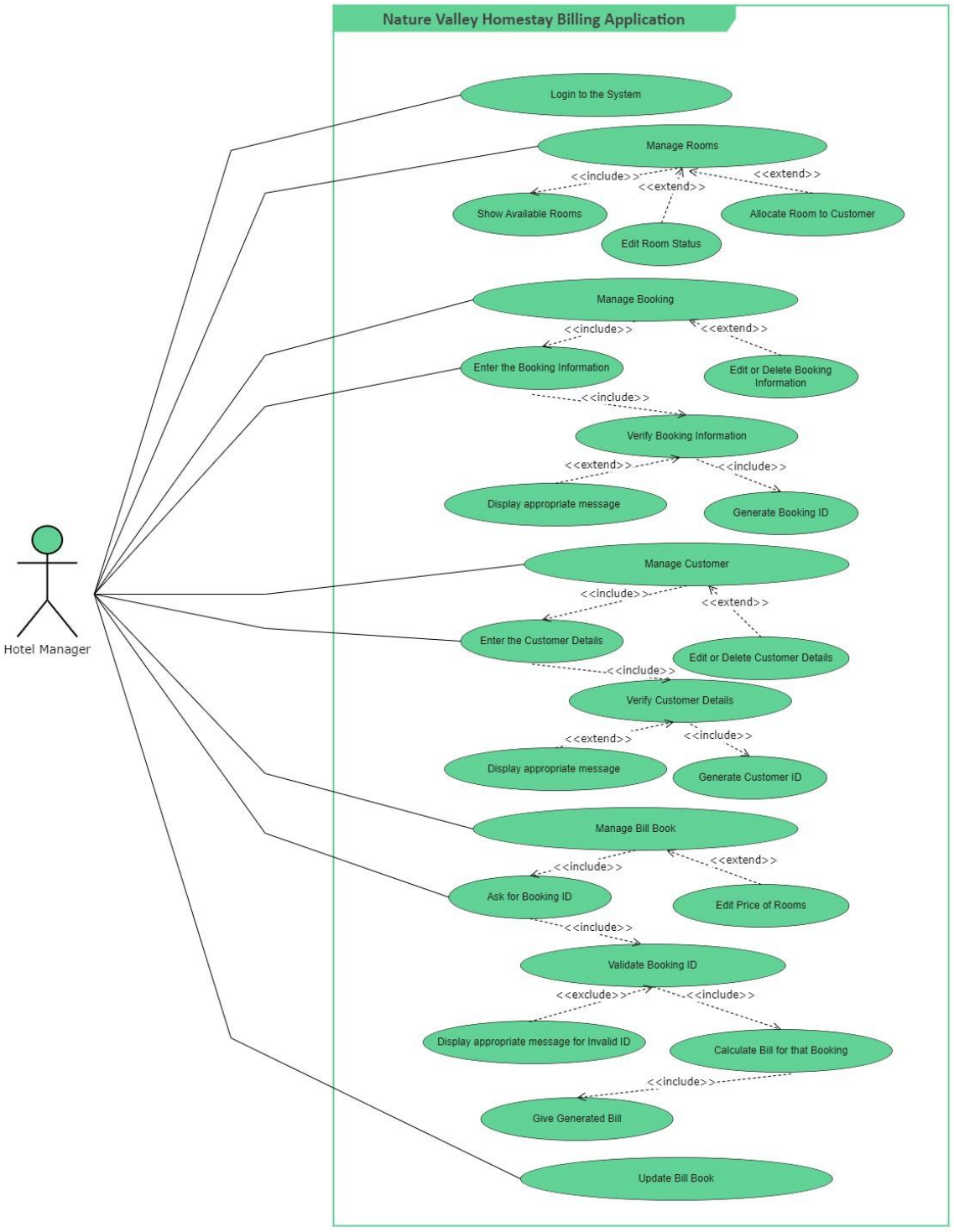
Sharing of knowledge and Multi-user Transaction Processing :

- A multi-user DBMS, as its name implies, must allow multiple users to access the database at an equivalent time or concurrently.

4. DETAILED SYSTEM DESIGN:

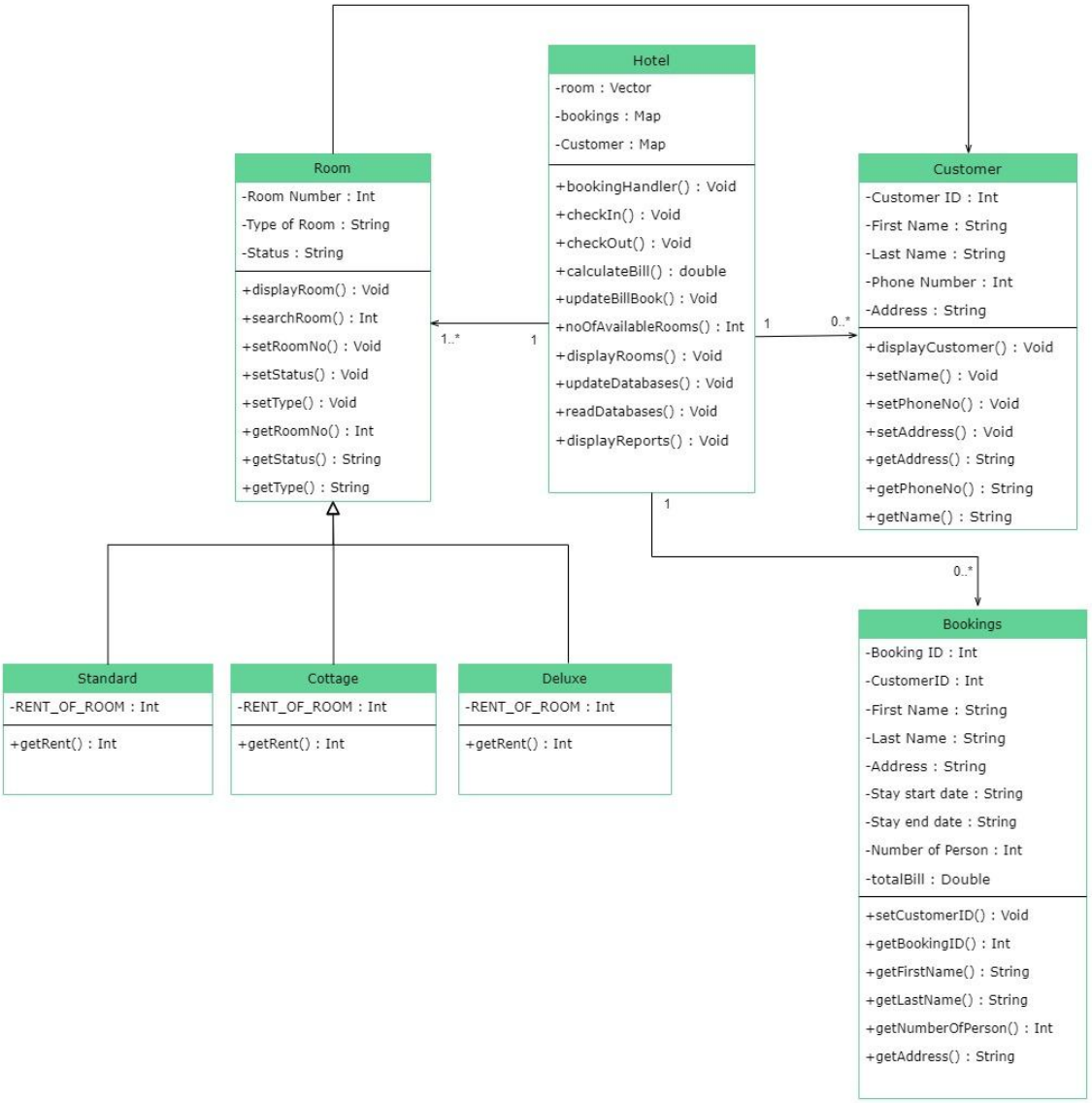


4.6 Flow Chart

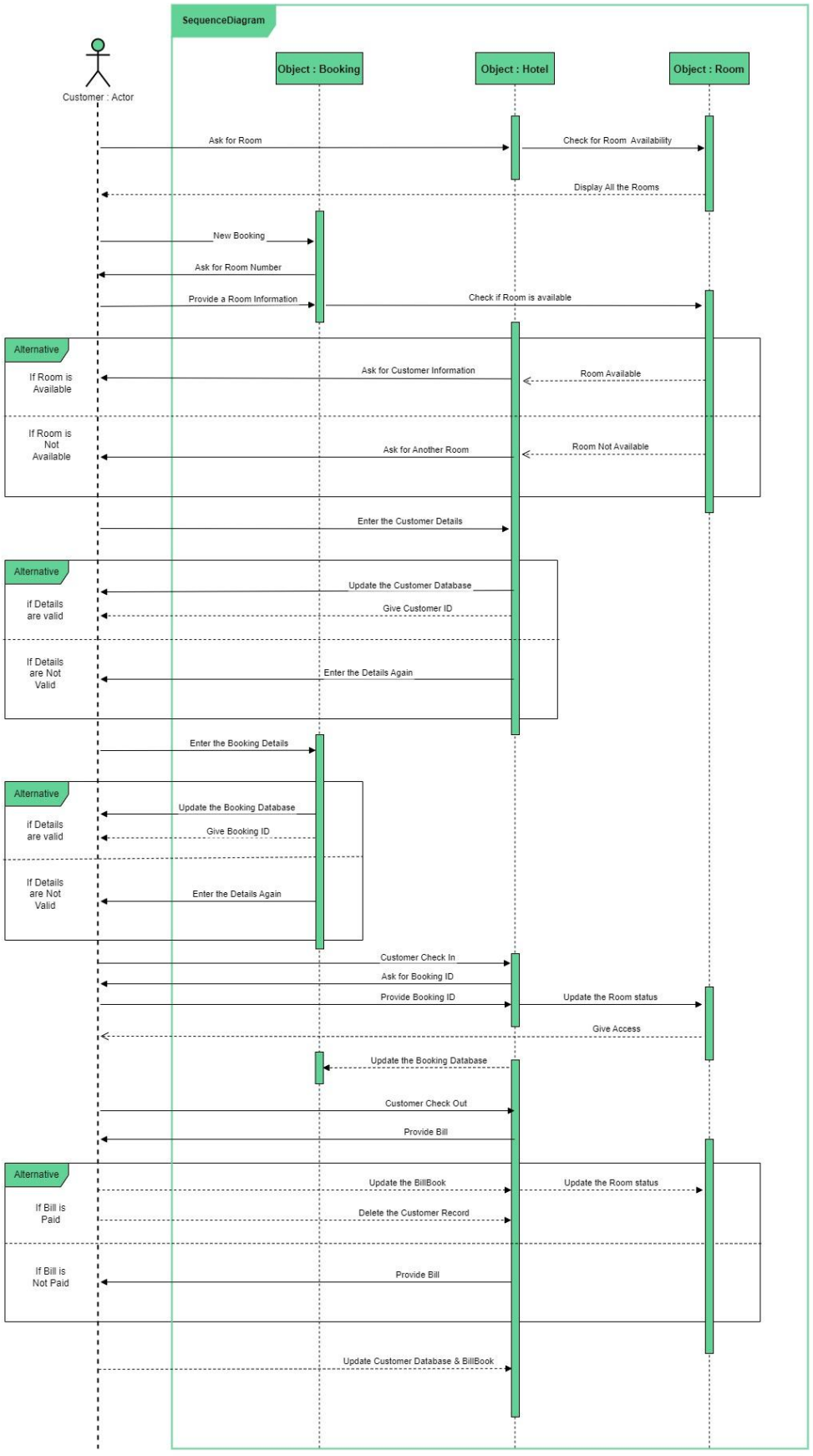


4.7 Use Case Diagram

4.8 Use Case Diagram(Customer)



4.9 Class Diagram


SequenceDiagram
Customer : Actor
Object : Booking
Object : Hotel
Object : Room
Ask for Room
Check for Room Availability
Display All the Rooms
New Booking
Ask for Room Number
Provide a Room Information
Check if Room is available
Alternative
If Room is Available
Ask for Customer Information
Room Available
If Room is Not Available
Ask for Another Room
Room Not Available
Enter the Customer Details
Alternative
if Details are valid
Update the Customer Database
Give Customer ID
If Details are Not Valid
Enter the Details Again
Enter the Booking Details
Alternative
if Details are valid
Update the Booking Database
Give Booking ID
If Details are Not Valid
Enter the Details Again
Customer Check In
Ask for Booking ID
Provide Booking ID
Update the Room status
Give Access
Update the Booking Database
Customer Check Out
Provide Bill
Alternative
If Bill is Paid
Update the BillBook
Update the Room status
Delete the Customer Record
If Bill is Not Paid
Provide Bill
Update Customer Database & BillBook

5. TOOLS REPORT

```
==127628==
==127628== HEAP SUMMARY:
==127628==     in use at exit: 0 bytes in 0 blocks
==127628==   total heap usage: 146 allocs, 146 frees, 146,720 bytes allocated
==127628==
==127628== All heap blocks were freed -- no leaks are possible
==127628==
==127628== For lists of detected and suppressed errors, rerun with: -s
==127628== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
cg81-user17@instance-1:~/CODE/bin$
```

5.1 Valgrind Report

```
==127628==
==127628== HEAP SUMMARY:
==127628==     in use at exit: 0 bytes in 0 blocks
==127628==   total heap usage: 146 allocs, 146 frees, 146,720 bytes allocated
==127628==
==127628== All heap blocks were freed -- no leaks are possible
==127628==
==127628== For lists of detected and suppressed errors, rerun with: -s
==127628== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
cg81-user17@instance-1:~/CODE/bin$
```

6. Requirements Traceability Matrix(RTM)

Req	Design Mapping	Code Mapping	UT Mapping	IT Mapping
NV01	3.1.1	Menu Driven Application		IT_CASE 1
NV02	3.1.2	Customer Database		IT_CASE 5 TO 7
NV03	3.1.3	Bookings Database	Test_Case 3 (21 to 30)	IT_CASE 2 to 4
NV04	3.1.4	Check-ins & Check-outs		
NV05	3.1.5	Rooms Database	Test_Case 2 (11 to 20)	IT_CASE 8
NV06	3.1.6	Bill Book		
NV07	3.1.7	Bill Id		
NV08	3.1.8	Calculate Total Bill	Test_Case 1 (1 to 10)	
NV09	3.1.9	Base Room Fare		
NV10	3.1.10	Validations		IT_CASE 10
NV11	3.1.11	Invalid Entries		IT_CASE 9