VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**HOTEL MANAGEMENT SOFTWARE**

*Instructor:* **NGUYỄN NGỌC PHIÊN**

*Implementer:* **NGUYỄN ANH KHOA –** *SID:* **519H0303**

**NGUYỄN HUỲNH PHÚ ANH–** *SID:* **520h0001**

**NGUYỄN HOÀNG ANH TÂN** *– SID***: 520H0417**

**NGUYỄN TRUNG NAM***– SID***: 519H0321**

**HOÀNG HẢO HIỆP***– SID***: 519H0160**

**HO CHI MINH CITY, 2022**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**HOTEL MANAGEMENT SOFTWARE**

*Instructor:* **NGUYỄN NGỌC PHIÊN**

*Implementer:* **NGUYỄN ANH KHOA –** *SID:* **519H0303**

**NGUYỄN HUỲNH PHÚ ANH–** *SID:* **520h0001**

**NGUYỄN HOÀNG ANH TÂN** *– SID***: 520H0417**

**NGUYỄN TRUNG NAM***– SID***: 519H0321**

**HOÀNG HẢO HIỆP**– SID**: 519H0160**

**HO CHI MINH CITY, 2022**

**THANK YOU**

I would like to express my endless thanks and gratefulness to my supervisor . His kind support and continuous advice went through the process of completion of my thesis. His encouragement and comments had significantly enriched and improved my work. Without his motivation and instructions, the thesis would have been impossible to be done effectively.

**PROJECT COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I hereby declare that this is our own project and is under the guidance of Dr. Nguyen Ngoc Phien. The research contents and results in this topic are honest and have not been published in any publication before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data of other authors, other agencies and organizations, with citations and source annotations.

If I find any fraud, I will take full responsibility for the content of my project. Ton Duc Thang University is not related to copyright and copyright violations caused by me (if any).

*Ho Chi Minh, date*

*Author*

*(Sign)*

**PARTY CERTIFICATION AND ASSESSMENT OF THE TEACHERS**

**The certification part of the instructor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ho Chi Minh city, date: / /2022

(sign)

**The evaluation part of the teacher rated**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ho Chi Minh city, date: / /2022

(sign)

**TABLE OF CONTENTS**

[**1. Introduction**](#_heading=h.2sdr3ry56ui0) **2**

[1.1. Purpose and Scope](#_heading=h.30j0zll) 2

[1.2. Product Overview](#_heading=h.1fob9te) 3

[2. Project Management Plan](#_heading=h.3znysh7) **3**

[2.1. Project Organization](#_heading=h.2et92p0) 3

[2.2. Life Cycle Model Used](#_heading=h.tyjcwt) 4

[2.3. Risk Analysis](#_heading=h.3dy6vkm) 5

[2.4. Hardware and Software Resource Requirements](#_heading=h.1t3h5sf) 6

[3. Requirement Specifications](#_heading=h.2s8eyo1) **7**

[3.1. Functional requirements](#_heading=h.17dp8vu) 7

[3.2. Non-functional requirements](#_heading=h.3rdcrjn) 9

[Requirements for Safety](#_heading=h.lsq0vti0hxsp) 10

[Security Requirement](#_heading=h.nyotpit68hw) 10

[3.3. Use case model](#_heading=h.26in1rg) 12

[3.3.1. Graphical use case model](#_heading=h.lnxbz9) 12

[**4. Architecture**](#_heading=h.udja75qj6dto) **14**

[4.1. Architectural style(s) used](#_heading=h.7ztrwvuqx7ix) 14

[4.2. Architectural model](#_heading=h.1hzr6q8iy5v0) 14

[4.3 . Rationale for your architectural style and model](#_heading=h.hrx9woaxr707) 15

[**5. Design**](#_heading=h.5zblx2gf8bx) **18**

[5.1. Database design](#_heading=h.784l0emy8ptt) 18

[5.2. Static model – class diagrams](#_heading=h.th5e2j3sh13) 19

[5.3. Dynamic model – sequence diagrams](#_heading=h.mwt92rywivet) 19

[**6. Test Plan**](#_heading=h.fbruwdbrbzx4) **22**

[6.1. Requirements/specifications-based system level test cases](#_heading=h.g7dykb953vbo) 22

[**7. Demo**](#_heading=h.gaqlpgaqt29b) **24**

[**7.1. Database**](#_heading=h.58z6c4o1qmzq) **24**

[7.2. Source code](#_heading=h.jt0d97k9pmcy) 24

[7.3. Testing](#_heading=h.36pusdivkfc6) 36

[**References**](#_heading=h.m5b9n2qgfx3y) **40**

# 1. Introduction

## 1.1. Purpose and Scope

The purpose of the hotel booking system is to automate the existing manual system by the help of computerized equipment and full fledged computer software, fulfilling their requirement, so that their valuable information can be stored for a longer period with easy accessing and manipulating of the same . The required software and hardware are easily available and easy to work with. This proposes that efficiency of hotel organizations could be improved by integrating service-oriented operations service-oriented operations with project management principles. Such integration would instill innovation, proactive attitudes and regulated risk-taking needed to pursue ongoing improvement and proactive response to change. By managing each change as a project, embedded in smoothly running operations, hotels would extend their life span by continuously reinventing themselves.

## 1.2. Product Overview

* Hotels Manager Hotels – Administrator can manage hotels that will appear on the site with the hotel name, description, facilities, phone and fax
* Room Types – Administrators can define the type of rooms in the hotels, rooms prices and upload an image for each room.
* Hotel Rooms – For each Hotel the administrator can define the rooms available, rooms number, max occupants and remarks on the specific room.
* Bookings – All booking and reservations made on the site are displayed with all booking details: arrival date, departure date, hotel name, room type, number of passengers, price.
* Administrators can also search for room availability from the administrators panel and do not have to go on the site Reports.

# Project Management Plan

## Project Organization

Meeting: 18 hours in a week via google meet

Work division:

+ draw diagram

+ Make database

+ Coding function

+ Writing report

## Life Cycle Model Used

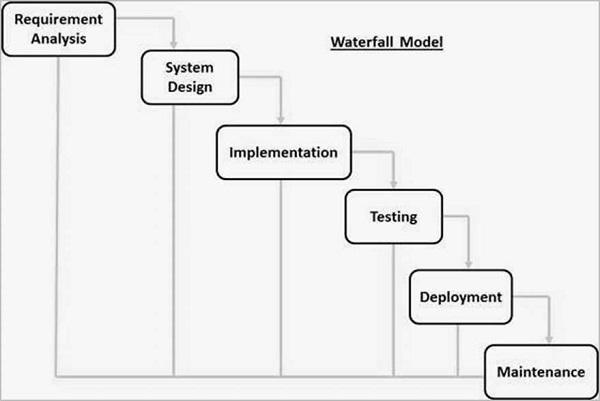


Figure 2.1: Waterfall Model

* **Requirement Gathering and analysis** − During this phase, all potential system requirements are captured and documented in a requirement specification document.
* **System Design** − This phase researches the requirement specifications from the first phase and prepares the system design. This system design aids in the specification of hardware and system requirements, as well as the definition of the overall system architecture.
* **Implementation** − The system is first developed in small programs called units, with input from the system design, and then integrated in the next phase. Unit Testing is the process by which each unit is developed and tested for functionality.
* **Integration and Testing** −After each unit has been tested, all of the units developed during the implementation phase are integrated into a system. Following integration, the entire system is tested for flaws and failures.
* **Deployment of system** − Following the completion of functional and non-functional testing, the product is deployed in the customer environment or released to the market.
* **Maintenance** − There are a few issues that arise in the client environment. Patches are released to address these issues. In order to improve the product, newer versions are released. Maintenance is performed in order to implement these changes in the customer environment.

## Risk Analysis

| Threat source | **Threat action** |
| --- | --- |
| Malicious insider | Personal identification information (PII) browsing  Unauthorized system entry  Employees' unintentional or ill-advised actions that cause unintended physical damage, system disruption, or exposure |
| Technical | Malicious software  Bugs in the system  Failure of a computer, device, application, or protective technology or control, which disrupts or harms operations or exposes the system to harm. |
| Organizational | Improper worker termination and reassignment actions |

Table 2.1: Risk Analysis

## Hardware and Software Resource Requirements

**Software**

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system, which enables the user to interact with several physical hardware devices. The minimum software requirement specifications for developing this project are as follows:

Operating System : Windows 11/10, Linux, MacOS.

Presentation layer : C#, CSS, HTML

Database : SQL Server

Presentation : Power Point

Tool code: VS Code

Documentation Tools : Google docs, word

**Hardware**

The collection of internal electronic circuits and external physical devices used in building a computer is called the Hardware.

The minimum hardware requirement specifications for developing this project are as follows:

Processor : Standard processor with a speed of 1.6 GHz

RAM : 256 MB RAM or more

Hard Disk : 20 GB or more Monitor : Standard color monitor

# Requirement Specifications

## Functional requirements

**1. Password** In this module, this website is for multiple users. If a User enters a password and the software checks its validity. If the password is valid then option is given to change the password, otherwise “Invalid User/ Password” message is displayed. There is an option for password recovery, log out, login, new users sign in. The Administrator can also update changes in the site after login

**2. Creating a new Entity (Hotel, Room, Customers,Members etc.)** This is used to add new employee details, delete entity details and view the details. In that screen, the automatic item is created. In this function, whenever a new entity is required to be added the corresponding forms are opened and the database is manipulated to check whether the data is already existing or not. If it already exists, then it prompts “Entry already existing” and if not then the data is entered with the various validation checks.

**3.Function NEW\_ROOM()** This is the function used to open a new room for a customer so that he/she can assign a separate room . In that screen, the automatic room number is created. After opening a new room for the customer, finally a room is assigned to a customer and the room records are appended in the data file.

**4.Function CHECKIN\_CUSTOMER()** This function is used to admit a customer in our Hotel after entering his all personal details like Name, Address, Phone, Sex and then he/she is assigned a room from NEW\_ROOM() function.

**5.Function CHECKOUT\_CUSTOMER()** This function is used to checkout the customer details from the database. When the user inputs his room number, the same room number will be checked in the database, if the room number is matched in the database, then the customer will be checkout from the database and transferred the record of the checkout to another table of database so that the Hotel Management has the record of customers who have check-out to fulfill his legal liabilities.

**6.Function GENERATE\_BILL()** When any customer checks-out, his/her bill is generated automatically by calculated check-out date minus check-in date and getting multiplied by daily room charge plus other charges and the bill has to be saved in the table in the database.

**7.Function DISPLAY\_RECORD()** This function is used to display all the transactions including the customer name, address, phone, bed number, and doctor assigned to him/her on the screen. This is a global report to display all the transaction records in the screen

## Non-functional requirements

**Performance Requirements**

* Time it takes for the system to respond

Response time is one of the most important performance requirements to consider when creating the Hotel Booking System. The system must be able to respond promptly to the user's inputs and requests; any delays between the user's inputs and the system's response (if necessary) should be kept to a minimum, for example, while retrieving customer information for a booking.

* Efficiency of the System

For a system like this, efficiency is critical; during peak demand periods, the system should be able to always perform at its highest level. Efficiency, in this context, refers to how the system leverages the inputs from the users to generate the output. In some ways, even though a system produces an output, it can still be considered inefficient if the input to output ratio is negative.

* Fast Loading

Along with reaction times, the system's loading speeds must be quick. Users should not have to wait significant amounts of time to have access to information within the system.

### Requirements for Safety

Other than data loss or equipment damage that is not related to the system, there are not many safety needs that can be observed with the system. However, there are still things that can be done to assist mitigate some of the hazards described above, such as keeping several backups of the system to help lessen the risk of data loss or damage on the system.

Another action that may be done is to ensure that the system is only available to authorized members of staff. This is because anyone can steal, tamper with, or delete any of the materials within the system, thus limiting access will ensure that the system is always protected from these threats.

### Security Requirement

* **User Login**To prevent unauthorized personnel from accessing the system, some type of user login method will be implemented. Any member with system access will be asked to have a username and password that allows them to access the system's features.
* **Levels of Access**The system is expected to have several access levels depending on which staff member is accessing the system, such as hotel personnel or management, to assist prevent unauthorized parties from accessing, viewing, or changing parts and information within the system. This means that certain features of the system will be restricted to select individuals based on their level of power inside the firm.
* **Protection of Personal Information**It is critical for the company to maintain the privacy of customer information; this requirement is in line with the regulations of the Data Protection Act 1998, and thus it is critical for the system to be implemented with the goal of maintaining the data confidentiality of the customers on whom they hold information. The system is intended to be able to securely store data and retrieve it when needed.  
  Considering the security standards, certain information within the system may need to be restricted to select members of the company's workforce. At the highest access level, for example, information will only be accessible to authorized personnel such as hotel managers and/or supervisors; if enabled, a password may be required to access such a level.
* **Verification (fingerprints, ID cards)**System authentication is by far one of the most important forms of security that should be integrated into this system, on top of all the other security measures stated above. Returning to the access levels, an authentication mechanism such as ID scanning might be used at various access levels to validate certain staff members and grant them access to specified system elements.

## Use case model

### Graphical use case model

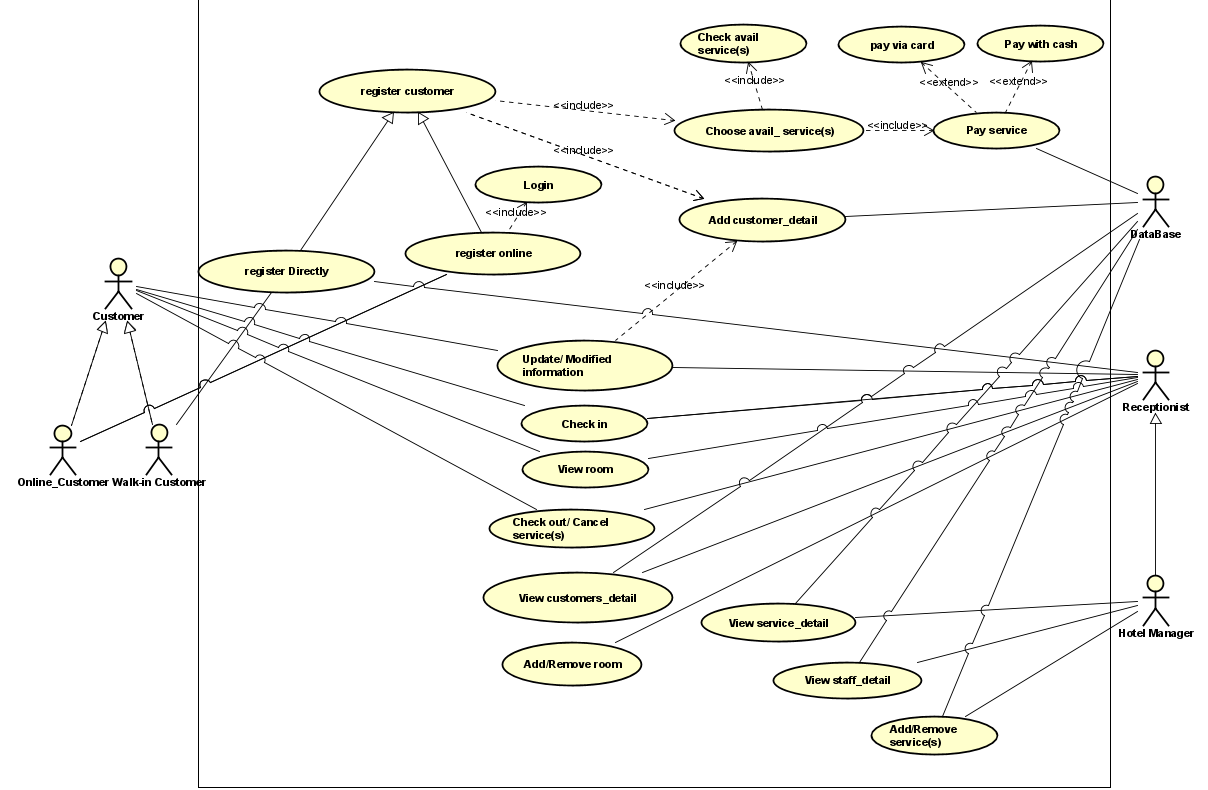


Figure 3.1: Use case

|  |  |
| --- | --- |
| Use case | Register Online |
| Actor | Online\_Customer |
| Description | |
| Register Online | |
| Precondition | |
| Online\_Customer should login to the system using system user account | |
| Flow of events | |
| * If a new customer add customer and login * If exist customer redirect to login | |
| Post condition | |
| System should be save account in database | |

Table 3.1. Register Online use case.

|  |  |
| --- | --- |
| Use case | Register Customer |
| Actor | Online\_Customer, Walk-in Customer |
| Description | |
| Register Customer | |
| Precondition | |
| Online\_Customer need to login in to system  Walk-in Custorm register directly | |
| Flow of events | |
| * Register Customer * Submit | |
| Post condition | |
| Customer update in database | |

Table 3.2. Register customer use case

|  |  |
| --- | --- |
| Use case | Add Customer Detail |
| Actor | Customer, DataBase |
| Description | |
| Add Customer Detail | |
| Precondition | |
| registered customer | |
| Flow of events | |
| * Add information of customer and submit | |
| Post condition | |
| Customer Detail update in Database | |

Table 3.3. Add Customer Detail

|  |  |
| --- | --- |
| Use case | Chose avail\_service(s) |
| Actor | Custormer, DataBase |
| Description | |
| Customer choose available service of hotel | |
| Precondition | |
| List available of service hotel have  Registered customer  Database should login to the system using system user | |
| Flow of events | |
| * Customer select service and submit | |
| Post condition | |
| Customer service update in database | |

Table 3.4 Chose avail\_service(s)

|  |  |
| --- | --- |
| Use case | Pay service |
| Actor | Customer, Database |
| Description | |
| Pay service | |
| Precondition | |
| Customer should reigistered, Customer should book a package. Database should login to the system using system user | |
| Flow of events | |
| * Calculate and generate payment details according to the package * Customer choose payment method * Add payment method * Submit | |
| Post condition | |
| Submit payment details to the system. System sends notification emails to the customer. | |

Table 3.5. Pay service

|  |  |
| --- | --- |
| Use case | Update/Modified information |
| Actor | Receptionist, Customer |
| Description | |
| Update/Modified information customer | |
| Precondition | |
| Resources should add. Customer should be registered. Receptionist should login to the system using system user | |
| Flow of events | |
| * Update information and submit | |
| Post condition | |
| Receptionist should save in database | |

Table 3.6. Update/Modified information

|  |  |
| --- | --- |
| Use case | Check out/ Cancel service |
| Actor | Receptionist, Customer |
| Description | |
| Check out/ Cancel service | |
| Precondition | |
| Customer should be registered. Receptionist should login to the system using system user | |
| Flow of events | |
| * Check-out time * Choose service cancel * Bubmit | |
| Post condition | |
| Receptionist should save in database | |

Table 3.7. Check out/ Cancel service

|  |  |
| --- | --- |
| Use case | Add/Remove room |
| Actor | Receptionist |
| Description | |
| Add/Remove room | |
| Precondition | |
| Receptionist should login to the system using system user | |
| Flow of events | |
| * Select room * Add/remove * Submit | |
| Post condition | |
| Receptionist should save in database | |

Table 3.8. Add/Remove room

|  |  |
| --- | --- |
| Use case | View service\_detail |
| Actor | Hotel maneger, Database |
| Description | |
| Add/edit service items to allocate services to packages | |
| Precondition | |
| Service type and item should be added. Hotel maneger, Database manager should login to the system using system user account | |
| Flow of events | |
| * Select package * Add services to the package * Submit. | |
| Post condition | |
| Service item and booking details updated in the database. | |

Table 3.9. View service\_detail

|  |  |
| --- | --- |
| Use case | View staff\_detail |
| Actor | Hotel maneger, Database |
| Description | |
| Add/edit staff | |
| Precondition | |
| Staff\_detail should be added. Hotel maneger, Database manager should login to the system using system user account | |
| Flow of events | |
| * Select staff * Add/edit * submit | |
| Post condition | |
| Staff detail should save in database | |

Table 3.10. View staff\_detail

|  |  |
| --- | --- |
| Use case | Add/remove service |
| Actor | Hotel maneger, Database |
| Description | |
| Check out/ Cancel service | |
| Precondition | |
| Hotel maneger, Database manager should login to the system using system user account | |
| Flow of events | |
| * Select service * Add/remove * Submit | |
| Post condition | |
| Service should be added in database | |

Table 3.11. Add/remove service

# 4. Architecture

## 4.1. Architectural style(s) used

## The three most prevalent styles of software architecture are data-centric style, layered style, and object-oriented style.

## 4.2. Architectural model

An architectural model (in software) is a rich and rigorous diagram created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem. Software architects use architectural models to communicate with others and seek peer feedback. An architectural model is an expression of a viewpoint in software architecture.

Architectural design models are application domain-specific and the most common two types of domain-specific models are:

* Generic model: These models are abstractions derived from a number of real systems and encapsulated the characteristics of these systems. This type of model usually follows a bottom-up approach.
* Reference models: These models provide information regarding the class of the system. They are derived from the application domain rather than from existing systems. It usually follows the top-down approach. It provides a comparison between different Software architectures.

Advantages of Architectural Design in Software Engineering:

* Architectural design works as a tool for stakeholder communication. It is used as a support or roadmap in the discussion with system stakeholders
* It is used for system analysis. Architectural design is used to analyze whether the system will be able to meet its non-functional requirements or not.
* It facilitates large-scale re-use. The software architecture that is the output of the architectural design process can be reused across a range of the system.

## 4.3 . Rationale for your architectural style and model

When developing a software system, its architecture must be considered so that it can be understood, updated, and improved. In general, considering the architectural artifacts is not enough. The reasons, assumptions and justifications bore in mind by the architects during the architecture design stage must be also known. Nevertheless, not all aspects analyzed during the design process can be identified, especially all those alternatives that were evaluated and rejected. In the present contribution, a model to represent the rationale generated by architects during the architectural design is proposed so that it can last over time and it can be retrieved, analyzed and reused whenever necessary. The model includes concepts representing architectural artifacts, reasons, assumptions, and decisions and reasoning elements status.

* Technology:

1. Brainstorming
2. Mind-Map
3. Design pattern
4. ASP.NET C#

* software:

visual studio

Microsoft SQL Server Management Studio 19

Chrome

* hardware: First, a couple of overall thoughts on computers.

Windows Computer

|  |  |
| --- | --- |
| Memory  16GB minimum | Hard Drive  512GB minimum |
| CPU   * Intel i5 minimum * Intel i7 (or greater) preferred * AMD Ryzen 5 minimum * AMD Ryzen 7 (or greater) preferred | Operating System  Windows 10 Version 21H2 (November 2021 Update) or later |

Table 4.1. Windows Computer

# 5. Design

## 5.1. Database design

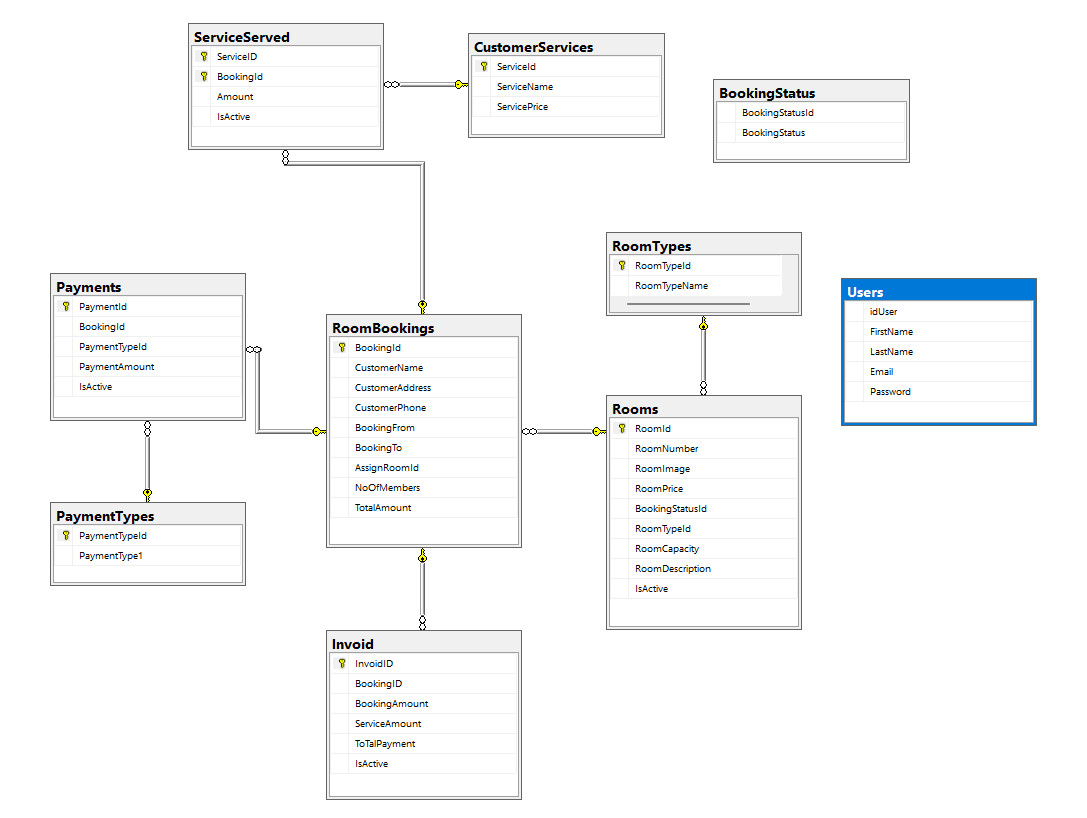


Figure 5.1. Database design

## 5.2. Static model – class diagrams

A class diagram describes the static structure of the symbols in your new system. It is a graphic presentation of the static view that shows a collection of declarative (static) model elements, such as classes, types, and their contents and relationships. Classes are arranged in hierarchies sharing common structure and behavior, and are associated with other classes

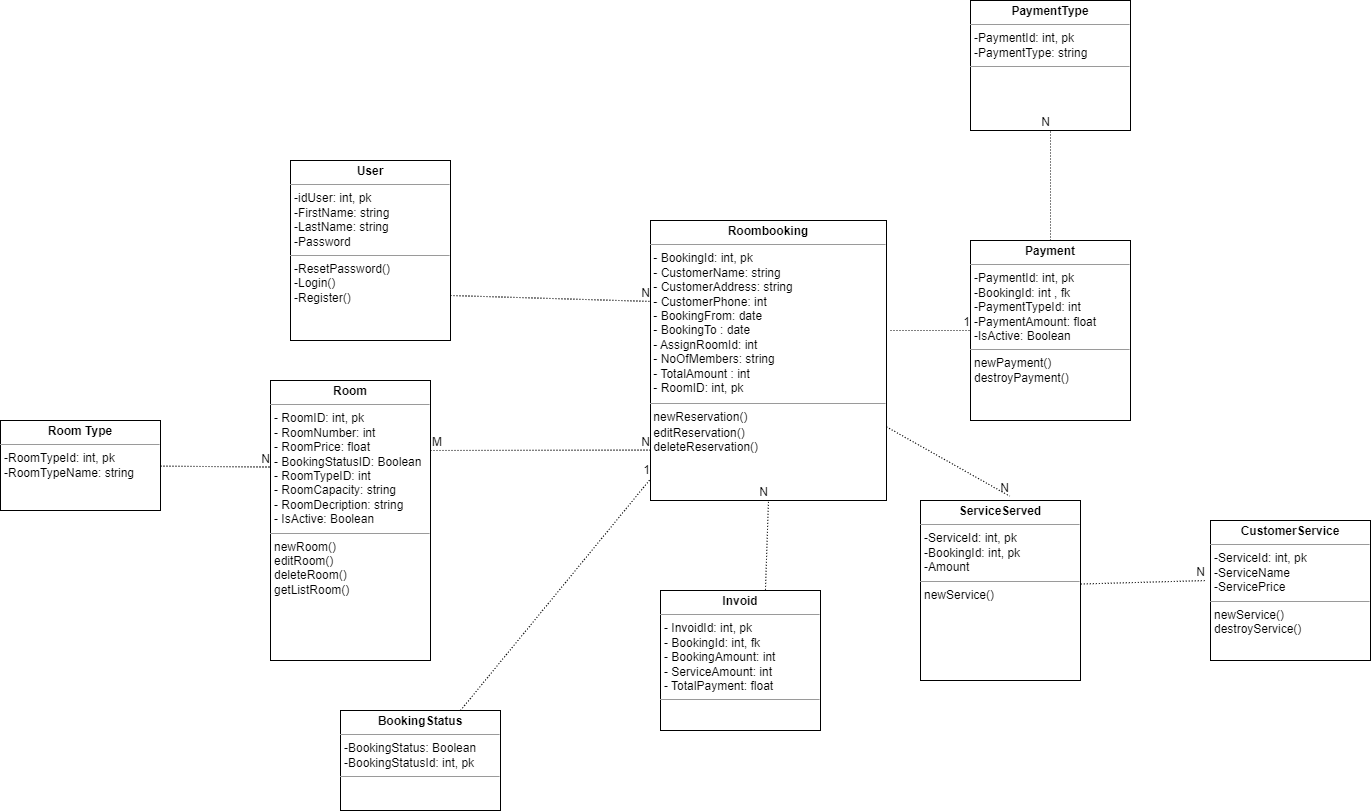


Figure 5.2. Class diagram

## 5.3. Dynamic model – sequence diagrams

​​

Login and Register.

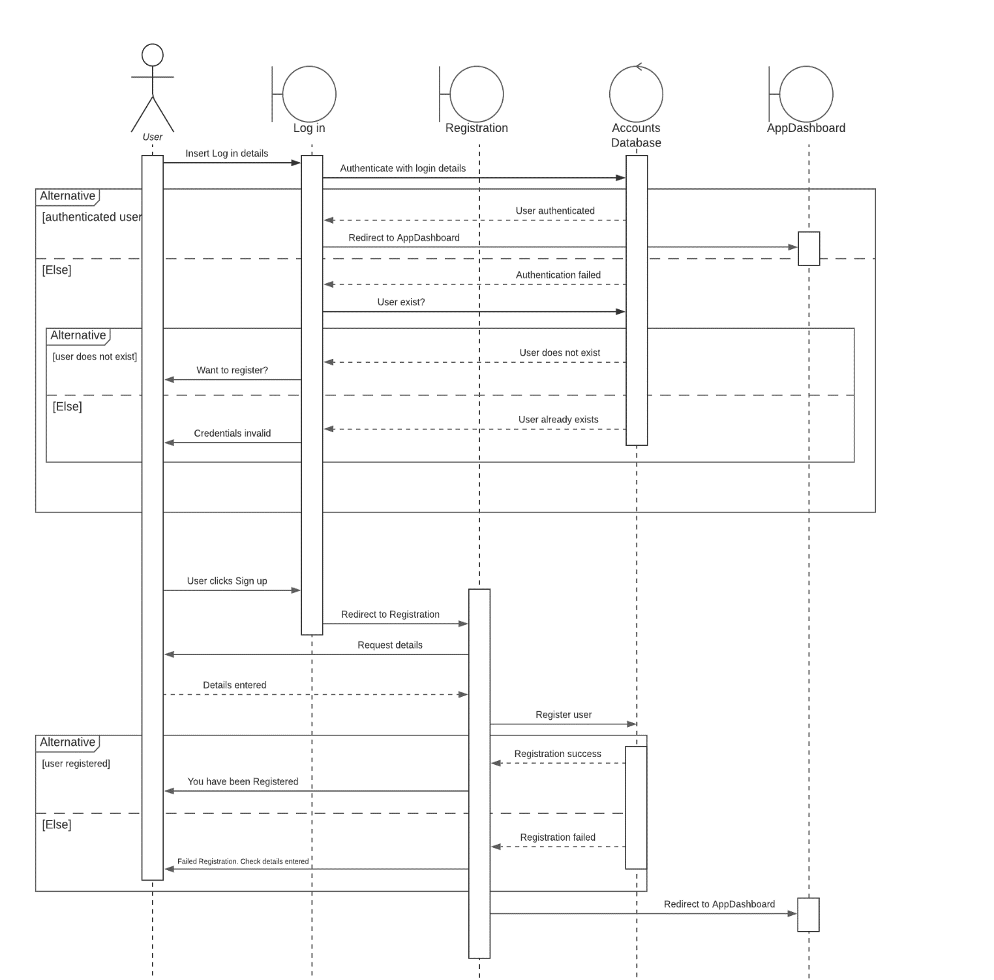


Figure 5.3. Login and Register.

Service

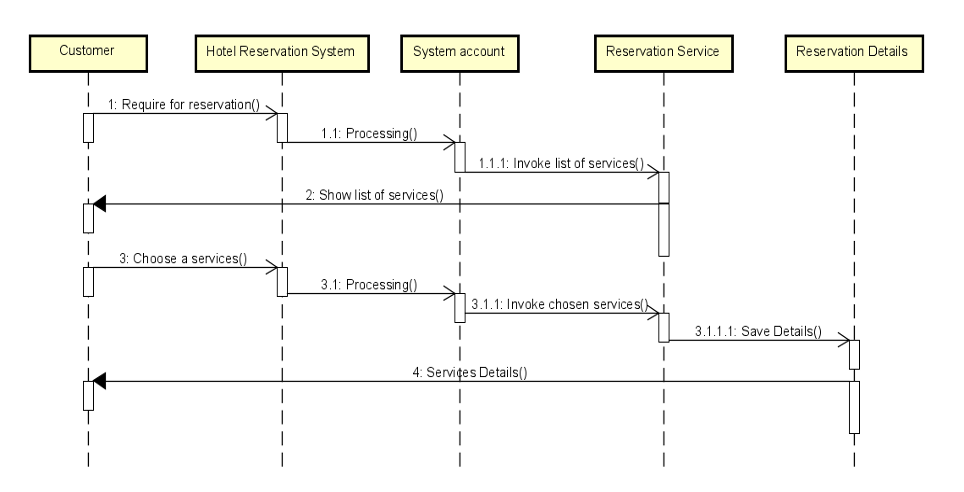


Figure 5.4. Service.

Booking

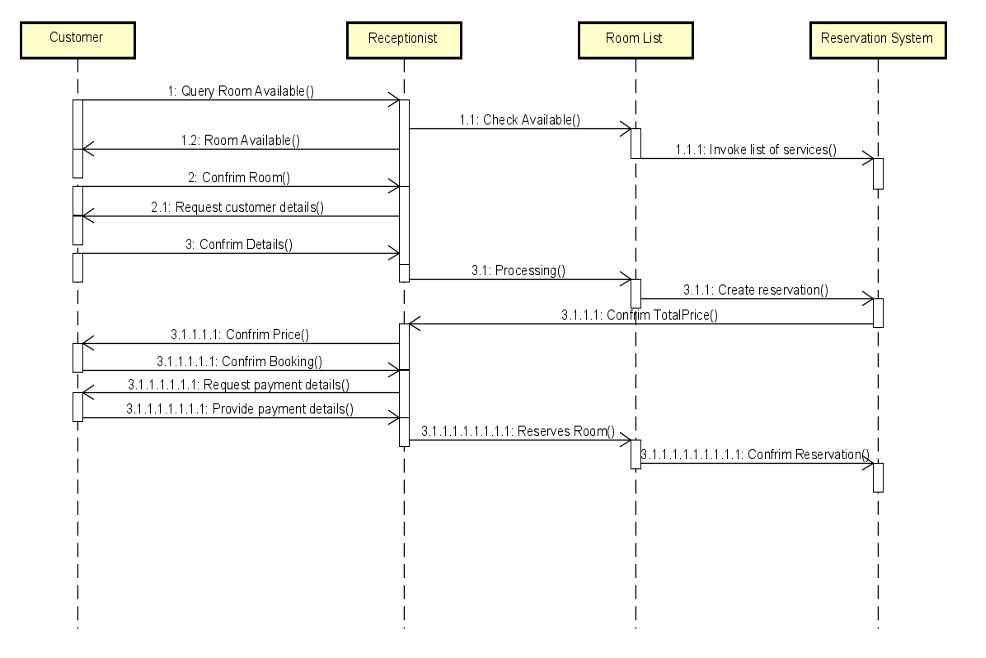


Figure 5.5. Booking.

# 6. Test Plan

## 6.1. Requirements/specifications-based system level test cases

**Deciding on the requirements and design of the system:**

In the early phases of the development process, the team decided it was essential to have a graphic user interface (GUI) for the software. After the sprint involving creating a GUI for the system, it was tested to check if all the tabs were working using a cout test in C#. Display tabs were set to non-editable tabs just to display the information.

**Checking for the functionality of XML to get room data and check if the form filters it:**

After inputting constant data for the rooms like cost and availability were called in the system using XML, they were tested in the GUI to display rooms by their type displaying all the room numbers for that type and their availability status. By selecting the type of room on the combo box in the GUI and clicking the filter, the system displayed all the available rooms of that type.

**Creating a new customer record and storing it using XML:**

After adding parts in the GUI for adding a new customer record by inputting customer details and generating an ID for the customer. The customer record was then stored in the system data using XML and checked if a new current customer was created in the customers sections of the GUI for that customer which is then used to pick the customer and make the booking for the previously selected room.

**Storing rooms and then using insertion sort to sort them by price:**

The rooms can be stored in an array because their room number and details never change because they are already built in the hotel, which makes it easier to sort the rooms by their price from low to high or vice-versa.

**Calculating final bill and customer check-out:**

The final bill is calculated by multiplying the room’s price per day of stay by the amount of days the guest stayed. After the member of staff receives the payment from the customer, they click on the check-out button which should set the room status back to available and set the guest Id and number of guests back to zero.

# 

# 7. Demo

# 7.1. Database

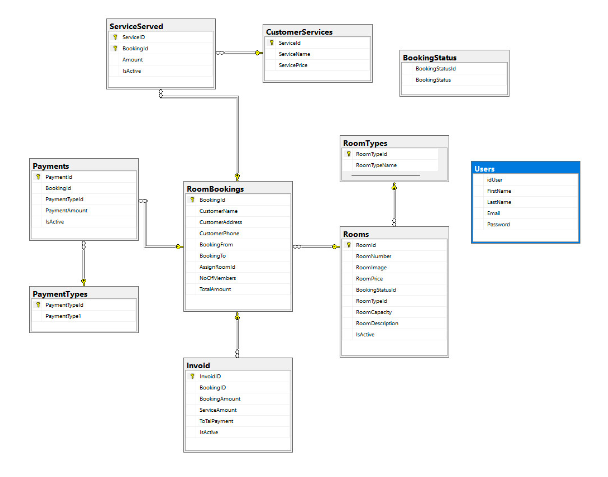
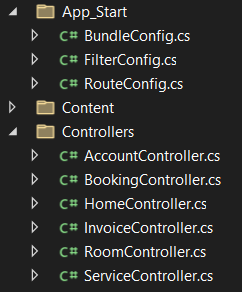
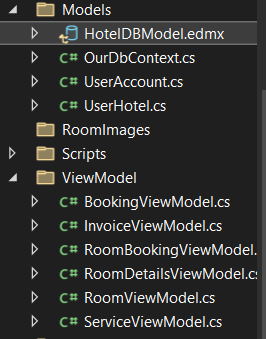


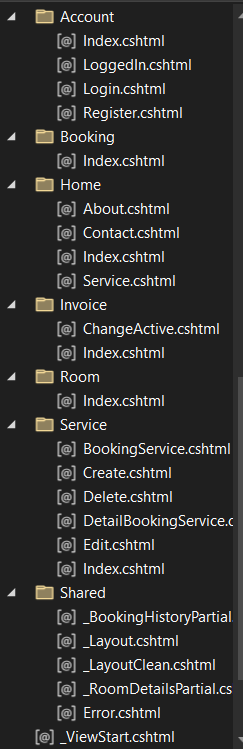
Figure 7.1. Database

## 7.2. Source code

structure of files.



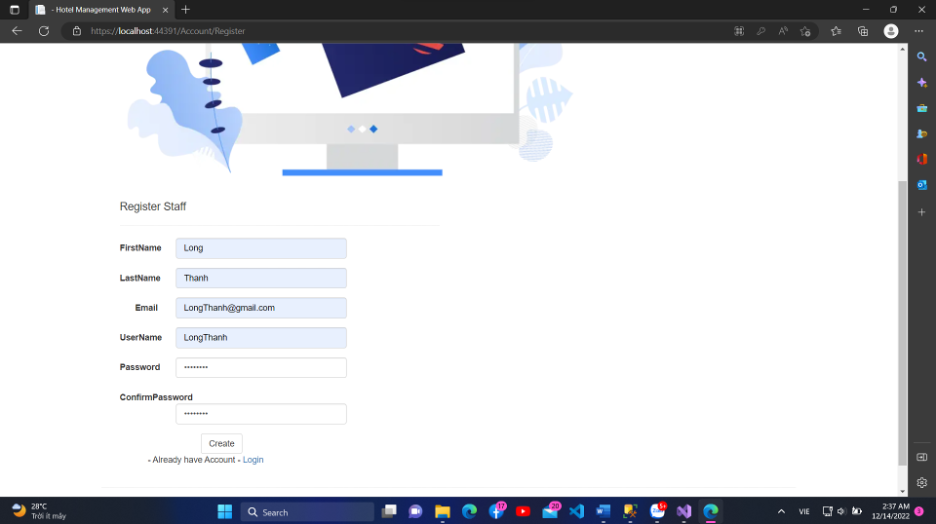




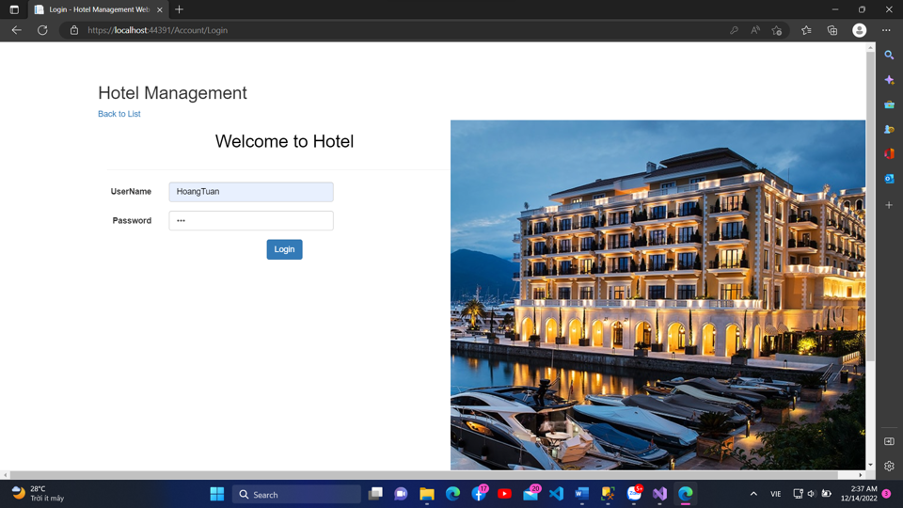
link source code:

7.4 Function

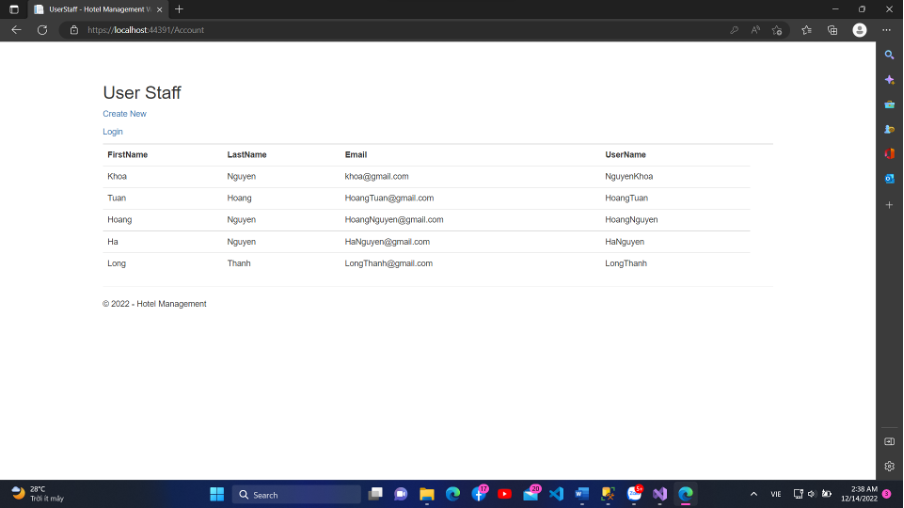
Register.



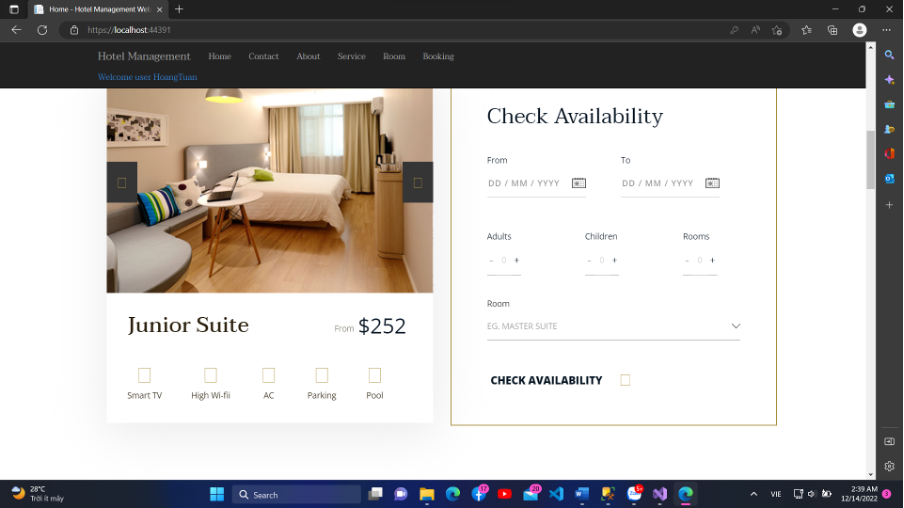
Login.

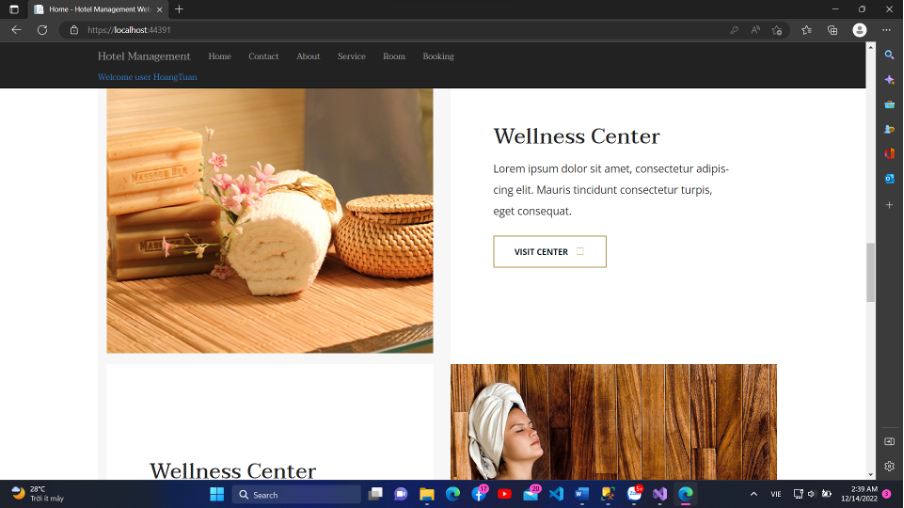


List of User.

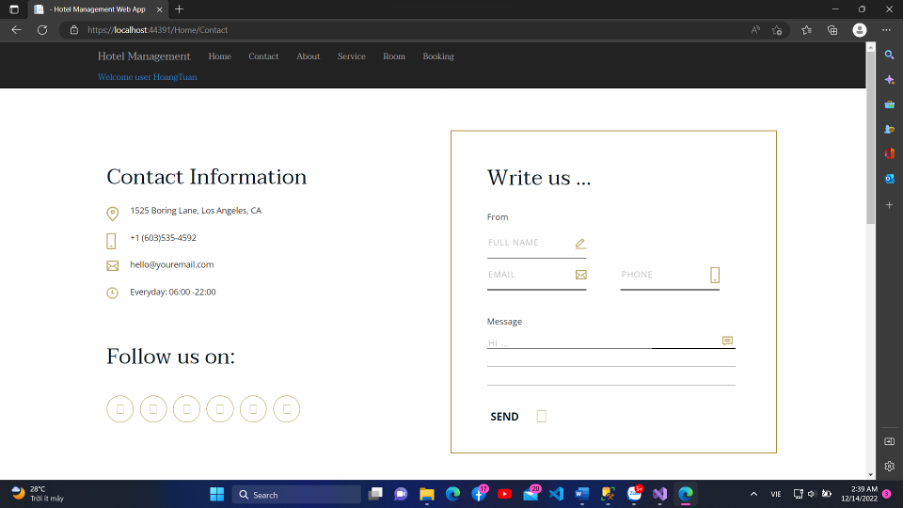


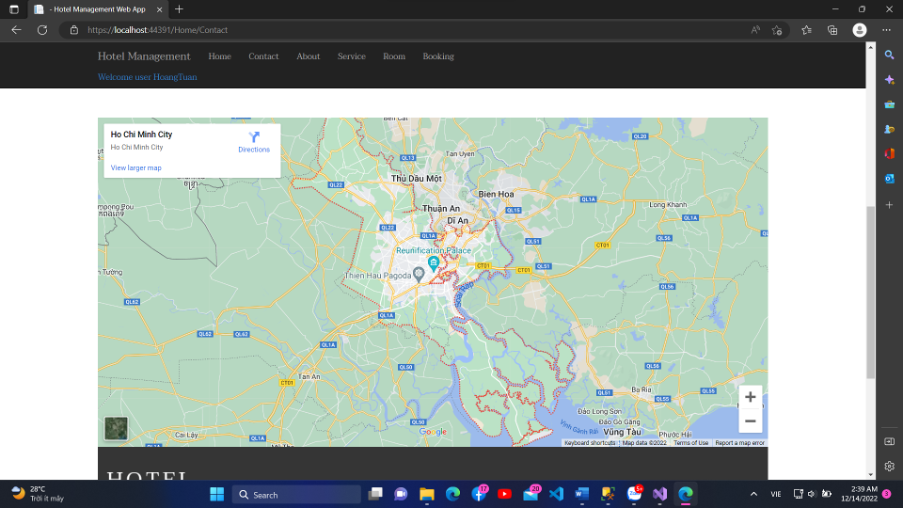
Home.



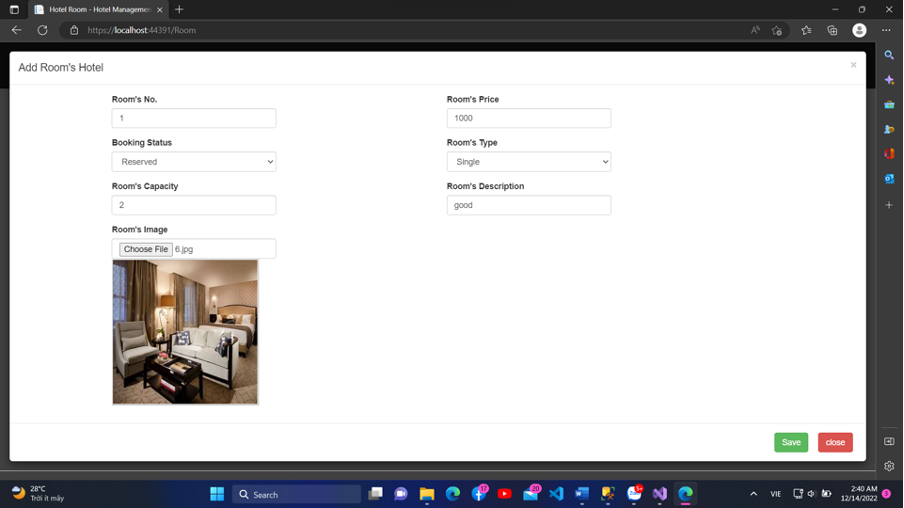


Contact with map api.

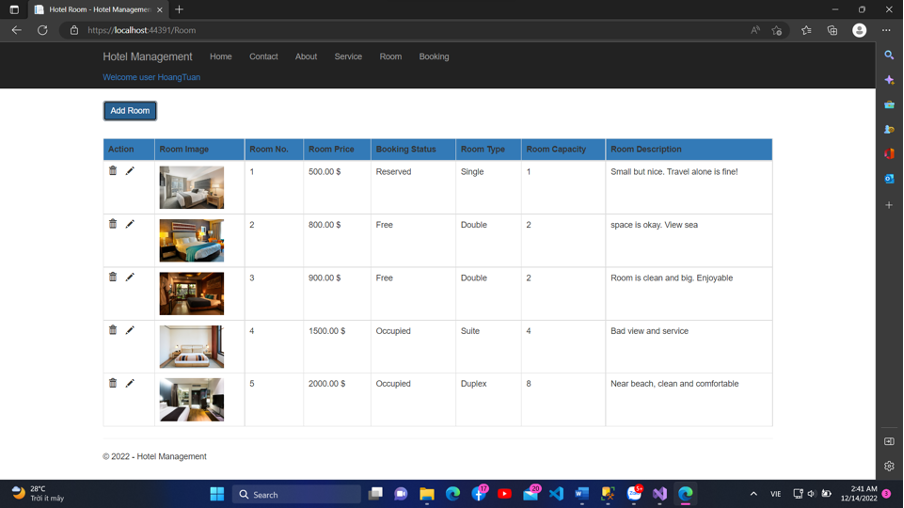




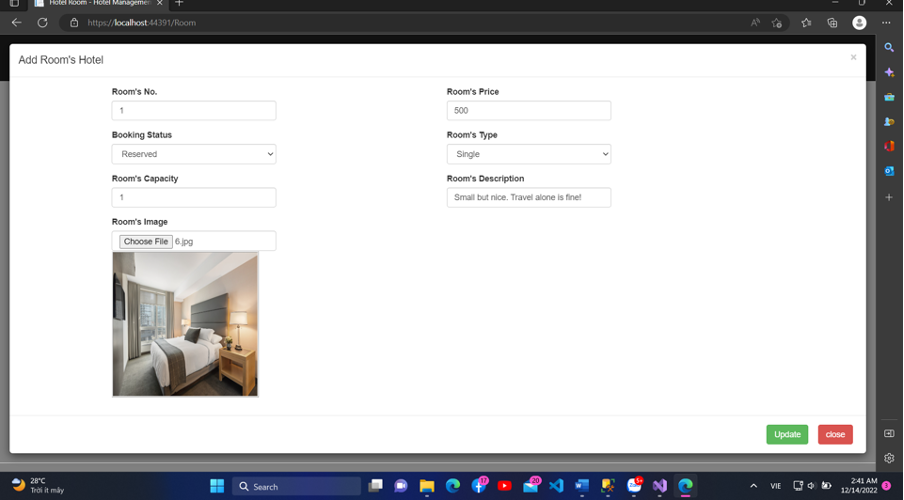
Add Room.



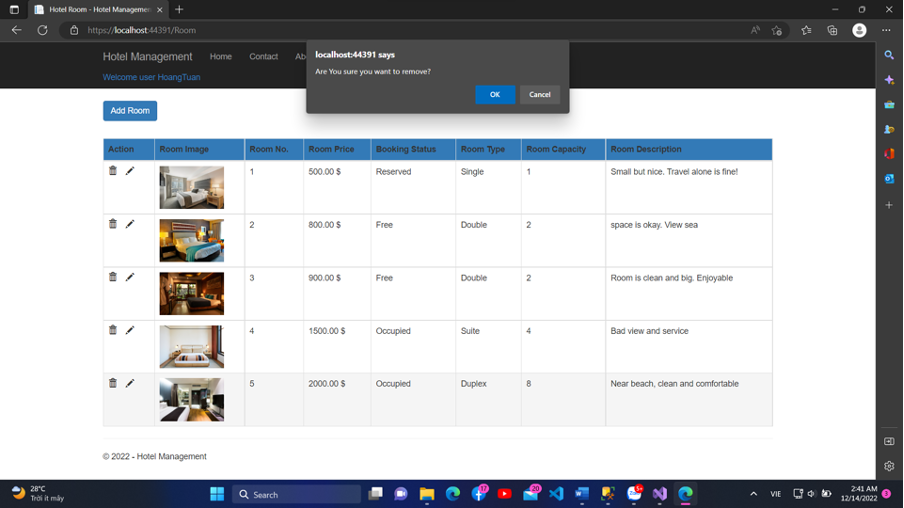
Show list of room.

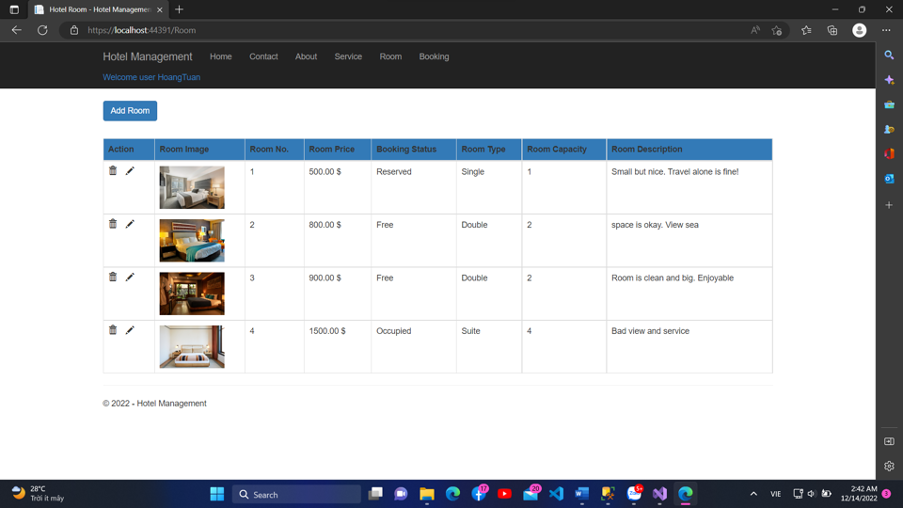


Update Room.

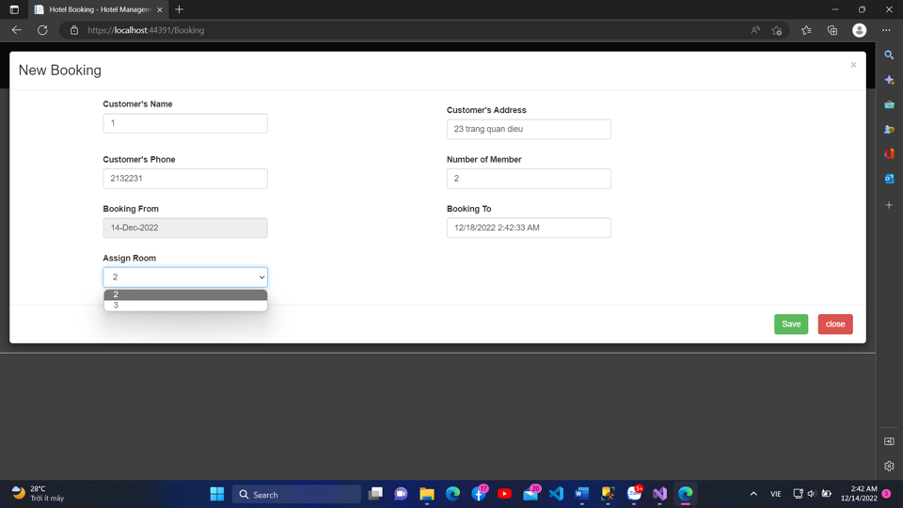


Remove Room.





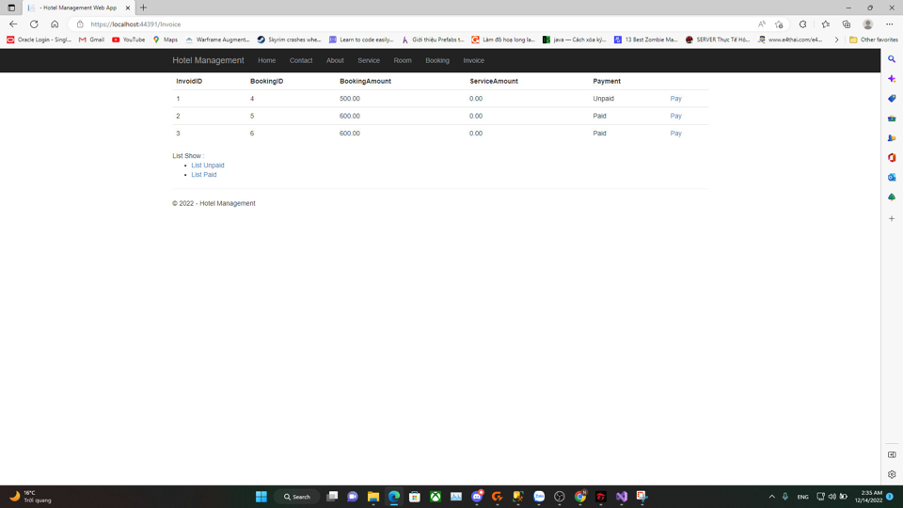
Booking avalable room.

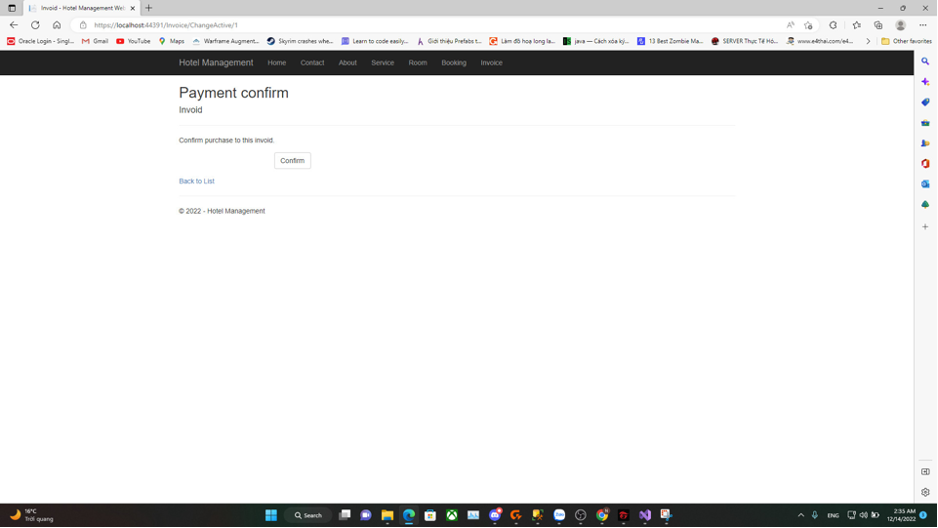


Show list booking with totalAmount to pay.



Payment.





## 7.3. Testing

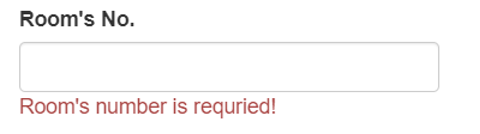
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Function | Describe | Display | Status |
| 1 | Login  (don’t fill) | 1. Go to login page.  2. Do not fill in Username or Password.  3. Press login button | “Username is required.”  “Password is required.” | Fail |
| 2 | Login  (Password or Username Wrong) | 1. Go to login page.  2. fill in Username or Password wrongly.  3. Press login button | “Username or Password is wrong.” | Fail |
| 4 | Login  (Password or Username Correct) | 1. Go to login page.  2. fill in Username or Password correctly.  3. Press login button | Home page displayed | Success |
| 5 | Register Staff  (don’t fill) | 1. login to admin page.  2. press account page and Create new button  3. Press create button | “First Name is required.”  “Last Name is required.”  … | Fail |
| 6 | Register Staff  (Email invalid) | In register page, Enter missing “@gmail.com” | “The field Email must match” | Fail |
| 7 | Register Staff  (Password do not match) | In register page, password component and confirm Password does not match | “Please confirm your password” | Fail |
| 8 | Register Staff  (All information are valid) | Correctly Fill in all components (Firstname, Lastname, Email, username, password) | “Username successfully registered.” | Success |
| 9 | Add room’s hotel  (information needed) | 1. Login  2. Go Room -> Add Room  3. Save | “Room's Price should be equal or greater than 500$.”  “Room's Capacity should be equal or greater than 1 people.”  … | Fail |
| 10 | Add room’s hotel  (information valid) | 1. Login  2. Go Room -> Add Room  3. Add information  4. Save | “Room successfully Added”  Room information show in form | Success |
| 11 | Booking  (don’t fill) | 1. Login  2. Go Booking -> New Booking  3. Save | “Customer's Name is required.”  “Customer's Address is required.” | Fail |
| 12 | Booking  (Correct information and have room free) | 1. Login  2. Go Booking -> New Booking  3. Add customer information and room free to asign  4. Save | Show information in form | Success |

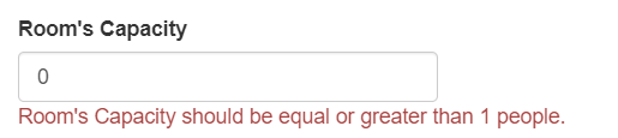
Test case for AddRoom().

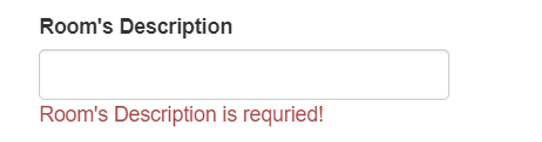
All fields must not be null.

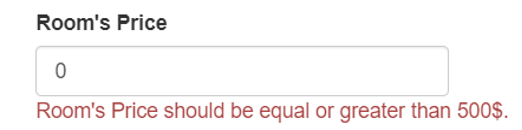
Room’s Capacity must be greater than 1 less than 10 person.

Room’s Price must be equal or greater than 500$



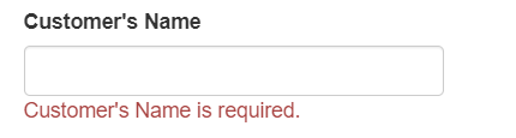


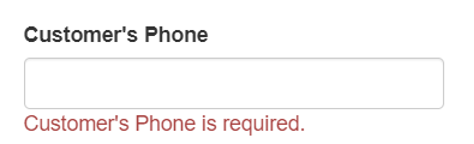


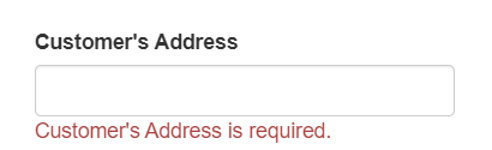


Test case for AddBooking().

All fields must not be null.

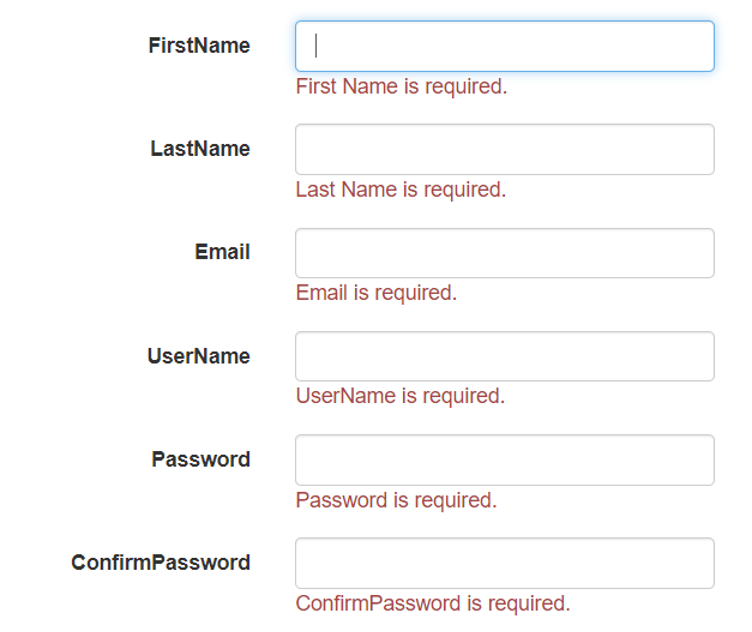






Test case for Register().

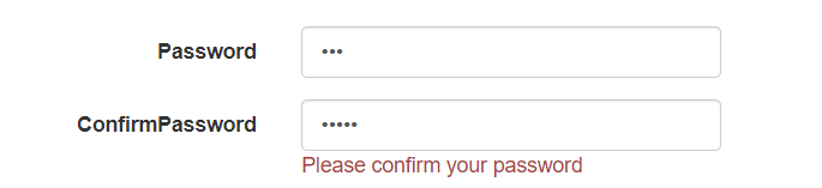
All fields must not be null.



Email must match the regular expression.



Password and confirmPassword have to be matched.



# 

# References

1. <https://viblo.asia/p/tim-hieu-ve-cach-thiet-ke-class-diagram-L4x5xLyY5BM>
2. <https://www.protel.net/hotel-management-system/>
3. <https://nevonprojects.com/hotel-management-system/>
4. geeksforgeeks.org/hotel-management-project-in-python/
5. https://dl.ucsc.cmb.ac.lk/jspui/bitstream/123456789/4030/1/1109138.pdf