VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY FACULTY OF**

**INFORMATION TECHNOLOGY**



**SOFTWARE ENGINEERING**

**FINAL REPORT**

*Instructing Lecturer*: **Mr. DANG MINH THANG**

*Student’s name*: **NGUYEN CONG THANH – 517H0084**

**DUONG DIA BAO – 517H0036**

Class **: 17050211**

Course **: 21**

**HO CHI MINH CITY, YEAR 2019.**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY FACULTY OF**

**INFORMATION TECHNOLOGY**



**SOFTWARE ENGINEERING**

**FINAL REPORT**

*Instructing Lecturer*: **Mr. DANG MINH THANG**

*Student’s name*: **NGUYEN CONG THANH – 517H0084**

**DUONG GIA BAO – 517H0036**

Class **: 17050211**

Course **: 21**

**HO CHI MINH CITY, YEAR 2019.**

**ACKNOWLEDGEMENT**

In order to make this report complete and achieve good results, we have received the support and assistance of many teachers and classmates.

With deep affection, sincerity, we express deep gratitude to all individuals and agencies who have helped us in our study and research.

First of all, I would like to express a special appreciation to Ton Duc Thang University’s teachers for their conscientious guidance and advices throughout the last semester by gave me their modern outlook and meticulous supervision to carry out the job perfectly.

Especially we would like to send our sincere thanks to Mr. Dang Minh Thang has paid attention, help, guide us to complete the report well over the past time.

We would like to express our sincere gratitude to the leadership of Ton Duc Thang University for supporting, helping and facilitating us to complete the report well during the study period.

With limited time and experience, this report can’t avoid mistakes. We are looking forward to receiving advice and comments from teachers so that we can improve our awareness, better serve the practical work later.

We sincerely thank you!

**THE PROJECT WAS COMPLETED AT TON DUC THANG UNIVERSITY**

I pledge that this is a product of our own project and is under the guidance of Mr. Dang Minh Thang. The content of research, results in this subject is honest and not published in any form before. The data in the tables used for the analysis, comment, and evaluation were collected by the authors themselves from various sources indicated in the reference section.

Besides, in my project, I also use many comments, evaluations as well as figures of other authors, organizations. Their documental sources are all quoted.

*Ho Chi Minh City, December 8th 2019*

*Authors*

*(Sign and write full name)*

*Nguyen Cong Thanh*

*Duong Gia Bao*

**EVALUATION AND INSTRUCTION LECTURER**

**Confirmation of the instructor**

Ho Chi Minh City, December 8th 2019

(Sign and write full name)

**The assessment of the teacher marked**

Ho Chi Minh City,December 8th 2019

(Sign and write full name)

**TABLE OF CONTENTS**

[**ACKNOWLEDGEMENT** i](#_Toc26874963)

[**THE PROJECT WAS COMPLETED AT TON DUC THANG UNIVERSITY** ii](#_Toc26874964)

[**EVALUATION AND INSTRUCTION LECTURER** iii](#_Toc26874965)

[**TABLE OF CONTENTS** 1](#_Toc26874966)

[**LIST OF FIGURES AND MODELS** 3](#_Toc26874967)

[**CHAPTER 1 – REQUIREMENT ENGINEERING** 4](#_Toc26874968)

[**1.1 Q\_A for project.** 4](#_Toc26874969)

[**1.2. Functional Requirements** 14](#_Toc26874970)

[**1.3. Non-Functional Requirement** 15](#_Toc26874971)

[**CHAPTER 2 – USE CASES** 16](#_Toc26874972)

[**2.1 Use case diagram** 16](#_Toc26874973)

[**2.2 Use case description** 17](#_Toc26874974)

[***2.2.1 Login*** 17](#_Toc26874975)

[***2.2.2 Logout*** 17](#_Toc26874976)

[***2.2.3 Manage customer*** 18](#_Toc26874977)

[***2.2.4 Manage reservation*** 19](#_Toc26874978)

[***2.2.5 Manage room*** 19](#_Toc26874979)

[***2.2.6 Manage service*** 20](#_Toc26874980)

[***2.2.7 Manage payment*** 21](#_Toc26874981)

[***2.2.8 Book room*** 21](#_Toc26874982)

[***2.2.9 View customer list*** 22](#_Toc26874983)

[***2.2.10 View reservation list*** 22](#_Toc26874984)

[***2.2.11 Add customer*** 24](#_Toc26874985)

[***2.2.12 Update customer*** 24](#_Toc26874986)

[***2.2.13 Delete customer*** 25](#_Toc26874987)

[***2.2.14 Update reservation*** 26](#_Toc26874988)

[***2.2.15 Delete reservation*** 27](#_Toc26874989)

[***2.2.16 Add service*** 27](#_Toc26874990)

[***2.2.17 Update service*** 29](#_Toc26874991)

[***2.2.18 Delete service*** 29](#_Toc26874992)

[***2.2.19 Confirm reservation*** 31](#_Toc26874993)

[***2.2.20 Check available*** 31](#_Toc26874994)

[***2.2.21 Add payment*** 33](#_Toc26874995)

[***2.2.22 Add payment*** 33](#_Toc26874996)

[**CHAPTER 3 – DIAGRAMS AND MODELS** 34](#_Toc26874997)

[**3.1 Class Diagrams** 34](#_Toc26874998)

[**3.2 Activity Diagrams** 34](#_Toc26874999)

[**3.3 Sequence Diagrams** 39](#_Toc26875000)

[**3.4 Entity Relational Diagrams** 40](#_Toc26875001)

[**3.5 Relational Data Model** 41](#_Toc26875002)

[**CHAPTER 4 – PROJECT APPLICATION** 42](#_Toc26875003)

[**4.1 Programming language** 42](#_Toc26875004)

[***4.1.1 HTML definition*** 42](#_Toc26875005)

[***4.1.2 CSS definition*** 42](#_Toc26875006)

[***4.1.3 Bootstrap definition*** 42](#_Toc26875007)

[***4.1.4 PHP definition*** 43](#_Toc26875008)

[**4.2 Website** 43](#_Toc26875009)

[***REFERENCES*** 47](#_Toc26875010)

**LIST OF FIGURES AND MODELS**

[Figure 2.1.1 Use case diagram 4](#UsecaseDiagrams)

[Figure 3.1.1 Class Diagrams 16](#ClassDiagrams)

[Figure 3.2.1 a Activity diagram of employee management 29](#_Toc26730189)

[Figure 3.2.1 b Activity diagram of customer management 30](#_Toc26730190)

[Figure 3.2.1 c Activity diagram of Room management 31](#_Toc26730191)

[Figure 3.2.1 d Activity diagram of Reservation management 32](#_Toc26730192)

[Figure 3.2.1 e Activity diagram of the reservation 33](#_Toc26730193)

[Figure 3.3.1 Sequence Diagrams 18](#SequenceDiagrams)

[Figure 3.4.1 Entity Relational Diagrams 19](#EntityRelationalDiagrams)

[Figure 3.5.1 Relational Data Model 19](#RelationalDataDiagrams)

**CHAPTER 1 – REQUIREMENT ENGINEERING**

**1.1 Q\_A for project.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Module**  **Screen** | **Description** | **Creator** | **Create Date** | **Answer** | **Answer Date** | **Status** |
| 1. | Login | About login function. Description of the Login screen, when an incorrect username or password is entered, the error message is displayed. I propose the following 2 options:  1: The message will read "Invalid username or password" in red, at the bottom of the Login button.  2: Display a message box that says "Invalid username or password".  Hope you confirm which option you choose. | Dương Gia Bảo | 4/10/2019 |  |  | Open |
| 2. | Main Interface Screen | About the Home screen display. In the room list, we show the room list with basic important information to the customer or give specific details. As I understand, just leave basic information important, when users want to see more details about their room like that they can click to see specific details about that room. Did I get it right ? | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 3. | Book Room | Regarding the booking, in case the date of reservation for which the time frame is full or the time frame is not available, the customer will need to notify the guest. As I understand it, the message should be "the time frame you choose doesn't have this room type, please choose another time frame". This message will be displayed in the message box. I understand that true?  About booking function. Room description, when a customer successfully booked, a notification will be displayed to the user like. I propose the following 2 options:  1: The message box says "you have successfully booked and please wait for the confirmation email".  2: Display by linking to a page that says "you have successfully booked and please wait for the confirmation email" and enclose the thank you to the customer.  Hope you confirm which option you choose. | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 4. | Add Admin | To add employee information, when doing the following:  Step 1: Choose an admin profile.  Step 2: Select the add button at the top of the staff list, next to the word 'admin profile'.  Step 3: Fill in the information the employee needs to add to the form.  The program will automatically add that employee to the employee list.  After adding the employee, it will save the data into the database and display it on the employee list. I understand that true?  In case the add staff already exists, it is necessary to report an error or not. In my opinion, the error message should be, the content of the error message will be "The user already exists". This message will be displayed in the message box. I understand that true? | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 5. | Edit Admin | To correct employee information, do the following:  Step 1: Select the employee profile.  Step 2: Select the edit button in the edit section of each employee.  Step 3: Fill in the information the employee needs to correct on the form.  The program will automatically update the revised employee information in the data.  After editing the employee, it will save the data into the database and display the information on the employee list again. I understand that true? | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 6. | Delete Admin | To delete employee information, do the following:  Step 1: Select the employee profile.  Step 2: Select the delete button in the delete section of each employee.  The program will automatically delete that employee information and update the number of employees in the employee list.  After deleting the employee, it will save the data into the database and display the employee information and the number of employees on the employee list. I understand that true? | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 7. | Add Customer | To add customer information, when doing the following:  Step 1: Select a customer profile.  Step 2: Select the add button at the top of the customer list, next to the word 'customer profile'.  Step 3: Fill in the information the customer needs to add to the form.  The program will automatically add that customer to the customer list.  After adding the customer, it will save the data into the database and display it on the customer list. I understand that true? | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 8. | Delete Customer | To delete customer information, do the following:  Step 1: Select the customer profile.  Step 2: Select the delete button in the delete section of each customer.  The program will automatically delete that customer information and update the number of customers in the customer list.  After deleting the customer, it will save the data into the database and display the customer information and the number of customers on the customer list. I understand that true? | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 9. | Edit Reservation | To edit the reservation information, when doing the following:  Step 1: Choose a reservation.  Step 2: Select the edit button in the edit section of each reservation.  Step 3: switch from pending status to approval  The program will automatically add that customer information to the customer list.  It will then send a notification email to notify customers of the successful reservation request and provide information and room number. I understand that is true? | Dương Gia Bảo | 5/10/2019 |  |  | Open |
| 10. | Delete Reservation | To delete reservation information, when doing the following:  Step 1: Select a reservation.  Step 2: Select the delete button in the delete section of each reservation.  The program will automatically delete customer information and request that reservation in the reservation list.  After deleting the reservation, it will send an email notifying the customer of the unsuccessful request and reason. I understand that true? | Dương Gia Bảo | 5/10/2019 |  |  | Open |

**1.2. Functional Requirements**

* External interface: The web has a user-friendly, non-annoying interface, combining minimalism and pleasant colors that make users excited about the web experience, and easy to interact with web
* Authentication: Fast booking and authentication tool with minimal steps, simple and intuitive booking process, and must be able to send guests a message to be assured about the reservation and update the booking changes if The reservation tool allows you to automatically send promotional emails to guests before arrival, during your stay, after departure.
* Transaction corrections, adjustments and cancellations: Support multiple languages ​​and currencies. Integrate with various payment gateways. And payment automation (ability to send payment links and automatic credit card verification)
* Authorization levels: Enable users to search and find the most relevant booking options.

**1.3. Non-Functional Requirement**

* Confidentiality: absolutely protect customers 'personal information and prevent customers' information from being distributed.
* Maintenance: Easy to maintain and update, reasonably priced to maintain and update.
* Availability: There is 1 hour of downtime every 3 months and 1 hour of downtime to update the wed about promotions during the year to help customers get the best deals possible.
* Usability: Employees can easily use the system after 3 hours of training. Customers can easily use the web to search, view, book after 2 times through the simple steps of booking at wed.
* Capacity: can withstand up to 1200 visitors at the same time.
* Interoperability: Friendly interface, good looking and minimalist with customers, pleasant colors, easy to remember and easy to use.
* Reliability: Room and price information on the web must be true to the outside. Photos posted appropriate and similar to reality.
* Environment: The Web can run on different browsers like Firefox, Google Chrome, ... and can run on different devices such as laptops, ipads, phones, ...

**CHAPTER 2 – USE CASES**

**2.1 Use case diagram**

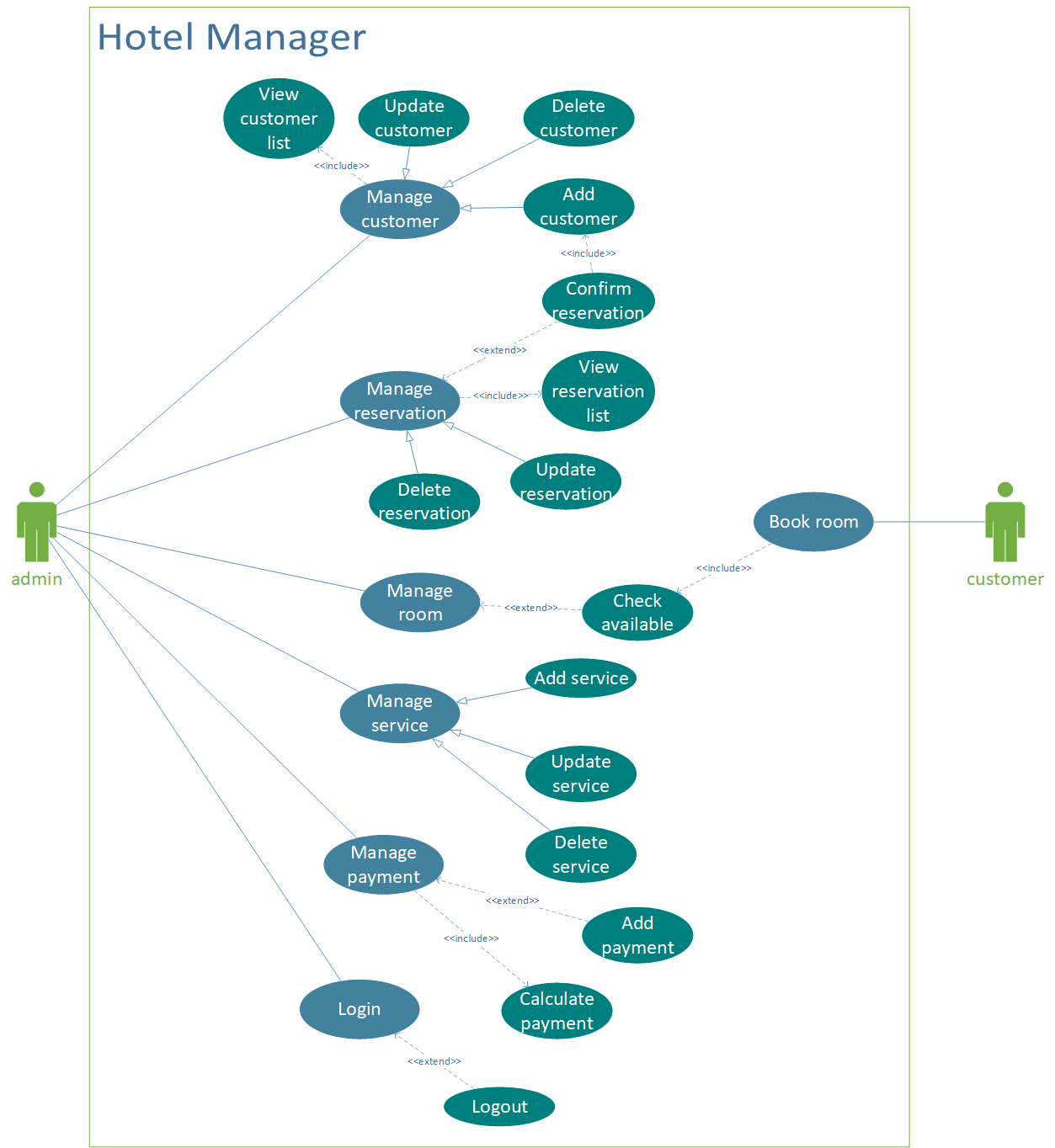


Figure 2.1.1 Use case diagram

**2.2 Use case description**

***2.2.1 Login***

|  |  |
| --- | --- |
| **ID:** | **1** |
| **Title:** | Login |
| **Description:** | This use case allows admin to login to admin panel |
| **Primary Actor:** | admin |
| **Preconditions:** | There are no preconditions for this use case. |
| **Postconditions:** | Admin can manage customer, manage reservation, manage room, manage service, manage payment |
| **Main  Success Scenario:** | This use case starts when admin want to access to admin panel  - Basic Flows:  Admin enter username and password to access. If username and password is correct, admin panel will display  - Alternative Flows:  If username and password is incorrect, system will display an error message in order to re-enter the username and password |
| **Extensions:** | There is no extension point for this use case |

***2.2.2 Logout***

|  |  |
| --- | --- |
| **ID:** | **2** |
| **Title:** | Logout |
| **Description:** | This use case allows admin to logout from admin panel |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case |
| **Postconditions:** | Admin can logout from admin panel |
| **Main  Success Scenario:** | This use case starts when admin want to leave from admin panel to the main page  - Basic Flows:  Admin click on the logout button, so that admin can logout from admin panel |
| **Extensions:** | There is no extension point for this use case |

***2.2.3 Manage customer***

|  |  |
| --- | --- |
| **ID:** | **3** |
| **Title:** | Manage customer |
| **Description:** | This use case allows admin to manage customer |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case |
| **Postconditions:** | Admin can manage customer from the admin panel: Add, Delete customer. |
| **Main  Success Scenario:** | This use case starts when admin want to add or delete customer profile.  - Basic Flows:  Admin access to customer profile, admin can add new customer or delete exist customer.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.4 Manage reservation***

|  |  |
| --- | --- |
| **ID:** | **4** |
| **Title:** | Manage reservation |
| **Description:** | This use case allows admin to manage reservation |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case |
| **Postconditions:** | Admin can manage reservation from the admin panel: Edit, Delete reservation. Admin can confirm reservation |
| **Main  Success Scenario:** | This use case starts when admin want to add or delete reservation profile.  - Basic Flows:  Admin access to reservation profile, admin can edit reservation or delete exist reservation  When system exists a reservation, admin have to confirm  System will edit or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.5 Manage room***

|  |  |
| --- | --- |
| **ID:** | **5** |
| **Title:** | Manage room |
| **Description:** | This use case allows admin to manage room |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case |
| **Postconditions:** | Admin can manage room from the admin panel: Admin can check the available of a room includes check in and check out |
| **Main  Success Scenario:** | This use case starts when admin want to update room  - Basic Flows:  Admin access to rooms profile, admin can update the check in and check out of a room  System will update to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.6 Manage service***

|  |  |
| --- | --- |
| **ID:** | **6** |
| **Title:** | Manage service |
| **Description:** | This use case allows admin to manage service |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case |
| **Postconditions:** | Admin can manage service from the admin panel: Add, Delete, Update service. |
| **Main  Success Scenario:** | This use case starts when admin want to add or delete service profile.  - Basic Flows:  Admin access to service profile, admin can add new service, update or delete exist service.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.7 Manage payment***

|  |  |
| --- | --- |
| **ID:** | **7** |
| **Title:** | Manage payment |
| **Description:** | This use case allows admin to manage payment |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case |
| **Postconditions:** | Admin can manage payment from the admin panel: Add payment or calculate the payment |
| **Main  Success Scenario:** | This use case starts when admin want to add or calculate payment profile.  - Basic Flows:  Admin access to payment profile, admin can add new payment or calculate all the payment  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.8 Book room***

|  |  |
| --- | --- |
| **ID:** | **8** |
| **Title:** | Book room |
| **Description:** | This use case allows customer to book room |
| **Primary Actor:** | customer |
| **Preconditions:** | There are no preconditions for this use case. |
| **Postconditions:** | Customer can book a room, customer has to fill in the information during reservation process |
| **Main  Success Scenario:** | This use case starts when customer want to book room  - Basic Flows:  Customer fill his(or her) information when his(or her) make a reservation  System will add new customer and new reservation to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.9 View customer list***

|  |  |
| --- | --- |
| **ID:** | **9** |
| **Title:** | View customer list |
| **Description:** | This use case allows admin to view customer list |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to customer profile |
| **Postconditions:** | Admin can view customer list |
| **Main  Success Scenario:** | This use case starts when admin want to view customer list  - Basic Flows:  Admin can see the customer list from customer profile  System will show the customer list from database |
| **Extensions:** | There is no extension point for this use case |

***2.2.10 View reservation list***

|  |  |
| --- | --- |
| **ID:** | **10** |
| **Title:** | View reservation list |
| **Description:** | This use case allows admin to view reservation list |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to reservation profile |
| **Postconditions:** | Admin can view reservation list |
| **Main  Success Scenario:** | This use case starts when admin want to view reservation list  - Basic Flows:  Admin can see the reservation list from reservation profile  System will show the reservation list from database |
| **Extensions:** | There is no extension point for this use case |

***2.2.11 Add customer***

|  |  |
| --- | --- |
| **ID:** | **11** |
| **Title:** | Add customer |
| **Description:** | This use case allows admin to add new customer |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to customer profile |
| **Postconditions:** | New customer has to be added |
| **Main  Success Scenario:** | This use case starts when admin want to add new customer from customer profile.  - Basic Flows:  Admin access to customer profile, admin can add new customer.  When customer book a room, system will add new customer to database  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.12 Update customer***

|  |  |
| --- | --- |
| **ID:** | **12** |
| **Title:** | Update customer |
| **Description:** | This use case allows admin to update customer |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to customer profile |
| **Postconditions:** | Customer has to be updated |
| **Main  Success Scenario:** | This use case starts when admin want to update new customer from customer profile.  - Basic Flows:  Admin access to customer profile, admin can update customer.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.13 Delete customer***

|  |  |
| --- | --- |
| **ID:** | **13** |
| **Title:** | Delete customer |
| **Description:** | This use case allows admin to delete customer |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to customer profile |
| **Postconditions:** | Customer has to be deleted |
| **Main  Success Scenario:** | This use case starts when admin want to update new customer from customer profile.  - Basic Flows:  Admin access to customer profile, admin can delete customer.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.14 Update reservation***

|  |  |
| --- | --- |
| **ID:** | **14** |
| **Title:** | Update reservation |
| **Description:** | This use case allows admin to update reservation |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to reservation profile |
| **Postconditions:** | Reservation has to be updated |
| **Main  Success Scenario:** | This use case starts when admin want to update new reservation from reservation profile.  - Basic Flows:  Admin access to reservation profile, admin can update reservation.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.15 Delete reservation***

|  |  |
| --- | --- |
| **ID:** | **15** |
| **Title:** | Delete reservation |
| **Description:** | This use case allows admin to delete reservation |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to reservation profile |
| **Postconditions:** | Reservation has to be deleted |
| **Main  Success Scenario:** | This use case starts when admin want to update new reservation from reservation profile.  - Basic Flows:  Admin access to reservation profile, admin can delete reservation.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.16 Add service***

|  |  |
| --- | --- |
| **ID:** | **16** |
| **Title:** | Add service |
| **Description:** | This use case allows admin to add new service |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to service profile |
| **Postconditions:** | New service has to be added |
| **Main  Success Scenario:** | This use case starts when admin want to add new service from service profile.  - Basic Flows:  Admin access to service profile, admin can add new service.  When service book a room, system will add new service to database  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.17 Update service***

|  |  |
| --- | --- |
| **ID:** | **17** |
| **Title:** | Update service |
| **Description:** | This use case allows admin to update service |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to service profile |
| **Postconditions:** | Service has to be updated |
| **Main  Success Scenario:** | This use case starts when admin want to update new service from service profile.  - Basic Flows:  Admin access to service profile, admin can update service.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.18 Delete service***

|  |  |
| --- | --- |
| **ID:** | **18** |
| **Title:** | Delete service |
| **Description:** | This use case allows admin to delete service |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to service profile |
| **Postconditions:** | Service has to be deleted |
| **Main  Success Scenario:** | This use case starts when admin want to update new service from service profile.  - Basic Flows:  Admin access to service profile, admin can delete service.  System will add or remove to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.19 Confirm reservation***

|  |  |
| --- | --- |
| **ID:** | **19** |
| **Title:** | Confirm reservation |
| **Description:** | This use case allows admin to confirm reservation |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to reservation profile |
| **Postconditions:** | Reservation has to be confirmed after customer book a room |
| **Main  Success Scenario:** | This use case starts when admin want to confirm reservation from reservation profile.  - Basic Flows:  Admin access to reservation profile, admin can confirm reservation then add new customer to database  System will add to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.20 Check available***

|  |  |
| --- | --- |
| **ID:** | **20** |
| **Title:** | Check available |
| **Description:** | This use case allows admin to check the available |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to reservation profile |
| **Postconditions:** | Reservation has to be confirmed after customer book a room |
| **Main  Success Scenario:** | This use case starts when admin want to check the available from reservation profile.  - Basic Flows:  Admin access to reservation profile, admin can check the available and confirm reservation  System will add to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.21 Add payment***

|  |  |
| --- | --- |
| **ID:** | **21** |
| **Title:** | Add payment |
| **Description:** | This use case allows admin to add payment |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to payment profile |
| **Postconditions:** | Payment has to be confirmed after customer book a room |
| **Main  Success Scenario:** | This use case starts when admin want to add payment from payment profile.  - Basic Flows:  Admin access to payment profile, admin can add payment to the system  System will add to database |
| **Extensions:** | There is no extension point for this use case |

***2.2.22 Add payment***

|  |  |
| --- | --- |
| **ID:** | **22** |
| **Title:** | Calculate payment |
| **Description:** | This use case allows admin to calculate payment |
| **Primary Actor:** | admin |
| **Preconditions:** | Admin must be login before starting this use case and access to payment profile |
| **Postconditions:** | Payment has to be confirmed and calculate after customer book a room |
| **Main  Success Scenario:** | This use case starts when admin want to add payment from payment profile.  - Basic Flows:  Admin access to payment profile, admin can add calculate to the system  System will add to database |
| **Extensions:** | There is no extension point for this use case |

**CHAPTER 3 – DIAGRAMS AND MODELS**

**3.1 Class Diagrams**

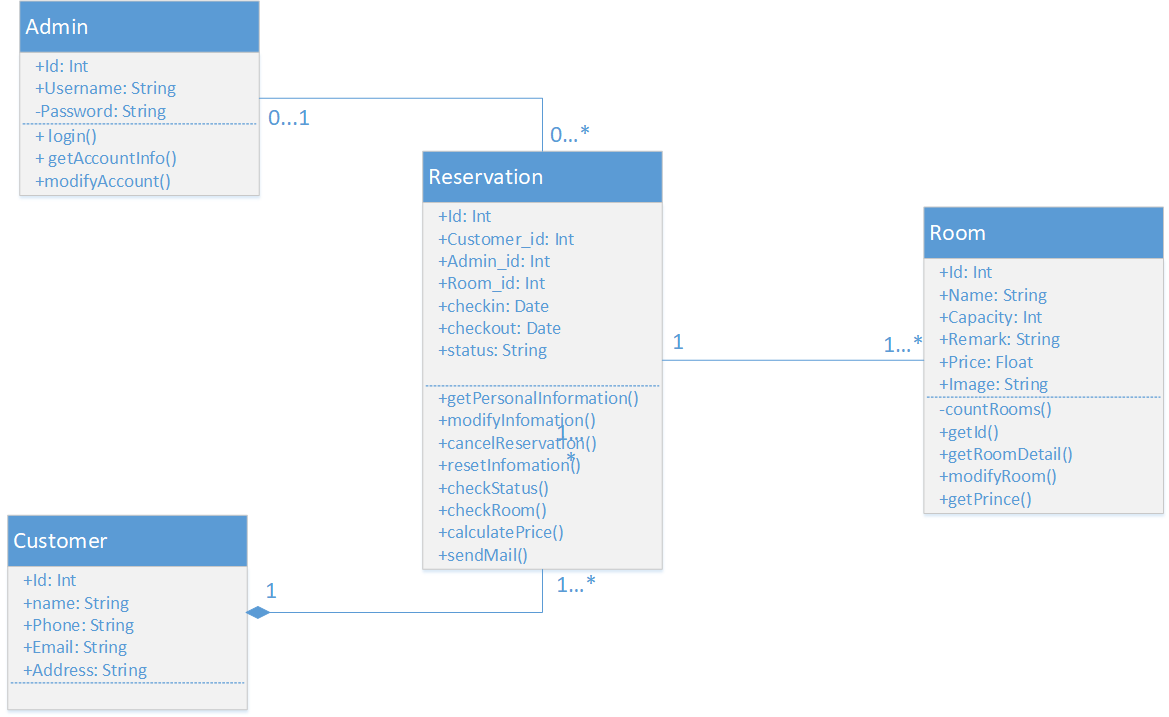


Figure 3.1.1 Class Diagrams

**3.2 Activity Diagrams**

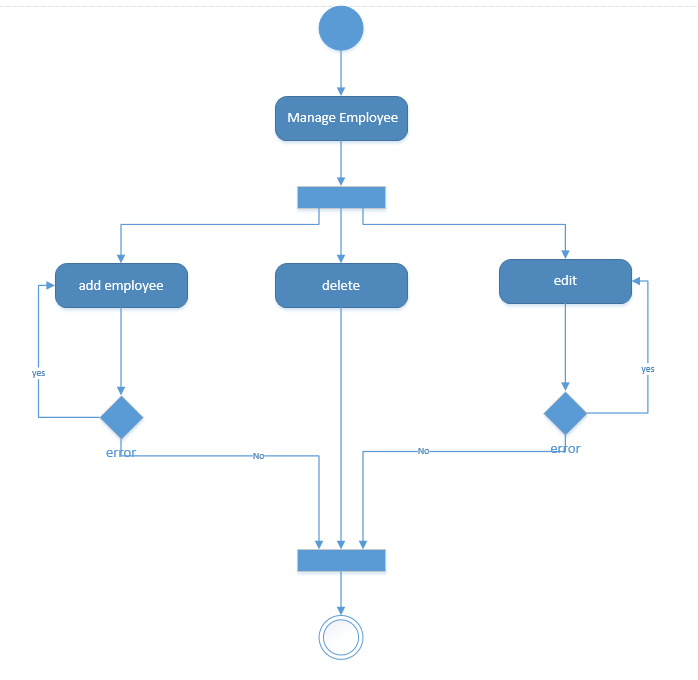


Figure 3.2.1 a Activity diagram of employee management

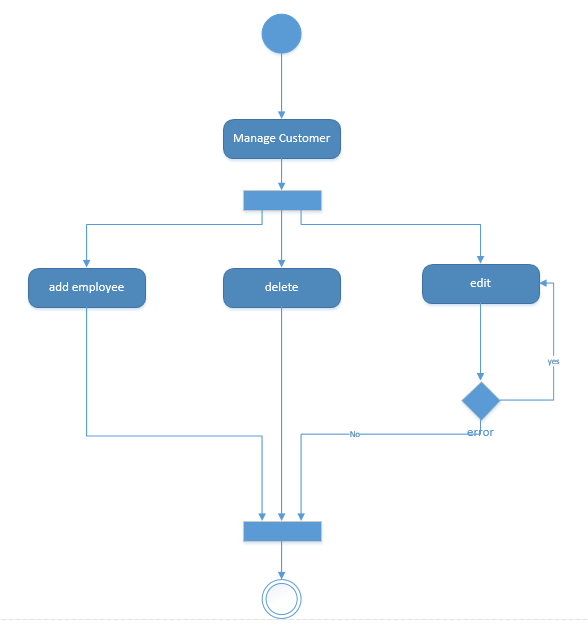


Figure 3.2.1 b Activity diagram of customer management

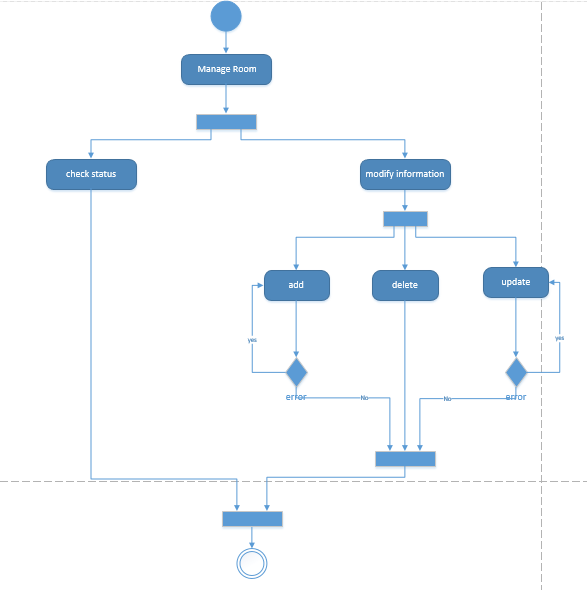


Figure 3.2.1 c Activity diagram of Room management

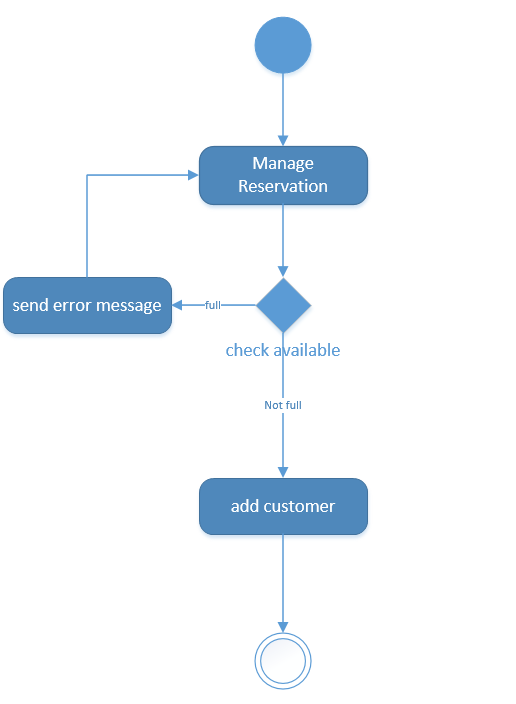


Figure 3.2.1 d Activity diagram of Reservation management

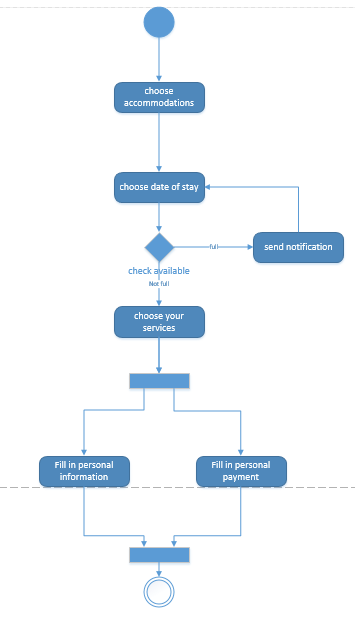


Figure 3.2.1 e Activity diagram of the reservation

**3.3 Sequence Diagrams**

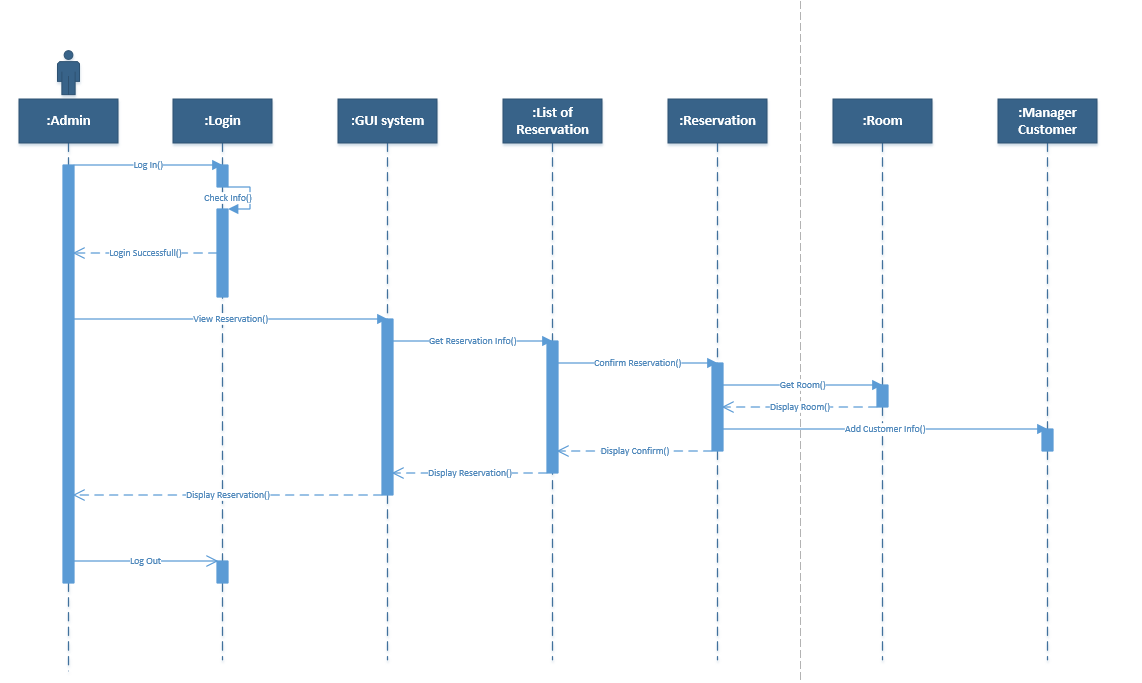


Figure 3.3.1 Sequence Diagrams

**3.4 Entity Relational Diagrams**

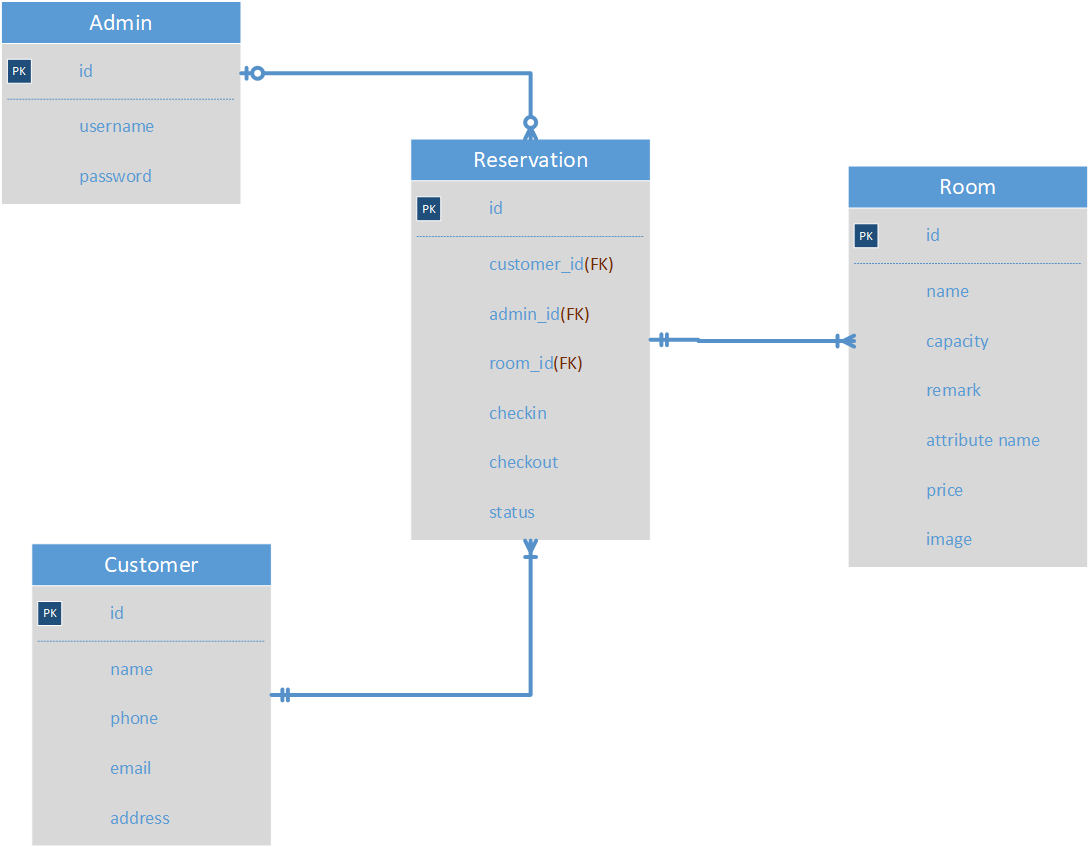


Figure 3.4.1 Entity Relational Diagrams

**3.5 Relational Data Model**

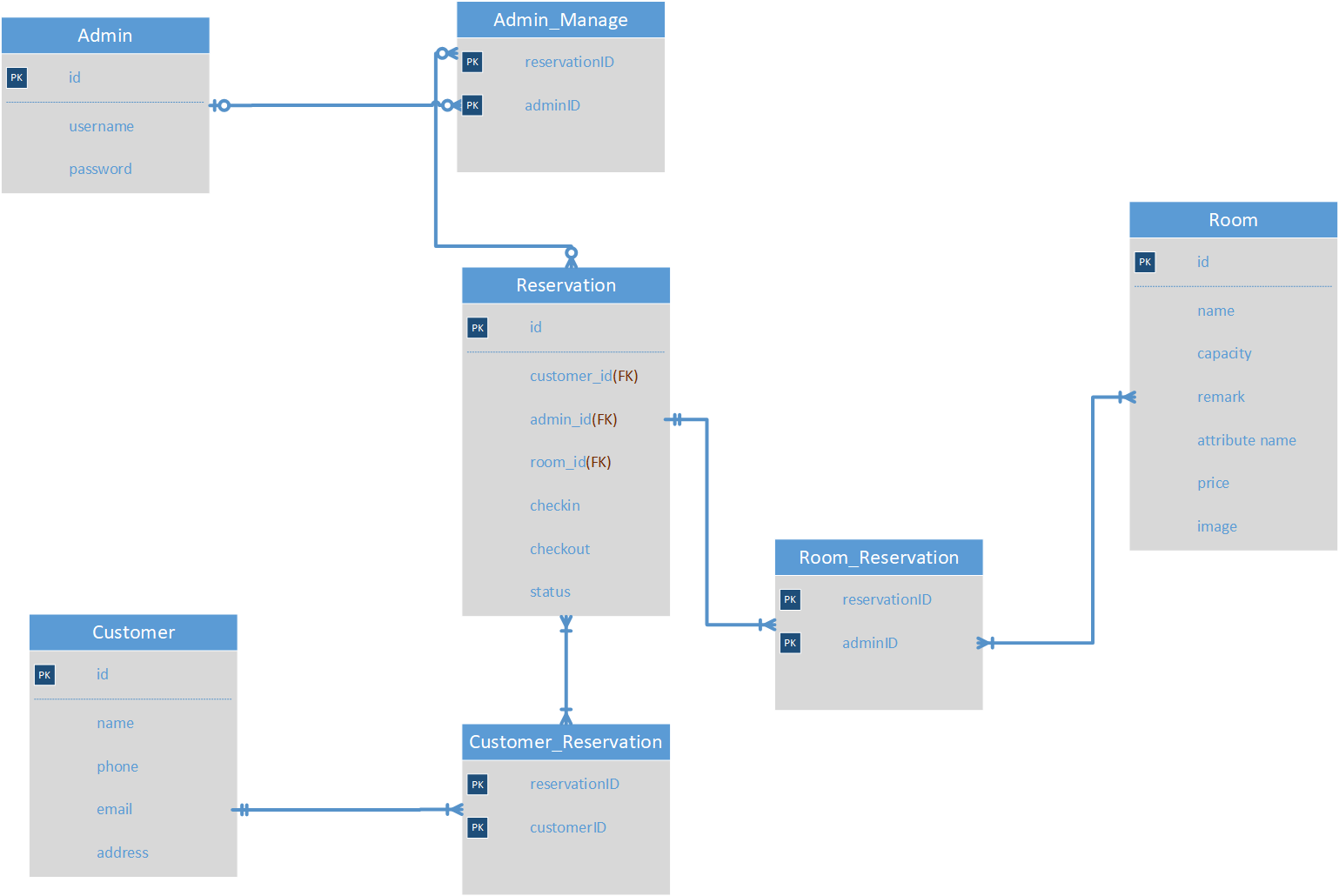


Figure 3.5.1 Relational Data Model

**CHAPTER 4 – PROJECT APPLICATION**

**4.1 Programming language**

In this project, we will use HTML, CSS, Bootstrap and PHP to create a website.

***4.1.1 HTML definition***

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

***4.1.2 CSS definition***

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. [1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate.css file, and reduce complexity and repetition in the structural content.

***4.1.3 Bootstrap definition***

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

Bootstrap is the sixth-most-starred project on GitHub, with more than 135,000 stars, behind freeCodeCamp (almost 307,000 stars) and marginally behind Vue.js framework. According to Alexa Rank, Bootstrap getbootstrap.com is in the top-2000 in US while vuejs.org is in top-7000 in US.

***4.1.4 PHP definition***

PHP is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

In PHP, we will connect many functions to database.

For customer, when they connect to the website, they can book a room and fill in their information. Then their information will save into the database.

For admin, they can add, delete, update information in admin panel.

**4.2 Website**

In PHP, we will connect many functions to database.

For customer, when they connect to the website, they can book a room and fill in their information. Then their information will save into the database.

For admin, they can add, delete, update information in admin panel.

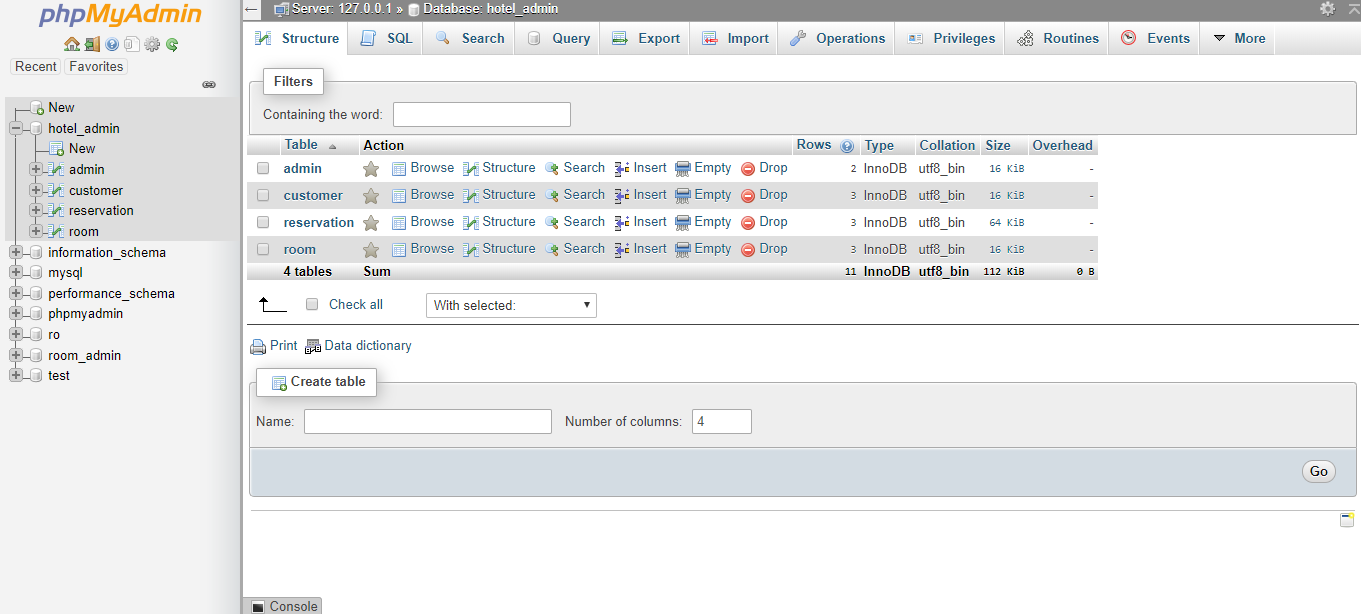


Figure 4.2.1 Database

In main page of customer, they can book a room. When customer book a room, they have to fill in the their information. Their information will be saved into the database.

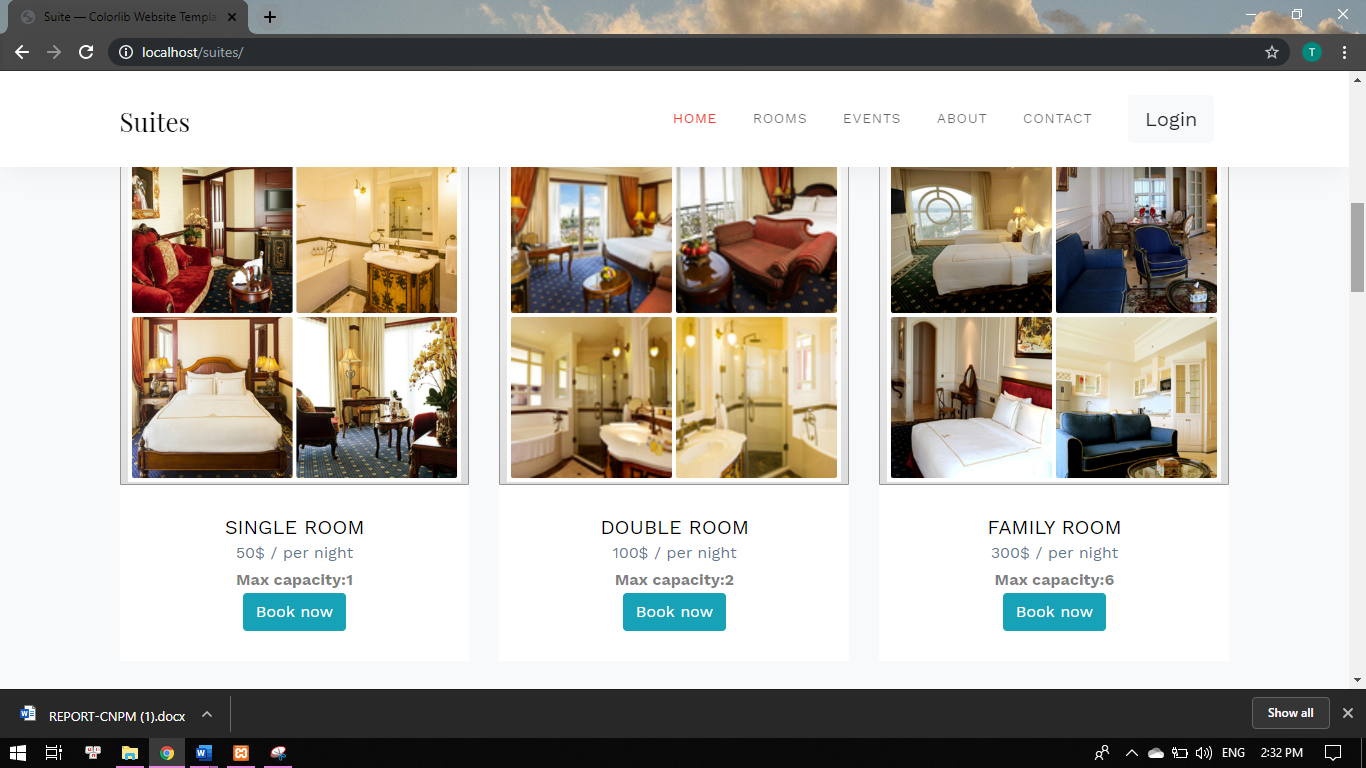


Figure 4.2.2 Main page for customer

If you want to access to admin panel, you have to login first.

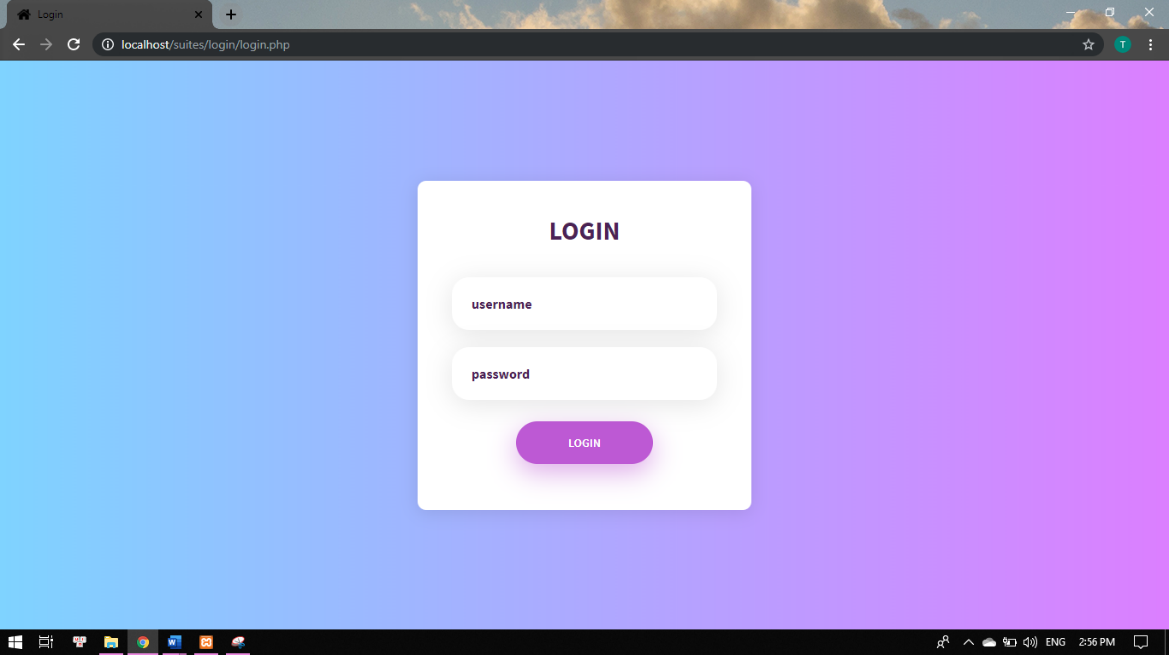


Figure 4.2.3 Login

When admin login successful, they can connect to admin page. Admin can use functions from admin page.

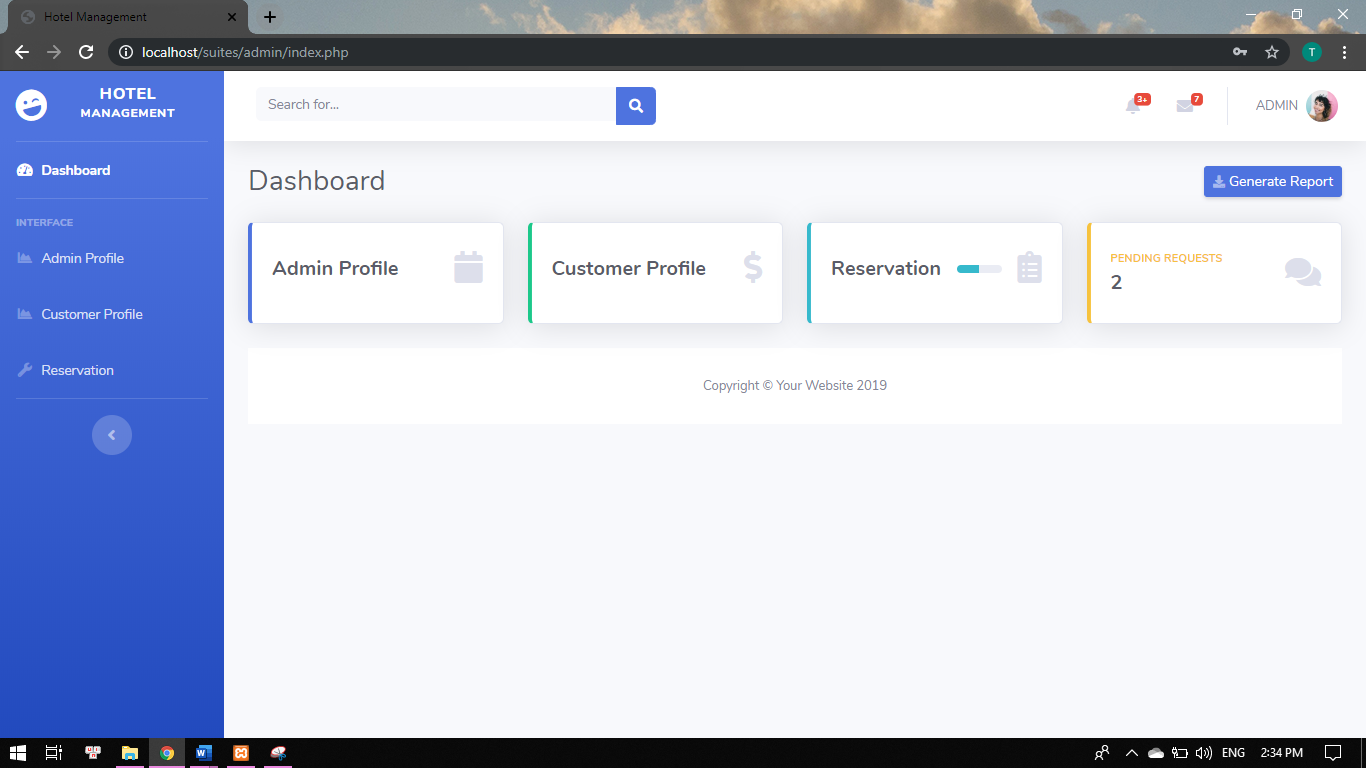


Figure 4.2.4 Main page for admin

In Admin Profile, you can VIEW, ADD, EDIT, DELETE information

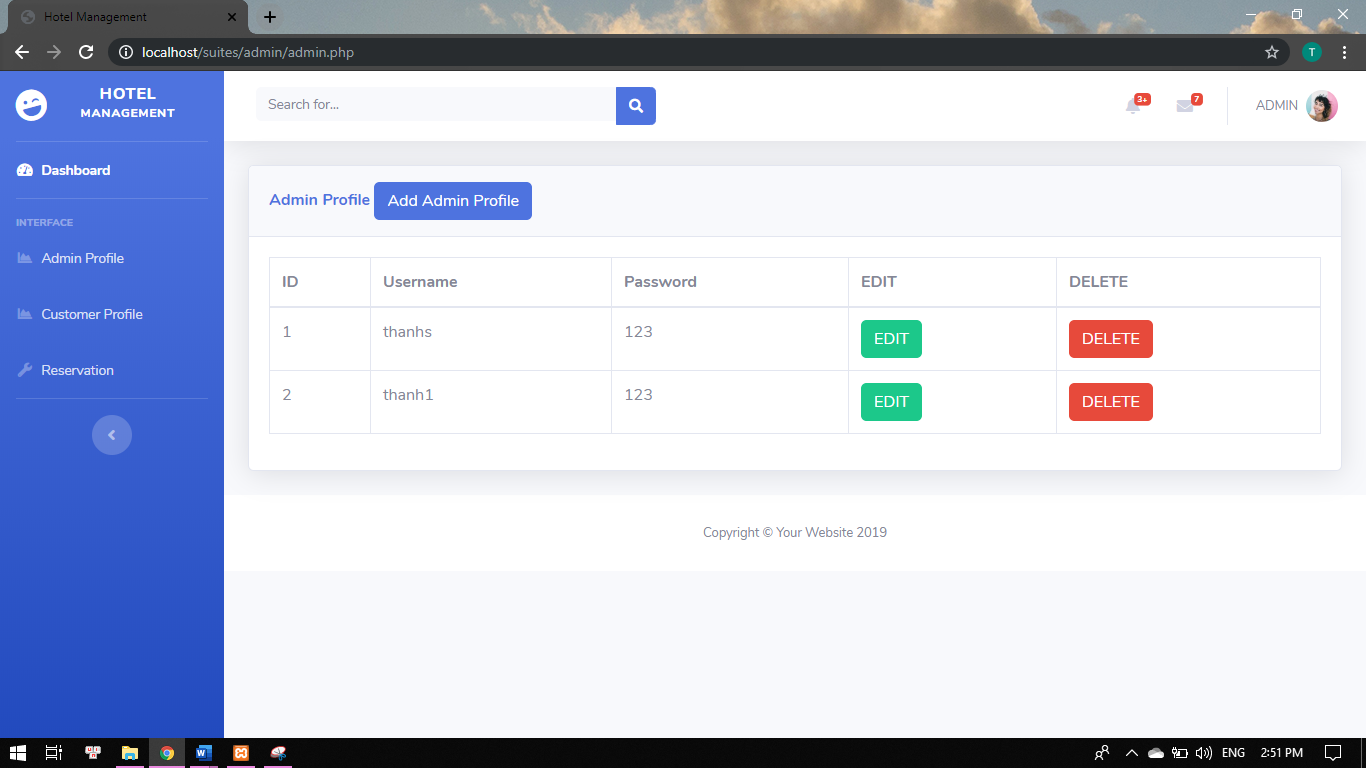


Figure 4.2.5 Admin Profile

In Customer Profile, you can VIEW, ADD, DELETE customer information

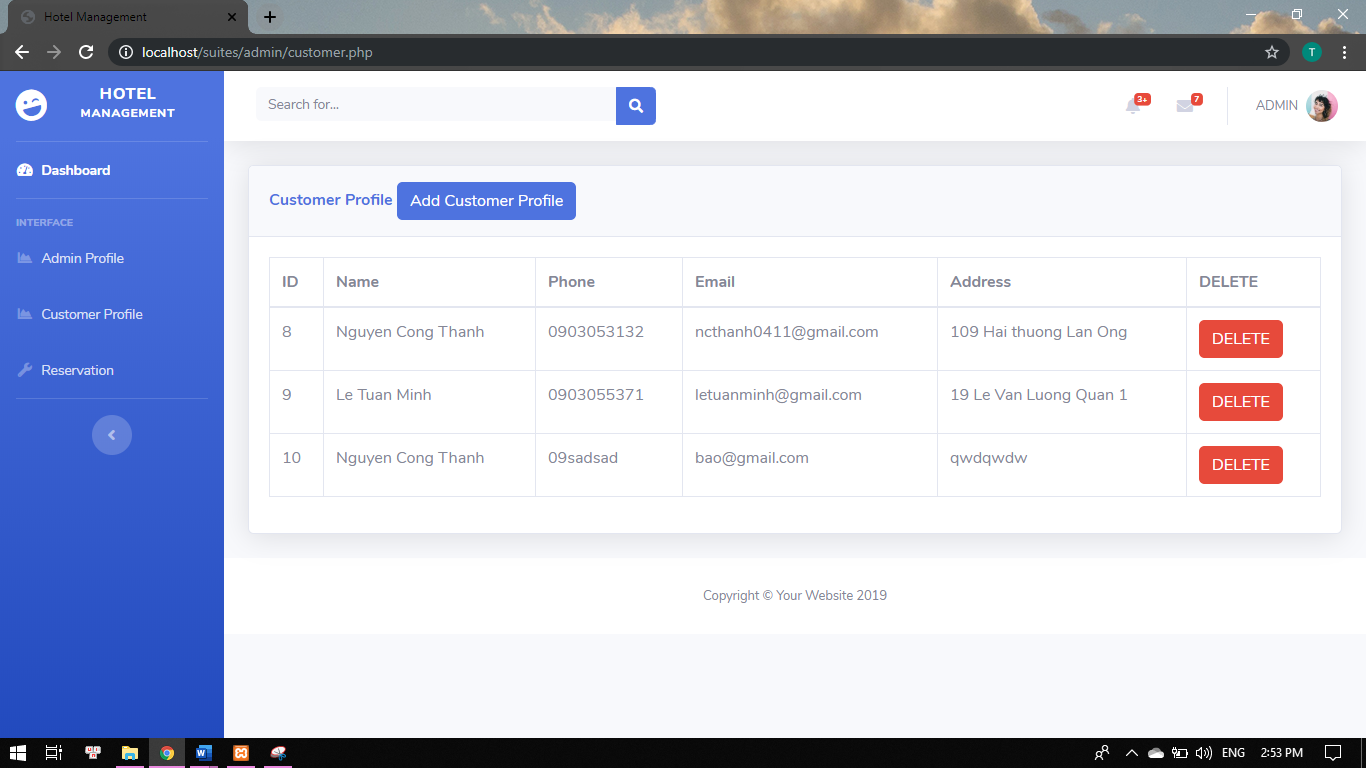
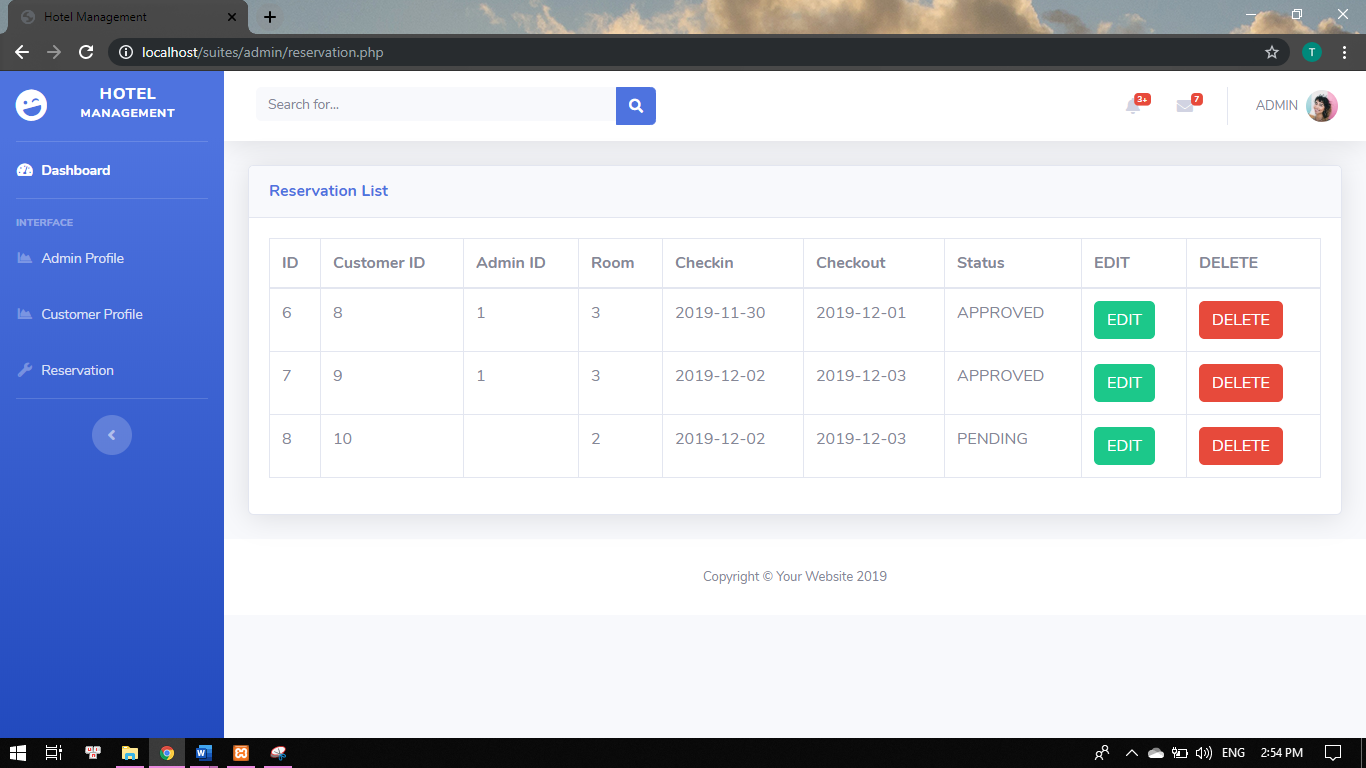


Figure 4.2.6 Customer Profile

In Reservation Profile, you can VIEW, UPDATE, DELETE reservation information

 Figure 4.2.7 Customer Profile

***REFERENCES***

[1] <https://en.wikipedia.org/wiki/HTML>

[2] <https://en.wikipedia.org/wiki/Cascading_Style_Sheets>

[3] <https://en.wikipedia.org/wiki/PHP>

[4] <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>

[5] <https://about.draw.io/entity-relationship-diagrams-with-draw-io/>

[6] <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

[7] <https://www.visual-paradigm.com/support/documents/vpuserguide/94/2577/7025_drawingseque.html>

[8] <https://elit.tdtu.edu.vn/courses/130>