NATIONAL UNIVERSITY OF HO CHI MINH CITY

**UNIVERSITY OF INFORMATION TECHNOLOGY**

****

**UIT**

**TRƯỜNG ĐẠI HỌC**

**CÔNG NGHỆ THÔNG TIN**

**FINAL REPORT**

**INFORMATION SYSTEM ANALYSIS AND DESIGN**

**HOTEL MANAGEMENT SYSTEM**

Authors:

* Phan Phạm Quỳnh Hoa - 19521520
* Lê Tiến Vinh – 19522521
* Lê Đinh Quốc Huy - 19521614
* Nguyễn Thành Đạt – 19521340

Under the Guidance of:

Ms. Nguyễn Thị Kim Phụng

Ho Chi Minh, April 15th 2021

INTRODUCTION

As a matter of first importance, the creators might want to communicate our genuine gratitude to the lecturers at University of Information Technology – National University of Ho Chi Minh City and Faculty of Information System has assisted the creators with having the fundamental information as the establishment for implementing this theme.

In specific, the creators might want to communicate their most profound thanks and appreciation to Ms. Nguyễn Thị Kim Phụng (Lecturer in Information system analyst and design). The lecturer has straightforwardly guided, remedied and contributed numerous important remarks to help the group effectively complete their subject reports.

During one semester of the theme, the creators applied the establishment has collected in the mix with learning and exploring new. From that point, the creators capitalize on what has been gathered to finish the best project report. In any case, during the time spent execution, the creators bunch definitely shortcomings. Hence, the creators are anticipating accepting remarks from the teachers to improve the information that the creators have learned and are for the creators to forge ahead different themes later on.

Sincerely thank you, Ms.Phụng Nguyễn.

Ho Chi Minh, April 15th 2021

EVALUATION

…………………………………………………………………………..………………………………………………………………………………..………………..……………………………………………………………………………………..…………………………………………………………………………………..……………………………………………………………………………………..…………………………………..…………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..………………………………………………………………………………..………………..……………………………………………………………………………………..…………………………………………………………………………………..……………………………………………………………………………………..…………………………………..…………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..………….…………………………………………………………………………..………………………………………………………………………………..………………..……………………………………………………………………………………..…………………………………………………………………………………..……………………………………………………………………………………..…………………………………..…………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..…………

…………………………………………………………………………..………………………………………………………………………………..………………..……………………………………………………………………………………..…………………………………………………………………………………..……………………………………………………………………………………..…………………………………..…………………………………………………..……………………………………………………………………………………..……………………………………………………………………………………..………

Contents

[Chapter 1: Introduction 5](#_Toc75630444)

[1 Project overview 5](#_Toc75630445)

[1.1 Scope and objectives of the project 5](#_Toc75630446)

[1.2 Roles and responsibilities of project members 6](#_Toc75630447)

[1.3 Software tools 7](#_Toc75630448)

[2 Software development model 7](#_Toc75630449)

[2.1 Model introduction 7](#_Toc75630450)

[2.2 WBS 8](#_Toc75630451)

[Propose 9](#_Toc75630452)

[Chapter 2: Analysis 15](#_Toc75630453)

[1 Requirement determination 15](#_Toc75630454)

[1.1 Survey 15](#_Toc75630455)

[1.2 Organizational structure 16](#_Toc75630456)

[1.3 Business analysis process 17](#_Toc75630457)

[2 Requirements analysis 18](#_Toc75630458)

[2.1 Storage requirements 18](#_Toc75630459)

[2.2 Functional requirements 18](#_Toc75630460)

[2.3 Non-functional requirements 19](#_Toc75630461)

[*Chapter 3: Design - system specification* 21](#_Toc75630462)

[1 Design of processing components 21](#_Toc75630463)

[1.1 Use Case diagram 21](#_Toc75630464)

[1.2 List of system actors 23](#_Toc75630465)

[1.3 List of usecase 24](#_Toc75630466)

[1.4 Use case specification and Activity diagram 24](#_Toc75630467)

[1.4.1 Use case: Login 24](#_Toc75630468)

[1.4.2 Use case: Room management 26](#_Toc75630469)

[1.4.3 Use case: Employee management 28](#_Toc75630470)

[1.4.4 Use case: Customer management 30](#_Toc75630471)

[1.4.5 Use case: Change password 32](#_Toc75630472)

[1.4.6 Use case: Rental history 33](#_Toc75630473)

[1.4.7 Use case: Booking 35](#_Toc75630474)

[1.5 Sequence Diagram 37](#_Toc75630475)

[1.5.1 Sequence: Login 37](#_Toc75630476)

[1.5.2 Sequence: Room management 38](#_Toc75630477)

[1.5.3 Sequence: Employee management 40](#_Toc75630478)

[1.5.4 Sequence: Customer management 42](#_Toc75630479)

[1.5.5 Sequence: Change password 44](#_Toc75630480)

[1.5.6 Sequence: Rental history 44](#_Toc75630481)

[1.6 Class Diagram 45](#_Toc75630482)

[2 Design of data components 46](#_Toc75630483)

[2.1 Entity Relationship diagram 46](#_Toc75630484)

[2.2 Database diagram 46](#_Toc75630485)

[2.3 Data organization 47](#_Toc75630486)

[2.4 Data modeling 47](#_Toc75630487)

[2.4.1 DBO.Staff 47](#_Toc75630488)

[2.4.2 DBO.Client 48](#_Toc75630489)

[2.4.3 DBO.Room 48](#_Toc75630490)

[2.4.4 DBO.Booking 49](#_Toc75630491)

[2.4.5 DBO.Users 49](#_Toc75630492)

[3 Design interface components 49](#_Toc75630493)

[3.1 Interface design 49](#_Toc75630494)

[3.2 Interface description 50](#_Toc75630495)

[3.2.1 Login 50](#_Toc75630496)

[3.2.2 Hotel Management 51](#_Toc75630497)

[3.2.3 The Info tab in Hotel Management 51](#_Toc75630498)

[3.2.4 Change Password form 52](#_Toc75630499)

[3.2.5 The Room management tab in the Hotel management 53](#_Toc75630500)

[3.2.6 The Customer management tab in Hotel Management 53](#_Toc75630501)

[3.2.7 The Employee management tab in the Hotel management 54](#_Toc75630502)

[3.2.8 The Rental history tab in the Hotel management 55](#_Toc75630503)

[Chapter 4: Testing 57](#_Toc75630504)

[1 Test Case 1 57](#_Toc75630505)

[2 Test Case 2 58](#_Toc75630506)

[Chapter 5: Summary – evaluation 60](#_Toc75630507)

[1 Summary 60](#_Toc75630508)

[2 Limitations 60](#_Toc75630509)

[3 Future & Scope 60](#_Toc75630510)

[*Appendix* 62](#_Toc75630511)

## Chapter 1: Introduction

# Project overview

While global economic integration makes the hotel industry market wider, it also makes the hotel industry facing increasingly fierce competition and rising customer expectations, forcing the insider to expand their customers constantly, improve service quality, lower administrative costs and improve customer satisfaction with new ways to enhance the hotel’s core competence. Which is one of the effective means of mass application of advanced information technology, especially with the constant deepening of information, change hospitality industry competition and management models which in the traditional sense, in order to win new competitive advantages. Leading hotels in the world have been falling over themselves to explore, implement and promote integrated hotel management information. Modern hotel is the integration of consumer sites including rooms, business culture, and a variety of other services. Due to a huge membership and service projects, so naturally, it’s the amount of carried in its management, in order to improve work efficiency, reduce operating costs, improve service quality and management levels, promote economic efficiency, we must use the computers to progress modern information management. The great development of modern computer technology, provide a favourable opportunity for hotel management to change.

## Scope and objectives of the project

***Objectives of the project:***

Build "Hotel Management Software" to meet the accompanying necessities:

• There is a decentralized system.

* Manager is allowed to add a new employee and manage all employees,
* Receptionists are allowed to view information and perform their functions.

• Manage rooms, update room status, make reservations

• Permitting to see client data, add clients, alter client data, erase clients

• View booking history, check-out date

**Scope:**

* For users:
* Clear interface, easy to use.
* Limited software errors, stable on most operating systems.
* Functional requirements:
* Customize, search, modify easily, efficiently, and with high security
* Request helpfulness

*For management*

* Search, add, delete, update and store booking information.
* Add, delete, edit client information.
* Add, delete, edit employee information.

*For the receptionist:*

* Search, add, delete, update and store booking information.
* Add, delete, edit client information.

Due to the limited time to implement the project, the development of “Hotel management system” will focus on the following main functions: Staff management, Client management, Room management, Booking and Rental history.

## Roles and responsibilities of project members

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Full name | Role | Main responsibility |
| 19521520 | Phan Phạm Quỳnh Hoa | Team lead |  |
| 19521614 | Lê Đinh Quốc Huy | Developer – Font end |  |
| 19521340 | Nguyễn Thành Đạt | Analyst |  |
| 19522521 | Lê Tiến Vinh | Developer – Back end |  |

## Software tools

During the implementation of the project, the group utilized several supporting tools for Software development, such as:

* Database management system: SQL server

• Software build tools: Apache Netbeans IDE

• Manage source code: GitHub

• Project management: Microsoft Project

• Analysis: Microsoft Visio

# Software development model

## Model introduction

The waterfall model is intuitively the most obvious way to develop software. Though the waterfall model is elegant and intuitively obvious, it is not a practical model in the sense that it can not be used in actual software development projects. Thus, this model can be considered as a theoretical way of developing software. But all other life cycle models are essentially derived from the classical waterfall model. So, in order to be able to appreciate other life cycle models it is necessary to learn the classical waterfall model. The full stages of the waterfall model:

• Feasibility study: collect and analyze requirements, then write them into requirements specification document.

• Design analysis: Analyze and design software systems, determine the overall system architecture of the software.

• Coding: The system is developed by unit and integrated into the next phase. Each Unit is developed and tested by dev is called Unit Test.

• Testing: Install and test the software. The main work of this section is to check and correct all the errors found so that the software works correctly and following the requirements specification document.

• Integration: Deploy the system in the customer environment and launched.

• Operation and maintenance: Maintain the system when there are any changes from customers and users.

Diagram

Description automatically generated

*Advantages:*

• Easy to use, easy to access, easy to manage.

• Products develop in well-defined stages.

• Validation at each stage, ensuring early detection of errors.

*Defect*

• Less flexibility, limited adjustment range.

• It is difficult to measure development in each stage.

• The model is not suitable for long, ongoing projects, or complex projects with many changes in requirements.

• Difficult to return once a certain period has ended

## WBS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task name | Duration | Start | Finish | Predecessors | Resource Names |
| **Building a** **hotel management system** | **57 days** | **Mon 4/12/21** | **Fri 6/18/21** |  |  |
| **Kick-off** | **0 days** | **Mon 4/12/21** | **Mon 4/12/21** |  |  |
| **Determine** | **4 days** | **Mon 4/12/21** | **Thu 4/15/21** |  |  |
| Regular group meetings | 0.5 days | Mon 4/12/21 | Mon 4/12/21 |  | Phan Phạm Quỳnh Hoa,Nguyễn Thành Đạt,Lê Tiến Vinh,Lê Đinh Quốc Huy |
| Identify problems, goals, results | 0.5 days | Mon 4/12/21 | Mon 4/12/21 | 4 | Phan Phạm Quỳnh Hoa |
| Writing project outlines and feasibility studies | 0.5 days | Tue 4/13/21 | Tue 4/13/21 | 5 | Phan Phạm Quỳnh Hoa |
| **Identify risks** | **1 day** | **Tue 4/13/21** | **Wed 4/14/21** |  |  |
| Risk assumption | 0.25 days | Tue 4/13/21 | Tue 4/13/21 | 6 | Phan Phạm Quỳnh Hoa |
| Risk Handling Solutions | 0.25 days | Tue 4/13/21 | Tue 4/13/21 | 8 | Phan Phạm Quỳnh Hoa |
| Make a risk list | 0.5 days | Wed 4/14/21 | Wed 4/14/21 | 9 | Phan Phạm Quỳnh Hoa |
| **Initial planning** | **1 day** | **Wed 4/14/21** | **Thu 4/15/21** |  |  |
| Personnel Planning | 0.25 days | Wed 4/14/21 | Wed 4/14/21 | 10 | Phan Phạm Quỳnh Hoa |
| Resource planning Propose solutions to users | 0.25 days | Wed 4/14/21 | Wed 4/14/21 | 12 | Phan Phạm Quỳnh Hoa |
| Estimate project time andcost | 0.25 days | Thu 4/15/21 | Thu 4/15/21 | 13 | Phan Phạm Quỳnh Hoa |
| Write an original plan | 0.25 days | Thu 4/15/21 | Thu 4/15/21 | 14 | Phan Phạm Quỳnh Hoa |
| Propose **s**olutions to users | **0.5 days** | **Thu 4/15/21** | **Thu 4/15/21** | 15 | Phan Phạm Quỳnh Hoa |
| **Complete the specified stage** | **0 days** | **Thu 4/15/21** | **Thu 4/15/21** | 16 |  |
| **Analyze** | **5 days** | **Fri 4/16/21** | **Wed 4/21/21** |  |  |
| Regular group meetings | 0.5 days | Fri 4/16/21 | Fri 4/16/21 | 17 | Phan Phạm Quỳnh Hoa,Nguyễn Thành Đạt,Lê Tiến Vinh,Lê Đinh Quốc Huy |
| **User request analysis** | **2.5 days** | **Sat 4/17/21** | **Tue 4/20/21** |  |  |
| Non-functional request analysis | 1 day | Sat 4/17/21 | Sat 4/17/21 | 19 | Nguyễn Thành Đạt |
| Functional request analysis | 1 day | Mon 4/19/21 | Mon 4/19/21 | 21 | Nguyễn Thành Đạt |
| Write analytical reports | 0.25 days | Tue 4/20/21 | Tue 4/20/21 | 22 | Nguyễn Thành Đạt |
| Write function specifications | 0.25 days | Tue 4/20/21 | Tue 4/20/21 | 23 | Nguyễn Thành Đạt |
| **Analyze resource status** | **0.5 days** | **Tue 4/20/21** | **Tue 4/20/21** | 24 | Nguyễn Thành Đạt |
| **Re-examine analysis** | **0.25 days** | **Wed 4/21/21** | **Wed 4/21/21** | 25 | Phan Phạm Quỳnh Hoa |
| **Overall design** | **0.25 days** | **Wed 4/21/21** | **Wed 4/21/21** | 26 | Lê Đinh Quốc Huy |
| **Write a project implementation plan** | **0.5 days** | **Wed 4/21/21** | **Wed 4/21/21** | 27 | Phan Phạm Quỳnh Hoa |
| **Complete the analysis phase** | **0 days** | **Wed 4/21/21** | **Wed 4/21/21** | 28 |  |
| **Design** | **11.5 days** | **Thu 4/22/21** | **Fri 5/7/21** |  |  |
| Regular meetings | 4 hrs | Thu 4/22/21 | Thu 4/22/21 | 18 | Phan Phạm Quỳnh Hoa,Nguyễn Thành Đạt,Lê Tiến Vinh,Lê Đinh Quốc Huy |
| **Overall design** | **9.5 days** | **Thu 4/22/21** | **Wed 5/5/21** |  |  |
| **Data design** | **1 day** | **Thu 4/22/21** | **Fri 4/23/21** | 31 | Lê Đinh Quốc Huy |
| **Component design and processing** | **5 days** | **Fri 4/23/21** | **Thu 4/29/21** |  |  |
| Processing diagram design | 1.5 days | Fri 4/23/21 | Sat 4/24/21 | 33 | Lê Đinh Quốc Huy |
| Functional diagram design | 1.5 days | Mon 4/26/21 | Tue 4/27/21 | 35 | Lê Đinh Quốc Huy |
| Screen map design | 2 days | Tue 4/27/21 | Thu 4/29/21 | 36 | Lê Đinh Quốc Huy |
| **Interface design** | **3.5 days** | **Thu 4/29/21** | **Wed 5/5/21** |  |  |
| Sign-in interface | 0.5 days | Thu 4/29/21 | Thu 4/29/21 | 37 | Lê Đinh Quốc Huy |
| Management interface | 1.5 days | Mon 5/3/21 | Tue 5/4/21 | 39 | Lê Đinh Quốc Huy |
| Employee interface | 1.5 days | Tue 5/4/21 | Wed 5/5/21 | 40 | Lê Đinh Quốc Huy |
| **Write design specifications** | **0.5 days** | **Thu 5/6/21** | **Thu 5/6/21** | 41 | Lê Đinh Quốc Huy |
| **Write an acceptance test plan** | **0.5 days** | **Thu 5/6/21** | **Thu 5/6/21** | 42 | Lê Đinh Quốc Huy |
| **Check estimates** | **0.5 days** | **Fri 5/7/21** | **Fri 5/7/21** | 43 | Lê Đinh Quốc Huy |
| **Completion of the design stage** | **0 days** | **Fri 5/7/21** | **Fri 5/7/21** | 44 |  |
| **Implement** | **23.5 days** | **Fri 5/7/21** | **Thu 6/3/21** |  |  |
| Regular meetings | 4 hrs | Fri 5/7/21 | Fri 5/7/21 | 45 | Phan Phạm Quỳnh Hoa,Nguyễn Thành Đạt,Lê Tiến Vinh,Lê Đinh Quốc Huy |
| **Build an interface** | **5.5 days** | **Sat 5/8/21** | **Fri 5/14/21** |  |  |
| **Manage employees tab** | **0.5 days** | **Sat 5/8/21** | **Sat 5/8/21** |  |  |
| Employee information display | 0.5 days | Sat 5/8/21 | Sat 5/8/21 | 47 | Lê Đinh Quốc Huy |
| **Manage Rooms tab** | **1 day** | **Sat 5/8/21** | **Mon 5/10/21** | 50 |  |
| Room information display | 0.5 days | Sat 5/8/21 | Sat 5/8/21 |  | Lê Đinh Quốc Huy |
| Booking screen | 0.5 days | Mon 5/10/21 | Mon 5/10/21 | 52 | Lê Đinh Quốc Huy |
| **Manage Clients tab** | **1.5 days** | **Mon 5/10/21** | **Tue 5/11/21** |  |  |
| Display of customer information | 1.5 days | Mon 5/10/21 | Tue 5/11/21 | 53 | Lê Đinh Quốc Huy |
| **User Information tab** | **1 day** | **Wed 5/12/21** | **Wed 5/12/21** | 55 |  |
| User information view screen | 0.5 days | Wed 5/12/21 | Wed 5/12/21 |  | Lê Đinh Quốc Huy |
| Password change view screen | 0.5 days | Wed 5/12/21 | Wed 5/12/21 | 57 | Lê Đinh Quốc Huy |
| **Rental History tab** | **1 day** | **Thu 5/13/21** | **Thu 5/13/21** | 58 |  |
| Screen view booking history | 1 day | Thu 5/13/21 | Thu 5/13/21 |  | Lê Đinh Quốc Huy |
| Sign-in screen | 0.5 days | Fri 5/14/21 | Fri 5/14/21 | 60 | Lê Đinh Quốc Huy |
| **Build functionality** | **13 days** | **Fri 5/14/21** | **Sat 5/29/21** |  |  |
| **Reception functions** | **6 days** | **Fri 5/14/21** | **Fri 5/21/21** |  |  |
| View employee information | 1 day | Fri 5/14/21 | Sat 5/15/21 | 61 | Lê Tiến Vinh |
| Client management | 1 day | Sat 5/15/21 | Mon 5/17/21 | 64 | Lê Tiến Vinh |
| Room management | 1 day | Mon 5/17/21 | Tue 5/18/21 | 65 | Lê Tiến Vinh |
| Booking | 1 day | Tue 5/18/21 | Wed 5/19/21 | 66 | Lê Tiến Vinh |
| View booking history | 1 day | Wed 5/19/21 | Thu 5/20/21 | 67 | Lê Tiến Vinh |
| Change your password | 1 day | Thu 5/20/21 | Fri 5/21/21 | 68 | Lê Tiến Vinh |
| **Functions for management** | **7 days** | **Fri 5/21/21** | **Sat 5/29/21** |  |  |
| Employee management | 1 day | Fri 5/21/21 | Sat 5/22/21 | 69 | Lê Tiến Vinh |
| Room management | 1 day | Sat 5/22/21 | Mon 5/24/21 | 71 | Lê Tiến Vinh |
| Client management | 1 day | Mon 5/24/21 | Tue 5/25/21 | 72 | Lê Tiến Vinh |
| View employee information | 1 day | Tue 5/25/21 | Wed 5/26/21 | 73 | Lê Tiến Vinh |
| Change your password | 1 day | Wed 5/26/21 | Thu 5/27/21 | 74 | Lê Tiến Vinh |
| View booking history | 1 day | Thu 5/27/21 | Fri 5/28/21 | 75 | Lê Tiến Vinh |
| Client management | 1 day | Fri 5/28/21 | Sat 5/29/21 | 76 | Lê Tiến Vinh |
| **Software finishing** | **2 days** | **Sat 5/29/21** | **Tue 6/1/21** | 77 | Lê Tiến Vinh |
| **Check software functionality** | **6 days** | **Wed 5/26/21** | **Wed 6/2/21** |  |  |
| Test each component | 5 days | Wed 5/26/21 | Tue 6/1/21 | 78FS-5 days | Nguyễn Thành Đạt |
| Write a test record for each component | 1 day | Tue 6/1/21 | Wed 6/2/21 | 80 | Nguyễn Thành Đạt |
| **Write a software construction report** | **0.5 days** | **Tue 6/1/21** | **Tue 6/1/21** | 78 | Phan Phạm Quỳnh Hoa |
| **Develop a system testing plan** | **0.5 days** | **Wed 6/2/21** | **Wed 6/2/21** | 81 | Lê Tiến Vinh |
| **Compile documents for users** | **2 days** | **Wed 6/2/21** | **Thu 6/3/21** | 82 | Phan Phạm Quỳnh Hoa |
| **Completion of the project implementation phase** | **0 days** | **Thu 6/3/21** | **Thu 6/3/21** | 83,84 |  |
| **System testing** | **7 days** | **Fri 6/4/21** | **Fri 6/11/21** |  |  |
| Regular meetings | 4 hrs | Fri 6/4/21 | Fri 6/4/21 | 85 | Phan Phạm Quỳnh Hoa,Nguyễn Thành Đạt,Lê Tiến Vinh,Lê Đinh Quốc Huy |
| Equip the necessary equipment | 0.5 days | Fri 6/4/21 | Fri 6/4/21 | 87 | Lê Đinh Quốc Huy |
| Check the operation of the software | 0.5 days | Sat 6/5/21 | Sat 6/5/21 | 88 | Nguyễn Thành Đạt |
| Planned system testing | 5 days | Sat 6/5/21 | Fri 6/11/21 | 89 | Nguyễn Thành Đạt |
| Write a record of system test results | 0.5 days | Fri 6/11/21 | Fri 6/11/21 | 90 | Nguyễn Thành Đạt |
| Complete stage of system testing | 0 days | Fri 6/11/21 | Fri 6/11/21 | 91 |  |
| **Acceptance test** | **1 day** | **Sat 6/12/21** | **Sat 6/12/21** |  |  |
| Perform the tests given in the acceptance test plan | 2 hrs | Sat 6/12/21 | Sat 6/12/21 | 92 | Lê Tiến Vinh |
| Demo for clients | 2 hrs | Sat 6/12/21 | Sat 6/12/21 | 94 | Lê Tiến Vinh |
| Write a demo report | 2 hrs | Sat 6/12/21 | Sat 6/12/21 | 95 | Lê Tiến Vinh |
| Write a user's confirmation record | 2 hrs | Sat 6/12/21 | Sat 6/12/21 | 96 | Lê Tiến Vinh |
| Complete the acceptance test phase | 0 days | Sat 6/12/21 | Sat 6/12/21 | 97 |  |
| **Operate** | **5 days** | **Mon 6/14/21** | **Fri 6/18/21** |  |  |
| System settings for customers | 1 day | Mon 6/14/21 | Mon 6/14/21 | 98 | Lê Đinh Quốc Huy |
| User training | 1 day | Tue 6/15/21 | Tue 6/15/21 | 100 | Lê Đinh Quốc Huy,Lê Tiến Vinh |
| Write a report on operating results | 0.5 days | Wed 6/16/21 | Wed 6/16/21 | 101 | Lê Đinh Quốc Huy |
| Project audit | 0.5 days | Wed 6/16/21 | Wed 6/16/21 | 102 | Phan Phạm Quỳnh Hoa |
| Project handover | 0.5 days | Thu 6/17/21 | Thu 6/17/21 | 103 | Phan Phạm Quỳnh Hoa |
| Write project transfer documents | 0.5 days | Thu 6/17/21 | Thu 6/17/21 | 104 | Phan Phạm Quỳnh Hoa |
| Experience meetings | 4 hrs | Fri 6/18/21 | Fri 6/18/21 | 105 | Phan Phạm Quỳnh Hoa,Nguyễn Thành Đạt,Lê Tiến Vinh,Lê Đinh Quốc Huy |
| Project summary report | 0.5 days | Fri 6/18/21 | Fri 6/18/21 | 106 | Phan Phạm Quỳnh Hoa |
| **End of project** | **0 days** | **Fri 6/18/21** | **Fri 6/18/21** | 107 |  |

# Chapter 2: Analysis

# Requirement determination

## Survey

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Interview Plan*  *Hotel Management system*  *Author: Phan Phạm Quỳnh Hoa*  *Date: April 18th 2021* | | | | |
| *S. No* | *Subject* | *Requirements* | *Start* | *Finish* |
| *1* | *Purpose of the system* | *In order to improve work efficiency, reduce operating costs, improve service quality and management levels* | *April 18th 2021* | *April 18th 2021* |
| *2* | *Reservation process* | Select an available room and enter reservation information, then return room status to booked, when clients check out, return to available | *April 18th 2021* | *April 18th 2021* |
| *3* | *Client management* | *Add, delete, edit client information* | *April 18th 2021* | *April 18th 2021* |
| *4* | *Employee management* | *Add, delete, edit employee information* | *April 18th 2021* | *April 18th 2021* |
| *5* | *Room management* | *Search, add, delete, edit and store booking data* | *April 18th 2021* | *April 18th 2021* |
| *6* | *Decentralized* *system* | *Manager and Receptionist are allowed to use it.* | *April 18th 2021* | *April 18th 2021* |

|  |  |
| --- | --- |
| Interview Schedule  Hotel Management System | |
| *Interviewee/Place: Deluxe Hotel*  *Start: 8 am April 20th 2021*  *Finish: 10 am April 20th 2021* | *Analyst: Phan Phạm Quỳnh Hoa* |
| *Interview details:*   * *Introduction* * *Overview of system* * *Overview of the interview:* * *Purpose of the system* * *Reservation process* * *Client management* * *Employee management* * *Room management* * *Decentralized system* | *Time expected:*   * *5 min* * *10 min* * *40 min* |
| Overview & Exception |  |

### Organizational structure

Deluxe Hotel

Receptionist

Manager

### 1.3 Business analysis process

|  |  |  |  |
| --- | --- | --- | --- |
| Position | Professional | | Process |
| *Receptionist, Manager* | *Client management* | *Look into client data* | *At the point when you need to view, or refresh client data, the receptionist directs a query of that client data first.* |
| *Update client data* | *In the process of managing client information, the employee can change wrong information, refreshes client data* |
| *Room management* | *Look into room data* | *At the point when you need to view room information, the receptionist directs a query of that room.* |
| *Update room data* | *Employees can update room information.* |
| *Reservation* | *Employees enters client information and chooses an available room.* |
| *Rental history* | *View rental history* | *Employees can view rental history.* |
| *Manager* | *Employee management* | *Look into employee data* |  |
| *Update employee data* | *The manager can change all the employee information* |
| *Detele employee data* | *In case the employee has left work, manager can delete the employee.* |
| *Add new employee* | *When recruiting new employees, the manager enters employee information.* |

# Requirements analysis

## Storage requirements

The Deluxe hotel management system needs to store the information below:

|  |  |
| --- | --- |
| Object | Data storage |
| Client | Client ID, Full name, Phone Number, Date of birth, CCCD |
| Employee | Staff ID, Full name, Gender, Address, Phone Number, Date of birth, Position |
| Room | Room ID, Type of room, Room status, Staff ID |
| Booking | Booking ID, Room ID, Client ID, Check-in date, Check-out date |
| User login | Username, password, authority |

## Functional requirements

* Sign-in function:
* Implementer: Manager, Receptionist
* Input: Enter valid username and password to
* Processing: In case invalid username or password, a message will appear to warn. In contrast, leading to the Homepage with functions per the right permissions.
* View user information
* Function: users are allowed to see their information
* Change password:
* Function: Users can change the default password
* Processing: Enter a new password and then confirm the password
* Client management:
* Functions: Look up client data, add new client, delete client, change client information
* Implementer: Receptionist, manager
* Input: Client information
* Output: Client data table
* Employee management:
* Functions: Add new Employee Details to the Records, edit the details of the employees & modify the employees records
* Implementer: Manager
* Input: Enter employee information
* Output: Employee data table
* Room management
* Functions: change room status
* Implementer: Manager
* Input: Enter room information
* Output: Room data table
* Booking
* Input: Choose an available room, then enter booking information
* Implementer: Manager, Receptionist
* Output: return the room status to booked.
* View rental history
* Function: users are allowed to view rental history, check-in date, check-out date

## Non-functional requirements

* Interface requirements: Simple, user-friendly, and suitable for hotel business.
* Performance requirements:
* The allowable time for the system to respond to information that has accepted a processing request on the user side is 5 seconds.
* The allowable time for returning information search results is 5 seconds.
* Security requirements:

|  |  |  |
| --- | --- | --- |
| Function | Receptionist | Manager |
| Client management | Yes | Yes |
| Room management | Yes | Yes |
| Employee management | No | Yes |
| Rental history | Yes | Yes |

# *Chapter 3: Design - system specification*

# Design of processing components

## Use Case diagram



Employee Management use case:



Customer Management use case:



Room Management use case:



## List of system actors

|  |  |  |
| --- | --- | --- |
| S. No | Authors | Task |
| 1 | Manager | Hotel managers are responsible for managing employees and for planning, marketing, coordinating, and administering hotel services such as accommodation facilities, etc. |
| 2 | Receptionist | The department that has the function of receiving customers, doing check-in, check-out procedures, receiving booking information, making reservations in the hotel. |

## List of usecase

|  |  |  |
| --- | --- | --- |
| S. No | Use Case | Description |
| 1 | Login | Allows users to log into the system with decentralized permissions. |
| 2 | Room management | Permits users to update room information. |
| 3 | Employee management | Permits the manager to add/update/delete employees. |
| 4 | Customer management | Permits users to add/update/delete customers. |
| 5 | Change Password | Users are allowed to change the password differently from the default password. |
| 6 | Check info | Users can see their information. |
| 7 | Rental history | Users can see rental history. |
| 8 | Booking | Enter client information to make a reservation. |

## Use case specification and Activity diagram

### Use case: Login

* Actors: Manager, Receptionist
* Description: Allows users to log into the system with decentralized permissions.
* Pre-condition: The user already has an account.
* Post-condition: Display Main frame.
* Flow of events:



### Use case: Room management

* Actors: Manager, Receptionist
* Description: Permit users to update the room.
* Pre-condition: Login successful.
* Post-condition: All changes are saved to the database.
* Flow of events:



### Use case: Employee management

* Actors: Manager
* Description: Permits the manager to add/update/delete employees.
* Pre-condition: Manager signed in.
* Post-condition: All changes are saved to the database.
* Flow of events:



### Use case: Customer management

* Actors: Manager, Receptionist
* Description: Permits users to add/update/delete customers.
* Pre-condition: Users signed in.
* Post-condition: All changes are saved to the database.
* Flow of events:



### Use case: Change password

* Actors: Manager, Receptionist
* Description: Users are allowed to change the password differently from the default password.
* Pre-condition: Users signed in.
* Post-condition: All changes are saved to the database.
* Flow of events:



### Use case: Rental history

* Actors: Manager, Receptionist
* Description: Users can see rental history.
* Pre-condition: Users signed in.
* Post-condition: Displays.
* Flow of events:



### Use case: Booking

* Actors: Manager, Receptionist
* Description: Enter client information to make a reservation.
* Pre-condition: Users signed in.
* Post-condition: All changes are saved to the database.
* Flow of events:



## Sequence Diagram

### Sequence: Login



### Sequence: Room management



### Sequence: Employee management



### Sequence: Customer management



### Sequence: Change password



### Sequence: Rental history



## Class Diagram



# Design of data components

## Entity Relationship diagram



## Database diagram



## Data organization

**Staff** (indx, StaffID, Fullname, Gender, cAddress, PhoneNumber, Dateofbirth, Position)

**Client** (indx, ClientID, Fullname, PhoneNumber, DateofBirth, CCCD)

**Room** (RoomID, TypeofRoom, cStatus, StaffID)

**Booking** (indx, BookingID, RoomID, ClientID, CheckInDate, CheckOutDate)

**Users** (Usrname, cPassword, authority)

## Data modeling

### DBO.Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | Attribute | Data type | Constraint | Description |
| 1 | Indx | Int |  | Order number |
| 2 | StaffID | Varchar (7) | Primary key | Staff Identification |
| 3 | Fullname | Nvarchar (50) | Not null | Full name |
| 4 | Gender | Varchar (5) | Check | Gender |
| 5 | cAddress | Varchar (50) |  | Address |
| 6 | PhoneNumber | Varchar (20) | Not null, Unique | Phone Number |
| 7 | DateofBirth | Datetime |  | Date of birth |
| 8 | Position | Varchar (20) | Not null | Position |

### DBO.Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | Attribute | Data type | Constraint | Description |
| 1 | Indx | Int |  | Order number |
| 2 | ClientID | Varchar (7) | Primary key | Client Identification |
| 3 | Fullname | Nvarchar (50) | Not null | Full name |
| 4 | PhoneNumber | Varchar (20) | Unique | Phone Number |
| 5 | DateofBirth | Datetime |  | Date of birth |
| 6 | CCCD | Varchar (20) | Unique | Identification card |

### DBO.Room

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | Attribute | Data type | Constraint | Description |
| 1 | RoomID | Varchar (5) | Primary key | Room Identification |
| 2 | TypeofRoom | Varchar (20) |  | Type of room |
| 3 | cStatus | Varchar (20) |  | Room status |
| 4 | StaffID | Varchar (7) | Foreign key | Staff Identification |
| 5 | BookingID | Varchar(7) | Foreign key | Booking Identification |

### DBO.Booking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | Attribute | Data type | Constraint | Description |
| 1 | Indx | Int |  | Order number |
| 2 | BookingID | Varchar (7) | Primary key | Booking Identification |
| 3 | RoomID | Varchar (5) | Foreign key | Room Identification |
| 4 | ClientID | Varchar (7) | Foreign key | Client Identification |
| 5 | CheckInDate | Datetime | Check, Default | Check-in date |
| 6 | CheckOutDate | Datetime | Check, Default | Check-out date |

### DBO.Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | Attribute | Data type | Constraint | Description |
| 1 | Usrname | Varchar (7) | Foreign key | Username |
| 2 | cPassword | Varchar (10) |  | Password |
| 3 | Authority | Char (1) | Check | Authority |

# Design interface components

## Interface design

|  |  |
| --- | --- |
| S. No | Interface |
| 1 | Login |
| 2 | Hotel management |
| 3 | Change password |
| 4 | Booking |

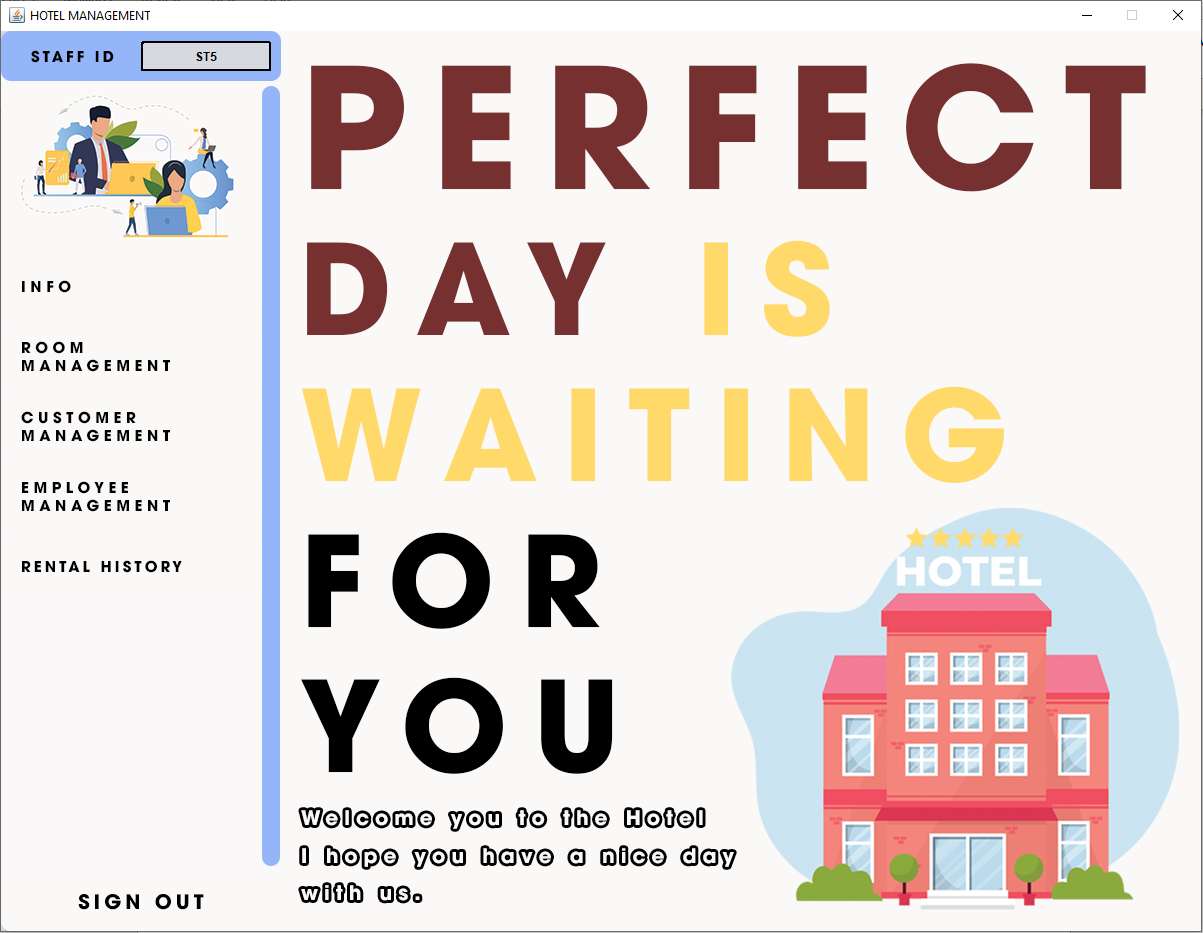
## Interface description

### Login

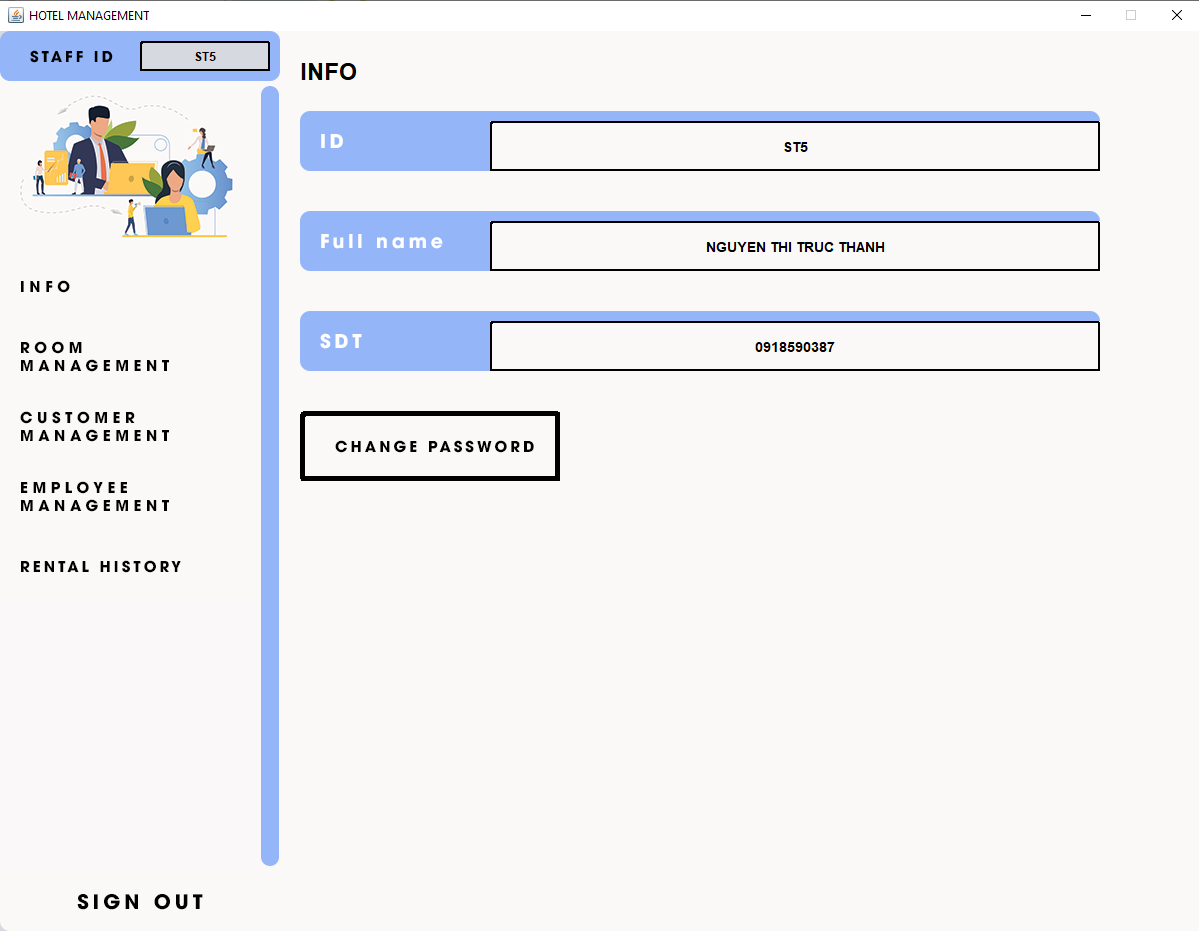
Graphical user interface, application

Description automatically generated

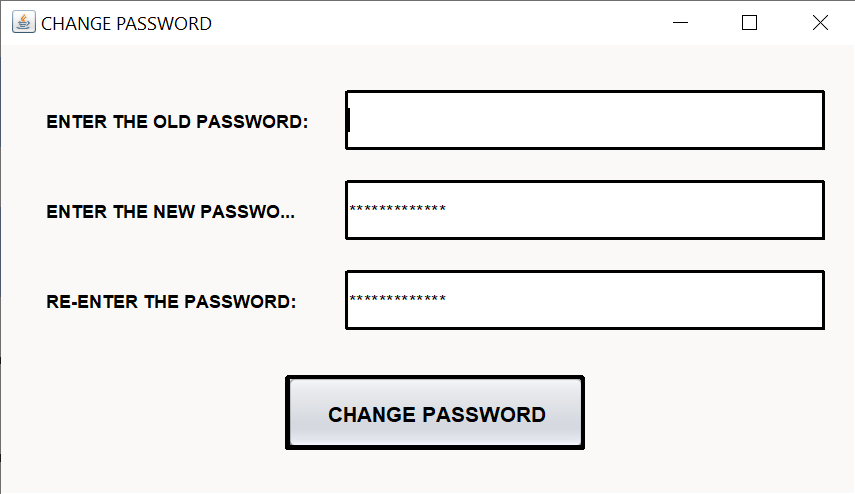
### Hotel Management



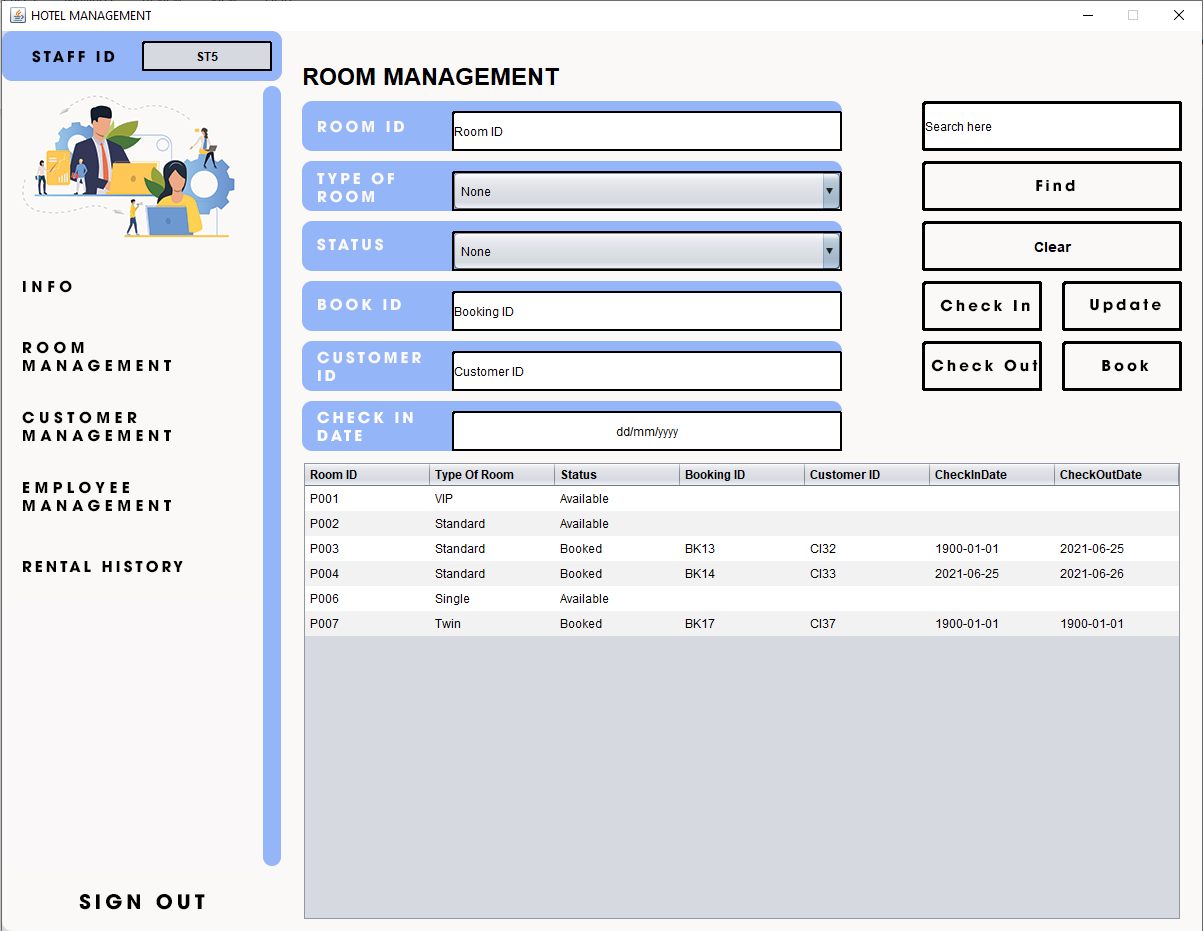
### The Info tab in Hotel Management



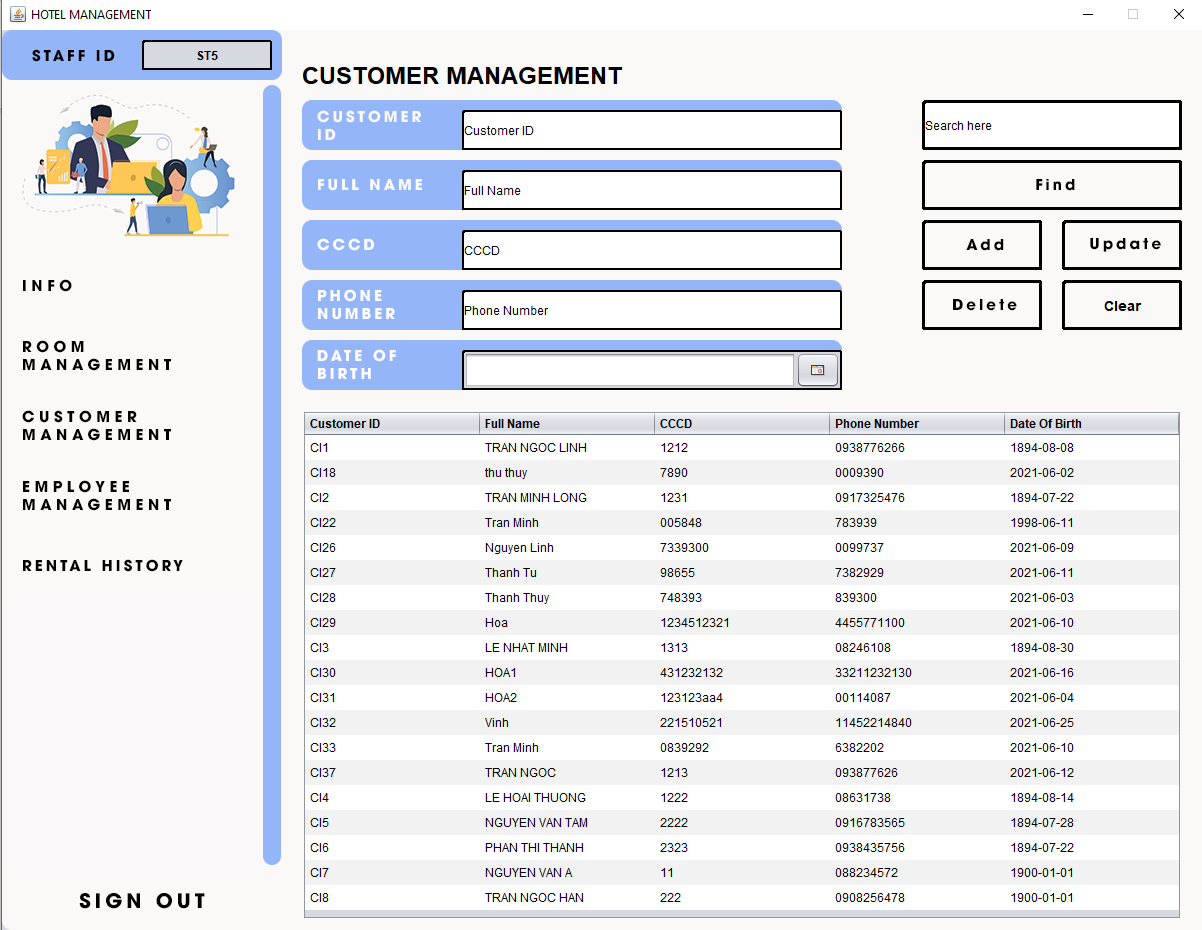
### Change Password form



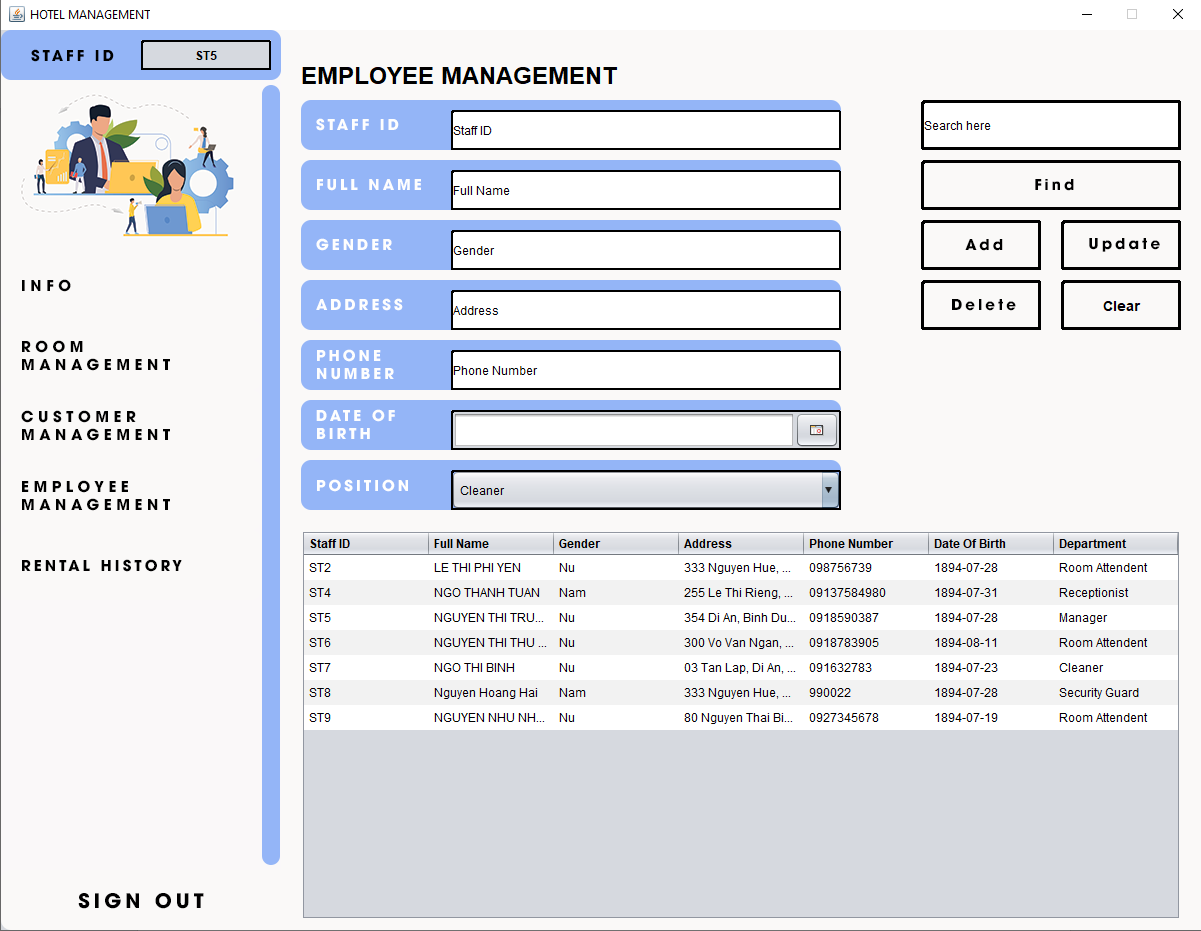
### The Room management tab in the Hotel management



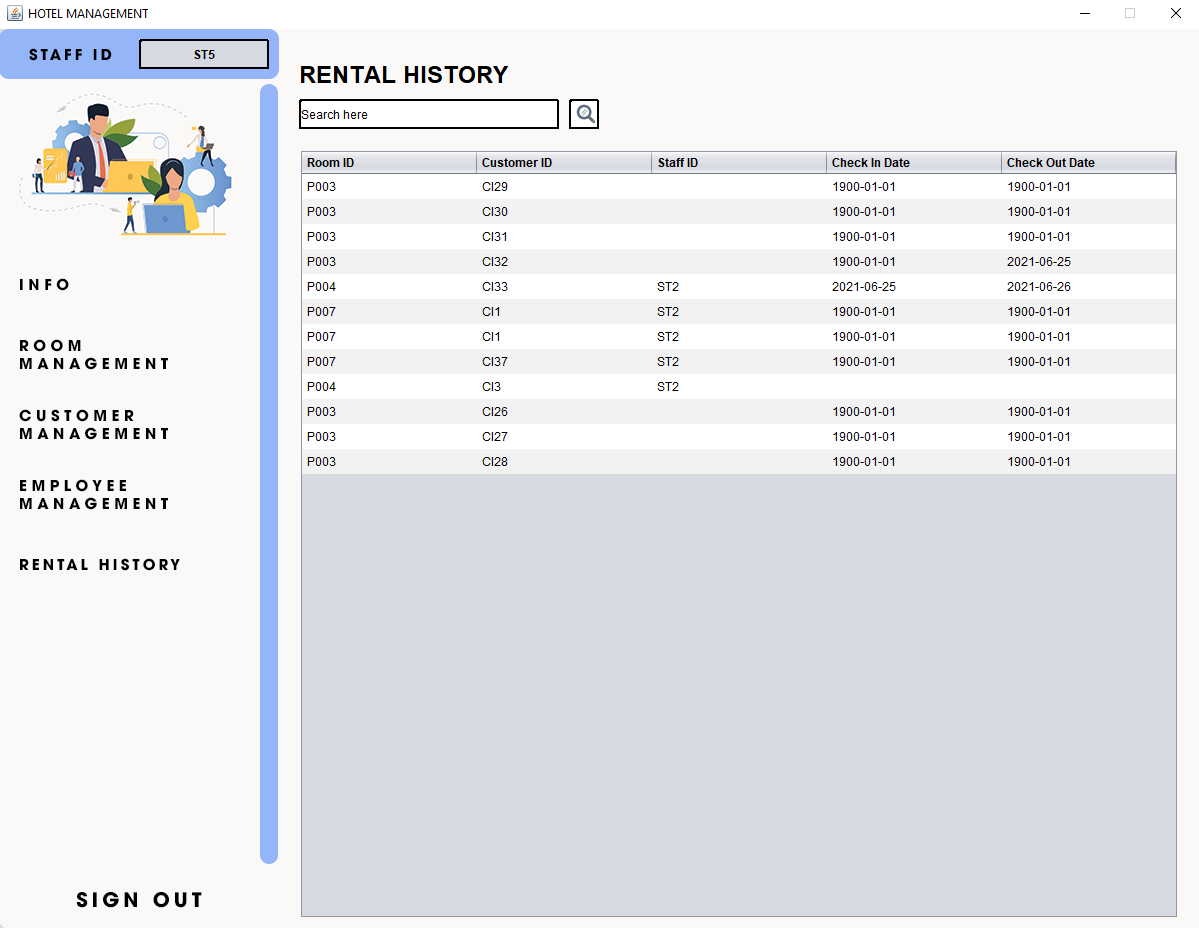
### The Customer management tab in Hotel Management



### The Employee management tab in the Hotel management



### The Rental history tab in the Hotel management



# Chapter 4: Testing

# Test Case 1

|  |  |
| --- | --- |
| Hotel Management System | Test Case |
| Test Case name: TC1 | Test Designed by: Phan Phạm Quỳnh Hoa |
| Test Priority (Low/Medium/High): Medium | Test Designed date: 05/6/2021 |
| Module Name: Login screen | Test Executed by: Phan Phạm Quỳnh Hoa |
| Test Title: Check login function | Test Executed date: 05/6/2021 |
| Description: User verification | |

|  |
| --- |
| Pre-conditions: valid username and password |
| Dependencies |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Steps | Data | Expected result | Actual result | Status |
| 1 | Direct login |  | Possible to login | Possible to login | Succeeded |
| 2 | Enter username | ST5 | Authenticated | Authenticated | Succeeded |
| 3 | Enter password | 12345678 | Authenticated | Authenticated | Succeeded |
| 4 | Click on Sign-in |  | Main screen is displayed. | Main screen is displayed. | Succeeded |

Decision Base Table for Login screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conditions | Rule 1 | Rule 2 | Rule 3 | Rule 4 |
| Username | F | T | F | T |
| Password | F | F | T | T |
| Output | E | E | E | H |

Legend:

* T – Correct username/password
* F – Wrong username/password
* E – Error message is displayed
* H – Home screen is displayed

# Test Case 2

|  |  |
| --- | --- |
| Hotel Management System | Test Case |
| Test Case name: TC2 | Test Designed by: Phan Phạm Quỳnh Hoa |
| Test Priority (Low/Medium/High): Medium | Test Designed date: 05/6/2021 |
| Module Name: DMK screen | Test Executed by: Phan Phạm Quỳnh Hoa |
| Test Title: Check change password function | Test Executed date: 05/6/2021 |
| Description: Change password | |

|  |
| --- |
| Pre-conditions: Valid old password, new password and confirmation password match |
| Dependencies |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Steps | Data | Expected result | Actual result | Status |
| 1 | Enter old password | 12345678 | Valid old password | Valid old password | Possible to change |
| 2 | Enter new password | 123123 |  |  | Possible to change |
| 3 | Enter confirm password | 123123 | New password and confirmation password match | New password and confirmation password match | Possible to change |
| 4 | Press enter |  | Show message “ Your password has been changed” | Show message “ Your password has been changed” | Succeeded |

# Chapter 5: Summary – evaluation

# Summary

The mission of the project is to facilitate easy management and administration of a hotel with capabilities to do Booking of the rooms, creating of a new client, etc. The system lets the user know which all rooms are available for occupancy at any point of time. This makes the booking considerably faster, and thus helps the hotel in better management and reduces a lot of paperwork as well as manpower.

This project is designed to meet the requirement of Offline Hotel Management. Overall, the project taught us the essential skills like:

* Using system analysis and design teachniques like data flow diagram in designing the system.
* Understanding the database handling and query processing.

# Limitations

Due to the limited time to implement the project, the development of the “Hotel management system” also faced some limitations:

* Automatic payments not yet available
* No online access system yet.
* Utmost care and back-up procedures must be established to ensure 100% successful implementation of the computerized hotel system.

# Future & Scope

* This project can be used in the hotel after adding some more useful modules in the project for which hotel are providing services**.**
* Advancement of payment capacities for the system.
* Moreover, we can make this application as online so that we can reserve the tables and do the online payment. So as the demand increase we can add these modules as a future scope**.**

# *Appendix*

* Sample Interview Plan in the slide of chapter 2
* Documents related to the operations of the above objects: (process, form, report, ...)
* Slide lecture on Information System Design Analysis - University of Information Technology – National University of Ho Chi Minh City