**CATALOG**

[1. Introduce 3](#_Toc499983887)

[1.1. Purpose of document 3](#_Toc499983888)

[1.2. Scope 3](#_Toc499983890)

[1.3. Definitions. 3](#_Toc499983891)

[1.4. References 3](#_Toc499983892)

[1.5. Document description 3](#_Toc499983893)

[2. Overview of software 5](#_Toc499983894)

[3. Architectural design 6](#_Toc499983895)

[3.1. Architectural models 6](#_Toc499983896)

[3.2. Architectural description 6](#_Toc499983897)

[4. Database Design 7](#_Toc499983898)

[4.1 Relational Database Design 7](#_Toc499983899)

[4.2 Database Design 7](#_Toc499983900)

[4.3 Database diagrams 11](#_Toc499983901)

[5. Component design 12](#_Toc499983902)

[5.1 Manager 12](#_Toc499983903)

[5.2 Salesman 14](#_Toc499983904)

[5.3 Import staff 15](#_Toc499983905)

[6. User interface design 17](#_Toc499983906)

[6.1 FormLogin 17](#_Toc499983907)

[6.2 FormForgetAccount 18](#_Toc499983908)

[6.3 FormRegister 19](#_Toc499983909)

[6.4 FormMain 20](#_Toc499983910)

[6.5 FormBook 22](#_Toc499983911)

[6.6 FormStationary 25](#_Toc499983912)

[6.7 FormStaff 27](#_Toc499983913)

[6.8 FormCustomer 29](#_Toc499983914)

[6.9 FormSale 31](#_Toc499983915)

[6.10 FormImport 34](#_Toc499983916)

[6.11 FormAccount 37](#_Toc499983917)

# 1. Introduce

### 1.1. Purpose of document

This is a detailed description of hotel management software design, which is the basis for programming functions as well as scripting testing and program testing.

### 1.2. Scope

Documentation is the result of the software design process, which is the communication basis of the developer, which is the basis for testing, commissioning, maintenance, and provision of book management functions.

### 1.3. Definitions.

| **Configuration** | **Definition** | **Explaining** |
| --- | --- | --- |
| HTM | Hotel Management | Name of project |
| DB | Database | Where to store information and allow access. |
| PK | Primary Key | Main key |
| UK | Unique Key | Unique key |
| … | … | … |

### 1.4. References

|  |  |  |
| --- | --- | --- |
| **Number** | **Name of document** | **File Name** |
| 1 | Requirements specification document | Requirements specification document.doc |

### 1.5. Document description

The structure of the document consists of 4 parts:

* **Part 1** **- Introduce**: General introduction of the document, help the reader visualize the content, purpose and general layout of the document.
* **Part 2 - Software Overview:** Pose the most basic requirements that software must have.
* **Part 3** **- Architectural Design**:Building architectural models and describing system architecture.
* **Part 4** **– Data Structure Design**: Describe data dictionary and build data dictionary.

# 2. Overview of software

The software is management solution designed specifically for small to midsize hotels, vacation rentals, hostels and B&B. It was created in 2018 with the aim of minimizing the work involved in managing any hospitality bussiness

|  |  |  |
| --- | --- | --- |
| TT | Problems in the operation of the old system | Requirements |
| 1 | It takes time to find room for customers | Room status are available |
| 2 | Have no database to store customer details and their booking history | Customer details and their booking story are stored in database |
| 3 | Records are duplicated when customers need more services or rooms | Simplify the procedure of multiple service booking |
| 4 | Payment takes time because of manual calculation | All calculations are done automatically by the system |
| 5 | Need to find the booking form when customer want to check out | The system store all the forms |
| 6 | Customers are not satisfied with the current system | Make a questionnaire about booking procedure and customer opinions and requirements |

**Requirements for the new software**

R1 Keep a full list of room types and their details likes room name, type and status;

R2 Store informations of all type of customers;

R3 Monitor all services that customers have used;

R4 Searching and cost evaluation for automatic room booking;

R5 Statistics room capacity, reporting, printing month revenue / year;

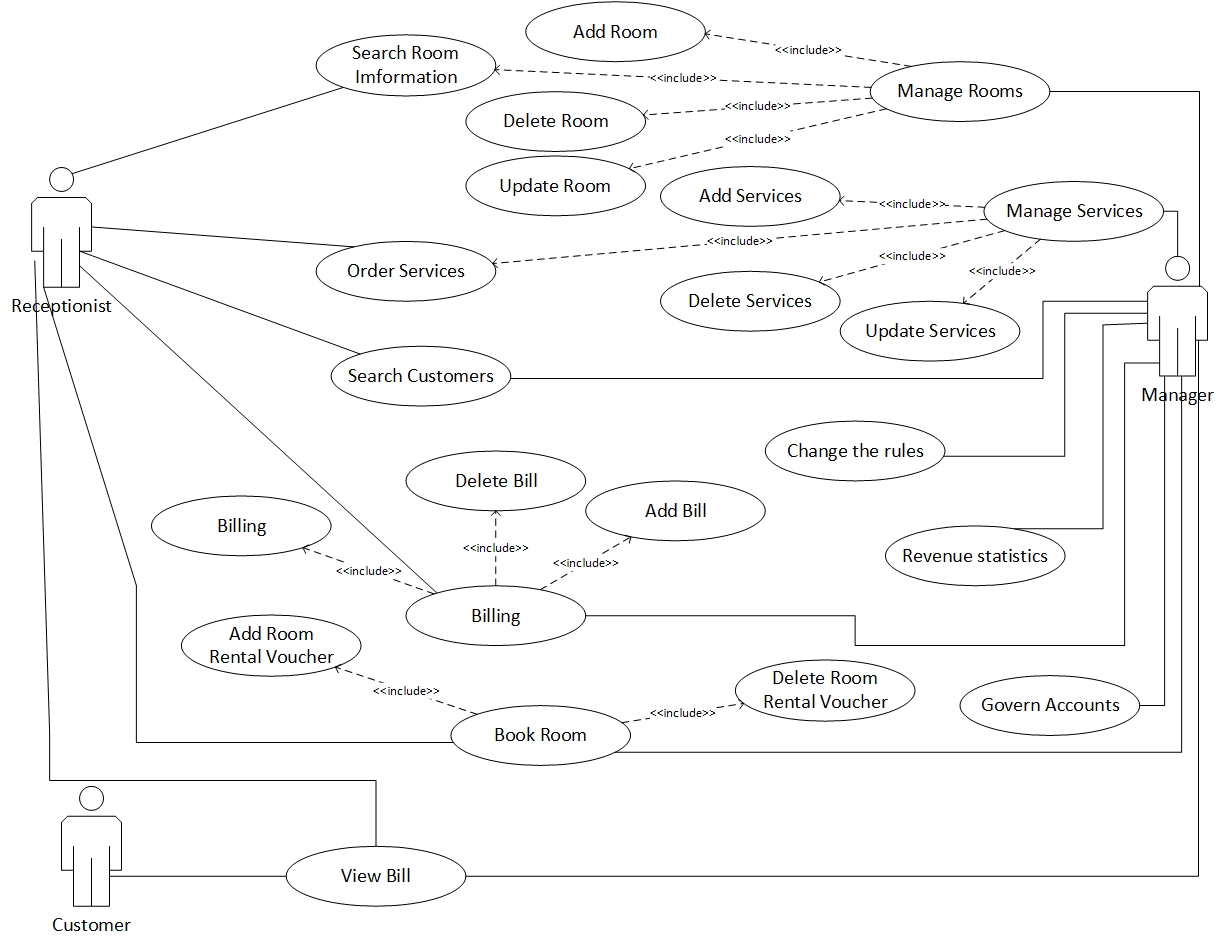
R6 Keep track of the status of each room;

R7 Provide tools to record the equipment to be added that related to the hotel of the staff.

# Hotel management software must response the following general requirements:

* Administering users by access.
* Process catalog update, update data, update user information.

**3. Use case model**



**H1. Use Case Diagram**

**4. Specific requirements for functional software**

**4.1.Use Case "Login" specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | User log in the system based on permision |
| **Description** | Allow user to log in to use the functions that supported for these kinds of user |
| **Actor** | Receptionist and manager |
| **Pre-condition** | Access the hotel management software |
| **Basic flows** | - Enter username  - Enter password |
| **Alternative Flows** | - Fail to log in because of wrong username 🡺 Re-enter  - Fail to log in because password does not match 🡺 Re-enter |
| **Post-condition** | Log in successfully and start to use other functions of the system |

**4.2. Use Case “Create new account” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Provide account for user |
| **Description** | Enter user information |
| **Actor** | Manager |
| **Pre-condition** | - Log in successfully with administrator permision  - Click "Staff" on "Admin"  - Click button "Provide account" in this form |
| **Basic flows** | Enter user name, password for user account |
| **Alternative Flows** | User account information is stored in Database |
| **Post-condition** | Enter username, password for user's account |

**4.3. Use Case “Change Password” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Change user password |
| **Description** | Change user password |
| **Actor** | Manager + Attendants |
| **Pre-condition** | - Log in successfully  - Click "Account" in Main menu  - Choose "Change password" |
| **Basic flows** | Change user account password |
| **Alternative Flows** | Informations are saved in database |
| **Post-condition** | Change user password |

**4.4. Use Case “Book room” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Reservation |
| **Description** | Check room list and show customer the room that not currently in use, then connect to reservation data |
| **Actor** | Receptionist |
| **Pre-condition** | - Log in successfully with Receptionist permision or higher  - Click “Room”, then “Reservation” This time room list displays  - Choose room by images or room number. |
| **Basic flows** | Reservation for customers |
| **Alternative Flows** | Out of room or don’t meet the customer's requirements 🡺 Reservation ends |
| **Post-condition** | Reservation complete, save reservation and customer data. Update room status from "Free" to "Currently in use" |

**4.5. Use Case “Check out” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Check out by demand of customer or out of time |
| **Description** | Process of payment, printing invoice and updating status of room, Delete guest list. |
| **Actor** | Receptionist |
| **Pre-condition** | - Log in successfully with Receptionist permision or higher  -Click “Room”, then “Check out”. This time room list displays  -Click onto room in list, Click “Add invoice” to invoicing.  -Click “Pay” to print invoice for our customer. |
| **Basic flows** | Add and print invoices. |
| **Alternative Flows** | Customer have response about invoice(bill) or want to cancel reservation 🡺 Delete invoice to re-check invoice information. |
| **Post-condition** | List of rooms, customers is updated by data and status |

**4.6. Use Case “Add room” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Add room for rent |
| **Description** | Add room for rent into database. |
| **Actor** | Manager |
| **Pre-condition** | -Log in successfully with manager permision.  - Click “Manager”, click “Room management”. This time room list displays  - Click “Add room” to enter information of the new room. |
| **Basic flows** | Type Room Code, Room Status, Room Type, Price and Note |
| **Alternative Flows** | Newly added room's code is duplicated 🡺 Re-enter |
| **Post-condition** | Room list is updated. |

**4.7. Use Case “Delete room” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Delete room that will be not in use out of the system |
| **Description** | Delete room that will be not in use out of the system |
| **Actor** | Manager |
| **Pre-condition** | - Log in successfully with manager permision.  - Click “Manager”, click “Room management”. This time room list displays  - Click onto the room you want to delete. Click “Delete room” to delete data of that room. |
| **Basic flows** | Delete room from list |
| **Alternative Flows** | Chosen room is at status "Full" or "Repairing" 🡺 Delete room is not permitted |
| **Post-condition** | Room list is updated. |

**4.8. Use Case “Update room” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Edit room information |
| **Description** | Edit room information |
| **Actor** | Manager |
| **Pre-condition** | -Log in successfully with manager permision.  - Click click “Manager”, click “Room management”. This time room list displays  - Click the room that you want to edit. Click click “Update room” |
| **Basic flows** | Enter room status, type and notes |
| **Alternative Flows** | Edited room code is duplicated 🡺 Re-enter |
| **Post-condition** | Room list is updated |

**4.9. Use Case “Add service” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Add new service |
| **Description** | Add new service when add service |
| **Actor** | Manager |
| **Pre-condition** | -Log in successfully with manager permision.  - Click “Manager”, click “Manager service”. This time service list displays  - Click “Add service” then enter information about new service. |
| **Basic flows** | Update service successfully |
| **Alternative Flows** | New service code is duplicated 🡺 Re-enter |
| **Post-condition** | Service list is updated. |

**4.10. Use Case “Delete service” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Delete services that are no longer supported |
| **Description** | Delete services that are no longer supported |
| **Actor** | Manager |
| **Pre-condition** | -Log in successfully with manager permision  - Click click “Manager”, click “Manager service”. This time service list displays  - Click “Delete service” to delete that service. |
| **Basic flows** | Delete service out of database. |
| **Alternative Flows** | None |
| **Post-condition** | Service list is updated. |

**4.11. Use Case “Update service” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Update service information |
| **Description** | Create list of the thing that customer require |
| **Actor** | Manager |
| **Pre-condition** | -Log in successfully with manager permision  - Click click “Manager”, click “Manager service”. This time service list displays  - Click “Update service” to change service information. |
| **Basic flows** | Enter service name, unit, price. |
| **Alternative Flows** | Edited service code is duplicated 🡺 Re-enter |
| **Post-condition** | Service list is updated. |

**4.12. Use Case “Order services” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Order services for customers to select |
| **Description** | Customer call the Receptionist to request for a service. Receptionist will execute the resquest |
| **Actor** | Receptionist and Manager |
| **Pre-condition** | - Log in successfully with Receptionist permision or higher  - Click “Manager”, then “Service management”, choose “Order service”. Or click “Room”, then “Order service”. This time order service table display |
| **Basic flows** | - Enter room code - Click service name - Enter quantity |
| **Alternative Flows** | Close 🡺 Cancel Oerder |
| **Post-condition** | Information about "Order services" is saved to attach into invoice |

**4.13. Use Case “Room search” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Check information of rooms in hotel |
| **Description** | Provide information by room, price, availability |
| **Actor** | Receptionist + Manager |
| **Pre-condition** | - Click search button  - Click "Room search" button |
| **Basic flows** | - Click type or price range.. to show proper room information  - Click empty ,Click button "Reservation". |
| **Alternative Flows** | Room doesn’t found 🡺 Search again |
| **Post-condition** | Information list of rooms |

**4.14. Use Case “Search customer” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Search for detail customer information |
| **Description** | See in detail customer information |
| **Actor** | Receptionist và manager |
| **Pre-condition** | Log in successfully with Receptionist permision or higher   * Click search button * Click “Search customer” in "Admin" |
| **Basic flows** | - Choose customer by writing their name (customer's name, address, customer type, room code) in search box.  - Show customer information |
| **Alternative Flows** | -Search is failed because of wrong information 🡺 Re-enter |
| **Post-condition** | List of customers with detail informations |

**4.15. Use Case “Change regulation” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Reinstall important value of the system |
| **Description** | Update: price, surcharge, coefficient, number of customers |
| **Actor** | Manager |
| **Pre-condition** | - Log in successfully with manager permision  - Click "System", choose "Change regulation" |
| **Basic flows** | - Choose parameters  - Enter new values  - Click "Update" to save |
| **Alternative Flows** | Parameters are not entered 🡺 No update occurs  Close 🡺 Section ends |
| **Post-condition** | Change regulation of the system |

**4.16. Use Case "Turnover statistics by room type"**

|  |  |
| --- | --- |
|  | |
| **Purpose** | Check turnover statistic by room type |
| **Description** | Calculate turnover statistics in each room in specific period of time |
| **Actor** | Manager |
| **Pre-condition** | - Log in successfully with manager permision  - Click "Turnover" in "Admin"  -Click "Turnover by room type". |
| **Basic flows** | - Click reservation time.  -Click button "Show statistic" |
| **Alternative Flows** | -Enter wrong format data type |
| **Post-condition** | Turnover statistics in each room in specific period of time |

**4.17. Use Case “Show used service list” specifications**

|  |  |
| --- | --- |
|  | |
| **Purpose** | To compare with the invoice (bill) |
| **Description** | When customer requests service, the history will be stored |
| **Actor** | Manager |
| **Pre-condition** | - Log in successfully with Receptionist or manager permision  - Click “Invoice list” |
| **Basic flows** | Enter room code and reservation time |
| **Alternative Flows** | Wrong room code 🡺 Re-enter  Reservation time doesn’t not match 🡺 Re-enter |
| **Post-condition** | Corresponding service list is shown |

# 5. Architecture design

5.1 Software architecture

### 5.1.1 Architecture models

The system is divided into 7 modules as follows:

- Room Management: Perform functions related to: Add Room, Update Room, Delete Room, Search Room

- Service Management: Perform functions related to: Add Service, Update Service, Delete Service, Order Service

- Lend Room: Perform functions related to: Add Booking Form, Delete Booking Form.

- Payment: Perform functions related to: Add Bill, Delete Bill,

- Search Customer (Manage Customer): Perform functions related to: Search Information of Customer

- Statistic: Perform functions related to: Revenue Statistics

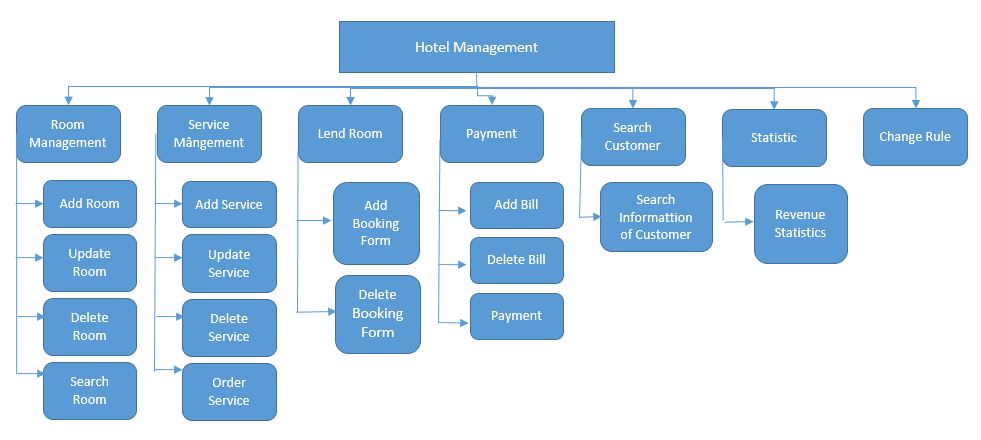
- Change Rule: Perform functions related to: Change rule in hotel

Physical architecture: The Client-Server architecture consists of two distinct components:

- The server plays the role of providing functionality and storing data

- The client as a consumer uses those functions.

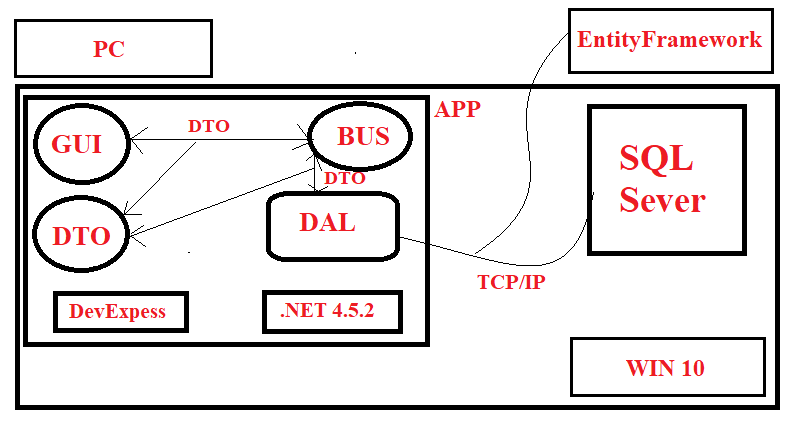
### 5.1.2. Architecture description



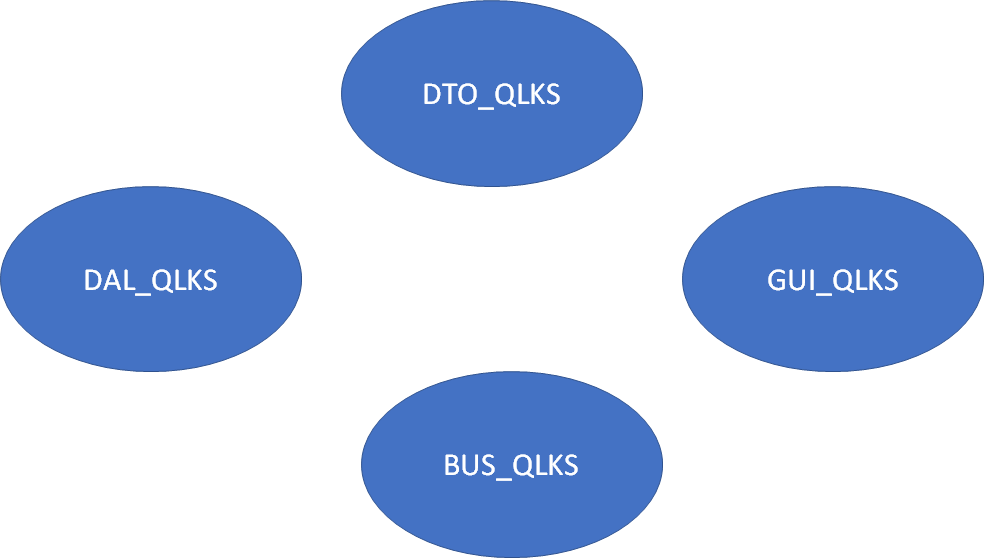
* 1. **. Hardware architecture design**

**5.2.1 General model of architechture**

This application is a 3-tier application , using EntityFramework

****

* + 1. **Component and package list**

****

GUI\_QLKS can link with BUS\_QLKS and DTO\_QLKS

BUS\_QLKS can link with DAL\_QLKS and DTO\_QLKS

DAL\_QLKS can link with DTO\_QLKS

* + 1. **Explaination for interactions between components**

GUI\_QLKS contains forms for application user to send input, requirements and get output information. Right here in UI, informations will be checked, if everything is OK, they will be transfer to BUS\_QLKS for the next process

BUS\_QLKS have classes that responsibility for calculationgồm các class có chức năng tính toán theo đúng yêu cầu GUI\_QLKS đã gửi, nếu không cần đến Database thì sẽ gửi trả kết quả về GUI\_QLKS, ngược lại nó sẽ đẩy dữ liệu (thông tin đã xử lý) xuống DAL\_ QLKS.

DAL\_ QLKS gồm các class có chức năng thao tác với Database và trả kết quả về cho BUS\_QLKS.

DTO\_QLKS gồm các class mô hình hoá lại các đối tượng lưu trữ trong database nhằm mục đính dễ dàng trung chuyển dữ liệu giữa các layer

# 6. Database Design

### 6.1 Relational Database Design

* **Entities**

Based on the specific activities of the hotel management system and based on the above functional decomposition model, we identify the entities: (viết in hoa)

+ Customer (KHACH\_HANG)

+ Report density of use (BAOCAO\_MATDOSUNGPHONG)

+ Bill Detail (CHI\_TIET\_HOA\_DON)

+ Booking Form Detail (CHI\_TIET\_PHIEU\_THUE)

+ Booking Form (PHIEU\_THUE\_PHONG)

+ Room (PHONG)

+ Type Room (LOAI\_PHONG)

+ Bill (HOA\_DON)

+ Order Sevice(DANG\_KI\_DICH\_VU)

+ Service (DICH\_VU)

+ Type Customer(LOAI\_KHACH\_HANG)

+ Report Density of Use Detail(CHITIET\_BAOCAOMATDOSUNG)

+ Revenue Report Detail (CHITIET\_BAOCAODOANHTHU)

+ User (NGUOI\_DUNG)

+ Report Revenue By Type Room (BAOCAO\_DOANHTHUTHEOLOAIPHONG)

* **Relationship between entities**
* Customer lend room
* Type customer has customers
* Customer has Booking Forms
* Booking Form contains customer's information of rental room and checking out
* There are many types of room
* Each room type has specific monthly turnover report
* When customer check out, they receive bill with corresponding room
* The room has its own report about using density
* Each room provides all services of the hotel

### 6.2 Database Design

**Specification tables**

1.KHACH\_HANG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaKhachHang | int | Primary key | Customer ID |
| 2 | TenKhachHang | nvarchar(50) |  | Cutomer name |
| 3 | CMND | nvarchar(15) |  | Identity card number |
| 4 | DiaChi | nvarchar(50) |  | Customer address |
| 5 | MaLoaiKhach | nvarchar(10) | Foreign key | Customer type code |

2.BAOCAO\_DOANHTHUTHEOLOAIPHONG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaBCDoanhThu | int | Primary key | Turnover report code |
| 2 | ThangBaoCao | int |  | Month of report |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaBCMatDoSuDung | int | Primary key | Using density report code |
| 2 | ThangBaoCao | int |  | Month of report |

3.BAOCAO\_MATDOSUNGPHONG

4.CHI\_TIET\_HOA\_DON

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaChiTietHD | int | Primary key | Billing details code |
| 2 | MaHoaDon | int | Primary key | Billing code |
| 3 | SoNgayThue | int |  | Days |
| 4 | DonGia | money |  | Price |
| 5 | ThanhTien | money |  | Total amount |
| 6 | NgayThanhToan | date |  | Pay day |
| 7 | MaPhong | int | Primary key | Room code |

5.CHI\_TIET\_PHIEU\_THUE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaChiTietPT | int | Primary key | Bill details code |
| 2 | MaKhachHang | int | Primary key | Customer code |
| 3 | MaPhieuThu | int | Primary key | Receipt code |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaPhieuThue | int | Primary key | Receipt code |
| 2 | NgayBatDauThue | date |  | Starting date |
| 3 | MaPhong | int | Foreign key | Room code |
| 4 | NgayTraPhongDK | date |  | Checkout date |

6.PHIEU\_THUE\_PHONG

7.PHONG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaPhong | int | Primary key | Room code |
| 2 | TinhTrang | date |  | Status |
| 3 | MaLoaiPhong | int | Foreign key | Room type code |
| 4 | GhiChu | date |  | Note |

8.LOAI\_PHONG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaLoaiPhong | Nvarchar(10) | Primary key | Room type code |
| 2 | TenLoaiPhong | Nvarchar(50) |  | Room type name |
| 3 | DonGia | monkey |  | Price |

9.HOA\_DON

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaHoaDon | Nvarchar(10) | Primary key | Bill code |
| 2 | TenHoaDon | Nvarchar(50) |  | Bill name |
| 3 | TriGia | monkey |  | Value |

10.DANG\_KI\_DICH\_VU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaDangKi | int | Primary key | Registration code |
| 2 | MaDichVu | int | Primary key | Service code |
| 3 | MaPhong | int | Primary key | Room code |
| 4 | SoLuong | int |  | Amount |

11.DICH\_VU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaDichVu | int | Primary key | Service code |
| 2 | TenDichVu | Nvarchar(50) |  | Service name |
| 3 | DonViTinh | Nvarchar(50) |  | Unit |
| 4 | DonGia | monkey |  | Price |

12.LOAI\_KHACH\_HANG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaLoaiKhach | Nvarchar(10) | Primary key | Customer type code |
| 2 | TenLoaiKhach | Nvarchar(50) |  | Customer type name |
| 3 | GhiChu | Ntext |  | Note |

13.CHITIET\_BAOCAOMATDOSUNG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaBCCTMatDoSuDung | int | Primary key | Using density detail report code |
| 2 | MaPhong | int | Primary key | Room code |
| 3 | MaBCMatDoSuDung | int | Primary key | Using density report code |
| 4 | SoNgayThue | int |  | Rental days |
| 5 | Tile | tile |  |  |

14.CHITIET\_BAOCAODOANHTHU

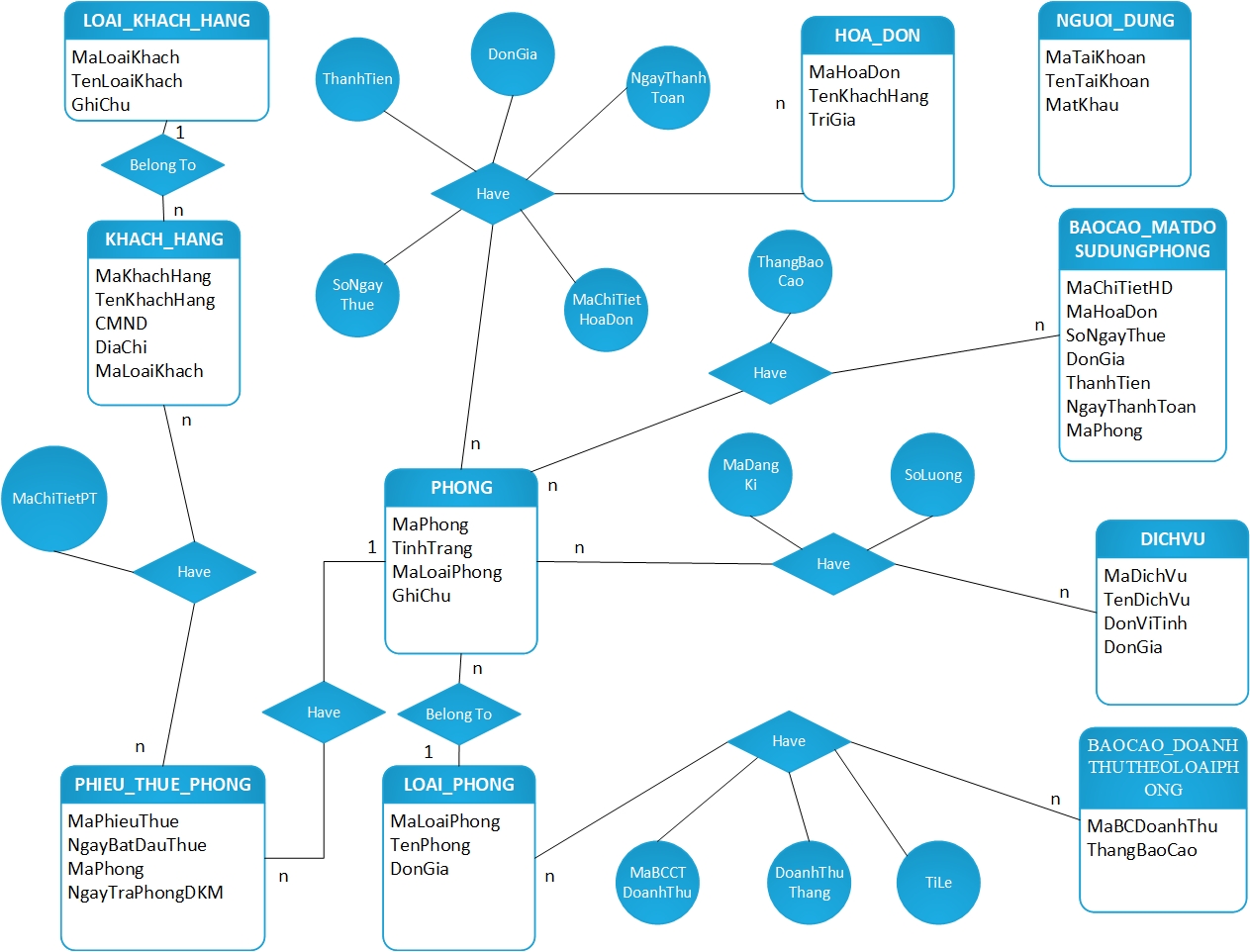
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaBCCTDoanhThu | int | Primary key | Turnover detail report code |
| 2 | MaLoaiPhong | Nvarchar(10) | Primary key | Room type code |
| 3 | MaBCDoanhThu | int | Primary key | Turnover report code |
| 4 | DoanhThuThang | money |  | Monthly turnover |
| 5 | Tile | tile |  |  |

15.NGUOI\_DUNG

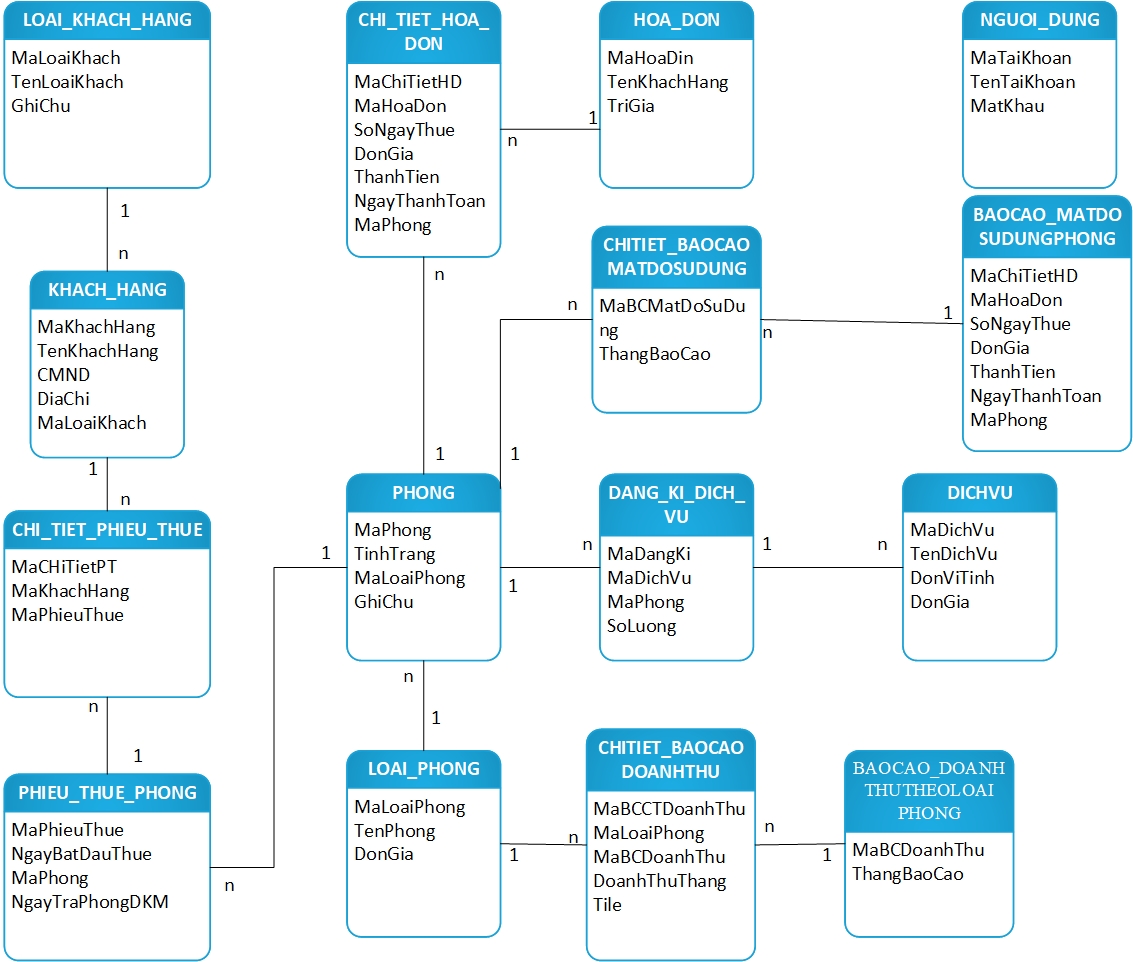
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| **1** | **MaTaiKhoan** | **int** | **Primary key** | **Account code** |
| 2 | TenTaiKhoan | char(10) |  | Username |
| 3 | Matkhau | Char(20) |  | Password |

### 6.3 Database diagrams

6.3.1 Entity – Relationship Model

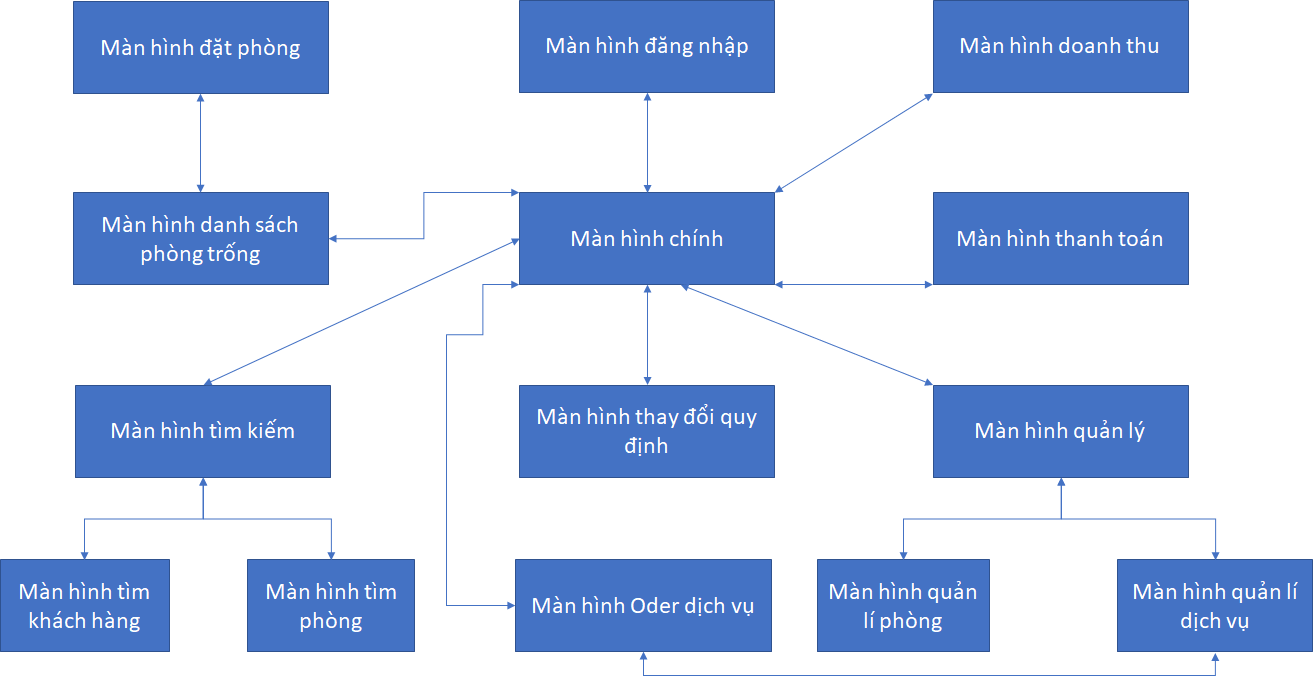


6.3.2 Relationship Model



**7. User interface design**

**7.1 Relational diagram of screens in interface**

****

**7.2 Screen list and function description**

|  |  |  |
| --- | --- | --- |
| **Id** | **Name** | **Function description** |
| 1 | Login screen | Login to application |
| 2 | Main screen | Display login information and go to next possible screens |
| 3 | Room management screen | Manage room info, allow add, edit and delete room info |
| 4 | Service management screen | Manage service info, allow add, edit and delete service info |
| 5 | Empty room list screen | Show empty room list (For booking purposes) |
| 6 | Booking room screen | Show booking form, allow to create a new one |
| 7 | Check out screen | Show invoice, allow to create and print |
| 8 | Search room screen | For finding room purpose |
| 9 | Search customer screen | Find info of customers that stay in hotel |
| 10 | Order service screen | Order services for currently used room |
| 11 | Turnover screen | Show turnover list of rooms by corresponding datetime period and room |
| 12 | Change regulation screen | Change parameter in current regulations |

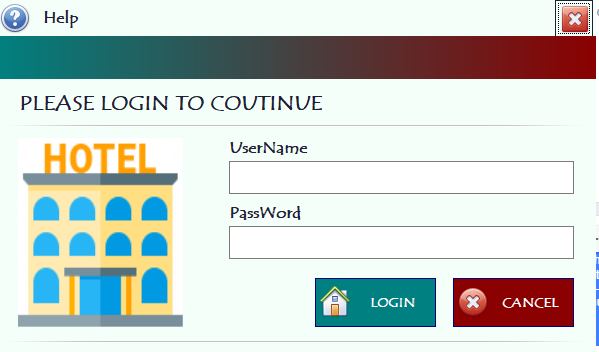
**7.3. User interface and details**

**7.3.1 Naming convention of form components**

|  |  |
| --- | --- |
| ***Tiền tố (prefix)*** | ***Loại Control*** |
| lbl | Label |
| llbl | LinkLabel |
| btn | Button |
| txt | Textbox |
| mnu | MainMenu |
| chk | CheckBox |
| rdo | RadioButton |
| grp | GroupBox |
| pic | PictureBox |
| grd | Grid |
| lst | ListBox |
| cbo | ComboBox |
| lstv | ListView |
| tre | TreeView |
| tab | TabControl |
| dtm | DateTimePicker |
| mon | MonthCalendar |
| sbr | ScrollBar |
| tmr | Timer |
| spl | Splitter |
| dud | DomainUpDown |
| nud | NumericUpDown |
| trk | TrackBar |
| pro | ProgressBar |
| rtxt | RichTextBox |
| img | ImageList |
| hlp | HelpProvider |
| tip | ToolTip |
| cmnu | ContextMenu |
| tbr | ToolBar |
| frm | Form |
| bar | StatusBar |
| nico | NotifyIcon |
| ofd | OpenFileDialog |
| sdl | SaveFileDialog |
| fd | FontDialog |
| cd | ColorDialog |
| pd | PrintDialog |
| ppd | PrintPreviewDialog |
| ppc | PrintPreviewControl |
| err | ErrorProvider |
| pdoc | PrintDocument |
| psd | PageSetupDialog |
| crv | CrystalReportDialog |
| pd | PrintDialog |
| sw | FileSystemWatcher |
| log | EventLog |
| dire | DirectoryEntry |
| dirs | DirectorySearcher |
| msq | MessageQueue |
| pco | PerformanceCounter |
| pro | Process |
| ser | ServiceController |
| rpt | ReportDocument |
| ds | DataSet |
| olea | OleDbDataAdapter |
| olec | OleDbConnection |
| oled | OleDbCommand |
| sqla | SqlDbDataAdapter |
| sqlc | SqlDbConnection |
| sqld | SqlDbCommand |
| dvw | DataView |

**7.3.2 Interfaces and details**

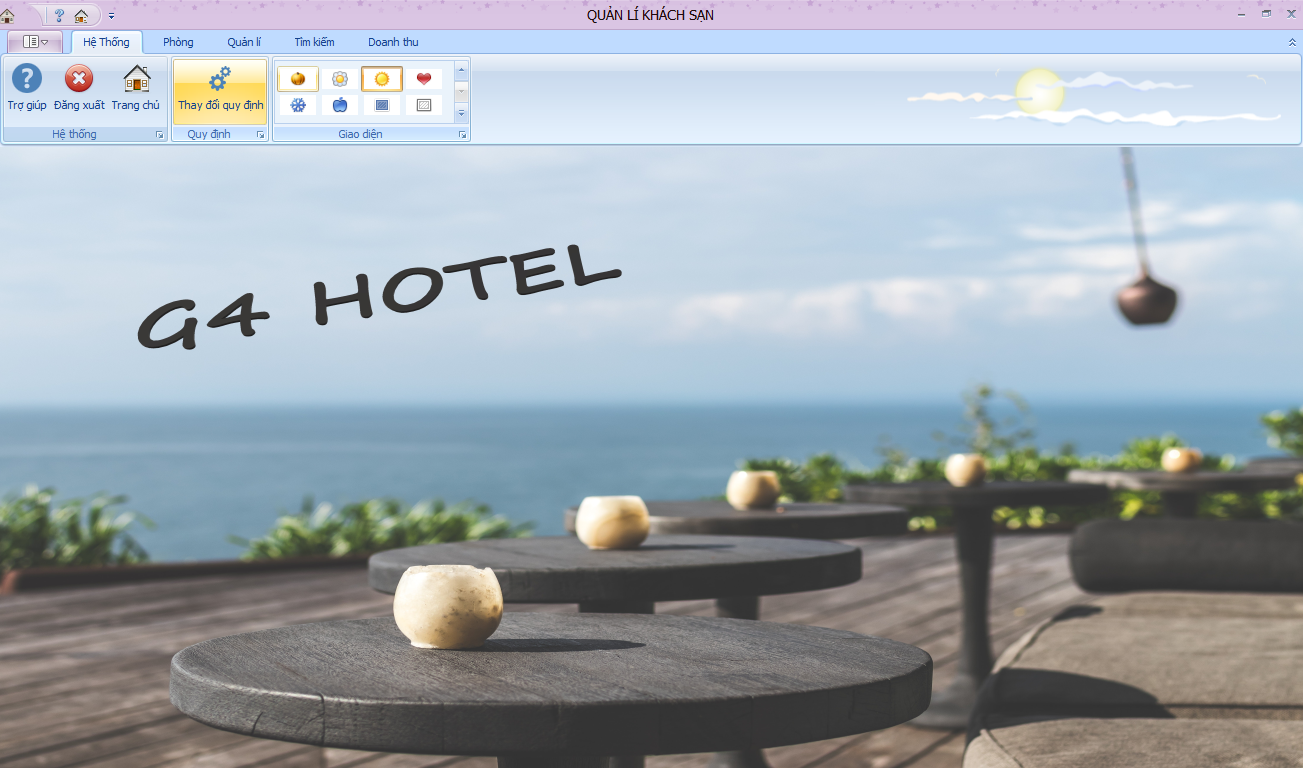
* **Login screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbTenDangNhap | Label | Username |
| 2 | lbMatKhau | Label | Password |
| 3 | lbTroGiup | Label | Help |
| 4 | lbLogin | Label | Login |
| 5 | txtTenDangNhap | TextBox | TextBox enter username |
| 6 | txtMatKhau | TextBox | TextBox enter password |
| 7 | btnDangNhap | Button | Button login |
| 8 | btnHuy | Button | Button cancel |
| 9 | btnTroGiup | Button | Button help |
| 10 | pnHotel | PictureBox | PictureBox Hotel |

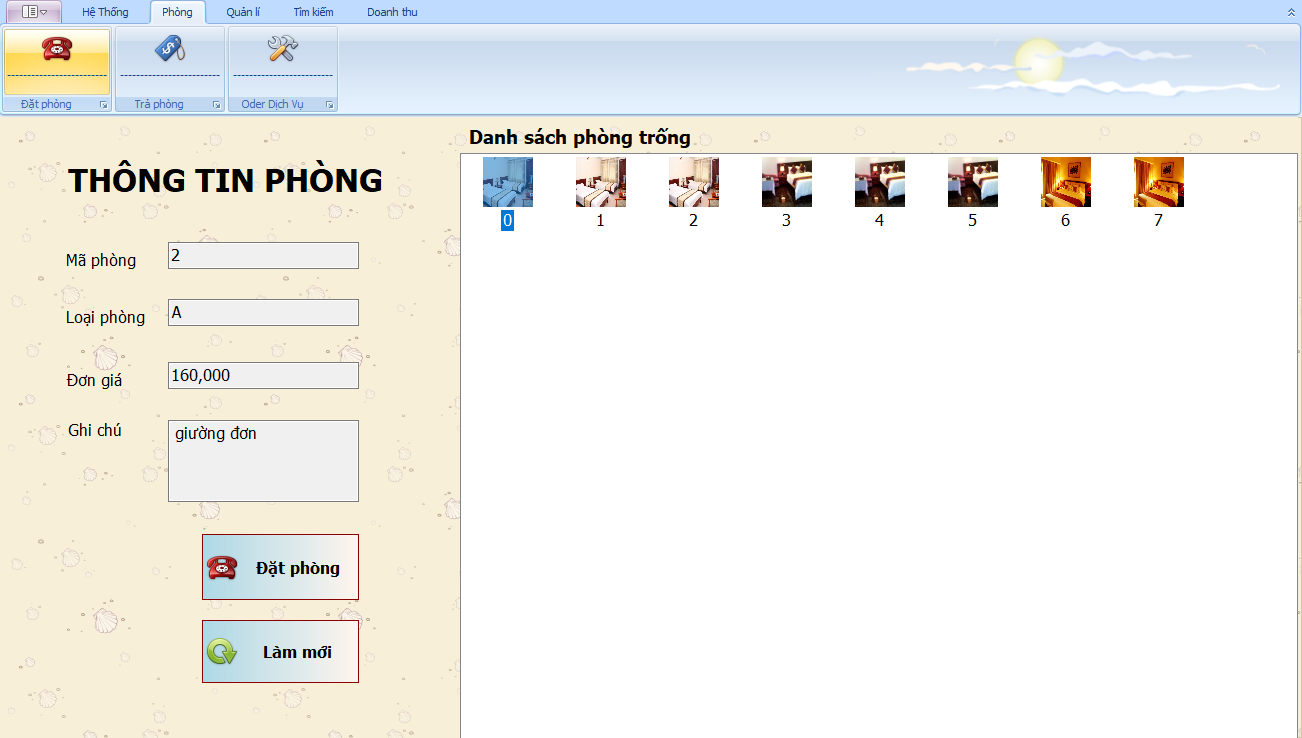
* **Main screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | rbHeThong | Ribbon page | Ribbon system |
| 2 | rbPhong | Ribbon page | Ribbon room |
| 3 | rbQuanLy | Ribbon page | Ribbon manage |
| 4 | rbTimKiem | Ribbon page | Ribbon search |
| 5 | rbDoanhThu | Ribbon page | Ribbon turnover |
| 6 | rbgTroGiup | Ribbon page group | Ribbon group helper |
| 7 | rbgDangXuat | Ribbon page group | Ribbon group logout |
| 8 | rbgTrangChu | Ribbon page group | Ribbon group main screen |
| 9 | rbgQuyDinh | Ribbon page group | Ribbon regulation group |
| 10 | rbgSkin | Ribbon page group | Button skin UI |
| 11 | pnHienThi | Panel | User control list panel |

* **Empty room list screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbMaPhong | Label | Room id |
| 2 | lbLoaiPhong | Label | Room type |
| 3 | lbThongTin | Label | Room info |
| 4 | lbDonGia | Label | Price |
| 5 | txtMaPhong | TextBox | Enter room code |
| 6 | txtLoaiPhong | TextBox | Enter room type |
| 7 | txtThongTin | TextBox | Enter room info |
| 8 | txtDonGia | TextBox | Enter price |
| 9 | txtGhiChu | TextBox | Enter note |
| 10 | BtnDat | Button | Book room |
| 11 | btnLamMoi | Button | Refresh room list |
| 12 | lvDanhSachPhong | Listview | Room photo list |

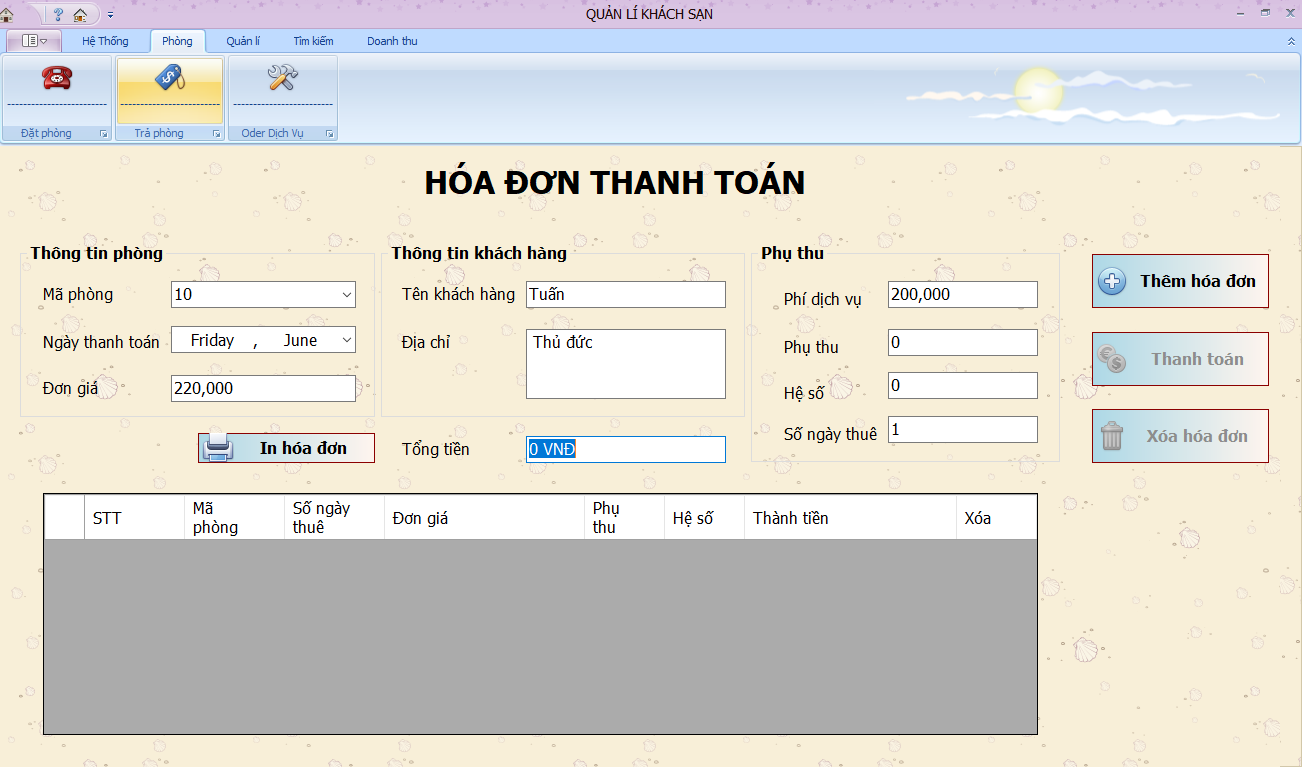
* **Book room screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | GrPhieuThue | GroupBox | Contains booking form info |
| 2 | lbMaPhong | Label | Label room id |
| 3 | lbLoaiPhong | Label | Label room type |
| 4 | lbNgayThue | Label | Label starting date |
| 5 | grKhachThue | GroupBox | Contains customer info |
| 6 | lbTenKhachHang | Label | Label customer name |
| 7 | lbLoaiKhachHang | Label | Label customer type |
| 8 | lbCMND | Label | Label identity card number |
| 9 | lbDiaChi | Label | Label address |
| 10 | btnThem | Button | Button Add |
| 11 | btnXoa | Button | Button Delete |
| 12 | btnLuu | Button | Button Save |
| 13 | dgvPhieuThue | DataGridView | Customer list |
| 14 | txtMaPhong | TextBox | Enter room id |
| 15 | txtLoaiPhong | TextBox | Enter room type |
| 16 | txtTenKhachHang | TextBox | Enter customer name |
| 17 | txtCMND | TextBox | Enter identity card number |
| 18 | txtDiaChi | TextBox | Enter address |
| 19 | cmbLoaiKhachHang | ComboBox | ComboBox Customer type list |
| 20 | dtpNgayThue | DateTimePicker | DateTimePicker pick starting date |
| 21 | dtpNgayTra | DateTimePicker | DateTimePicker pick checkout date (expecting) |

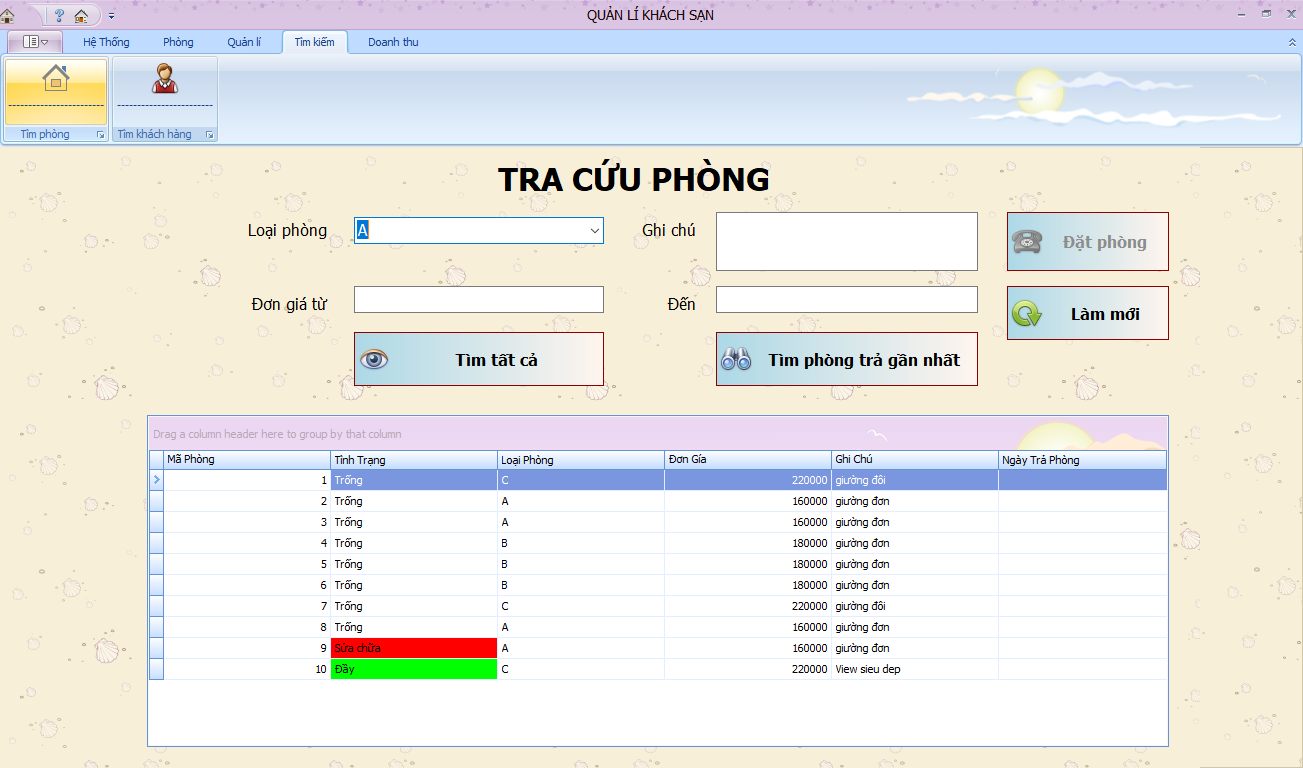
* **Checkout screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbHoaDon | Label | Invoice |
| 2 | lbMaPhong | Label | Room id |
| 3 | lbTongTien | Label | Total |
| 4 | lbDonGia | Label | Price |
| 5 | lbTenKhachHang | Label | Customer name |
| 6 | lbDiaChi | Label | Address |
| 7 | lbPhiDichVu | Label | Service fee |
| 8 | lbPhuThu | Label | Extra fee |
| 9 | lbHeSo | Label | Coefficient |
| 10 | lbSoNgayThue | Label | Stayed days |
| 11 | cmbMaPhong | ComboBox | Contains list of room id |
| 12 | txtTongTien | TextBox | Display total amount |
| 13 | txtDonGiaPhong | TextBox | Display room price |
| 14 | txtTenKhachHang | TextBox | Display customer name |
| 15 | txtDiaChi | TextBox | Display customer addr |
| 16 | txtPhiDichVu | TextBox | Display service fee |
| 17 | txtPhuThu | TextBox | Display extra fee |
| 18 | txtHeSo | TextBox | Display coefficient |
| 19 | txtSoNgayThue | TextBox | Display stayed day |
| 20 | btnThemHoaDon | Button | Button Add invoice |
| 21 | btnXoaHoaDon | Button | Buttuon Delete invoice |
| 22 | btnThanhToan | Button | Button Checkout |

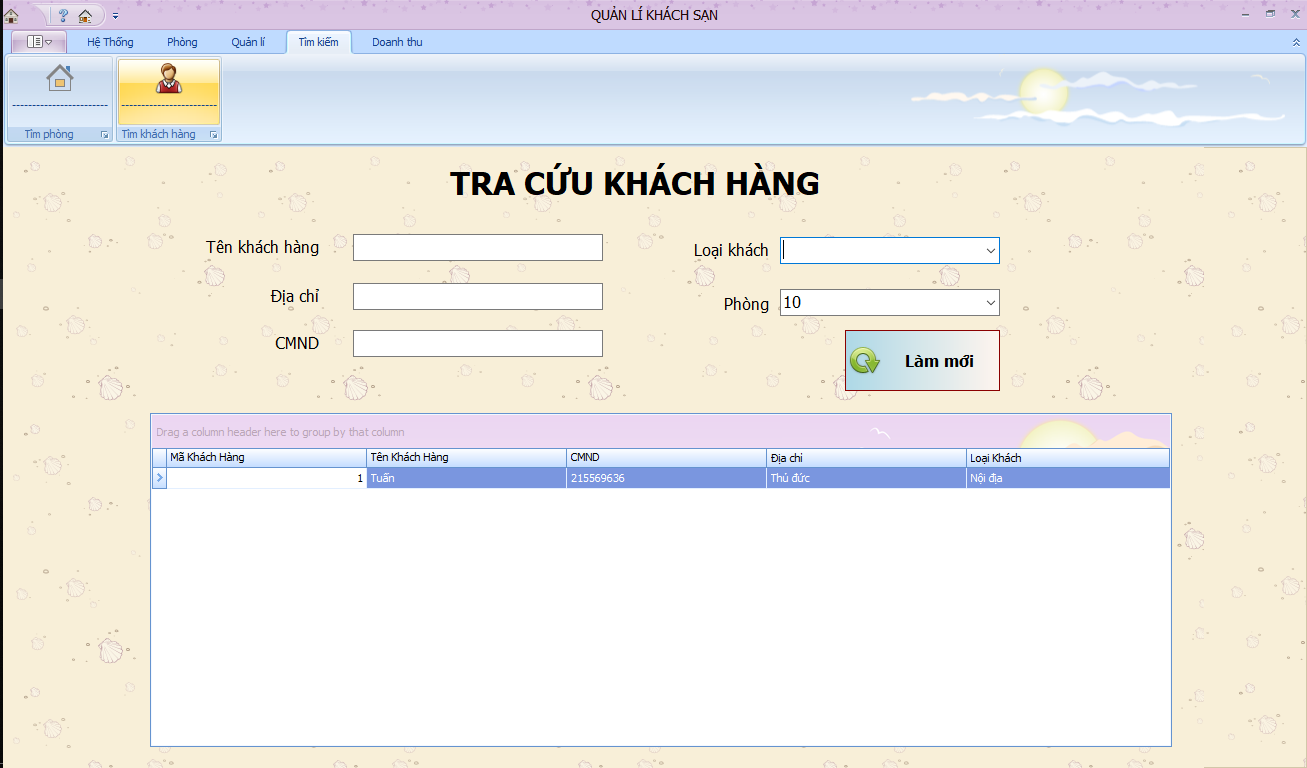
* **Find room screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbLoaiPhong | Label | Room id |
| 2 | lbDonGiaTu | Label | Price start from... |
| 3 | lbDonGiaDen | Label | To... |
| 4 | lbGhiChu | Label | Note |
| 5 | txtDonGiaDen | TextBox | Maximum price |
| 6 | txtDonGiaTu | TextBox | Minimum price |
| 7 | cmbLoaiPhong | ComboBox | Choose room type |
| 8 | txtGhiChu | TextBox | Enter note |
| 9 | btnDatPhong | Button | Button Book room |
| 10 | btnLamMoi | Button | Button Refresh |
| 11 | btnTimPhong | Button | Button Find room |
| 12 | btnTimPhongGN | Button | Button Find most recent checked out room |
| 11 | dgvSearchRoom | DataGridView | Datagridview contain room info |

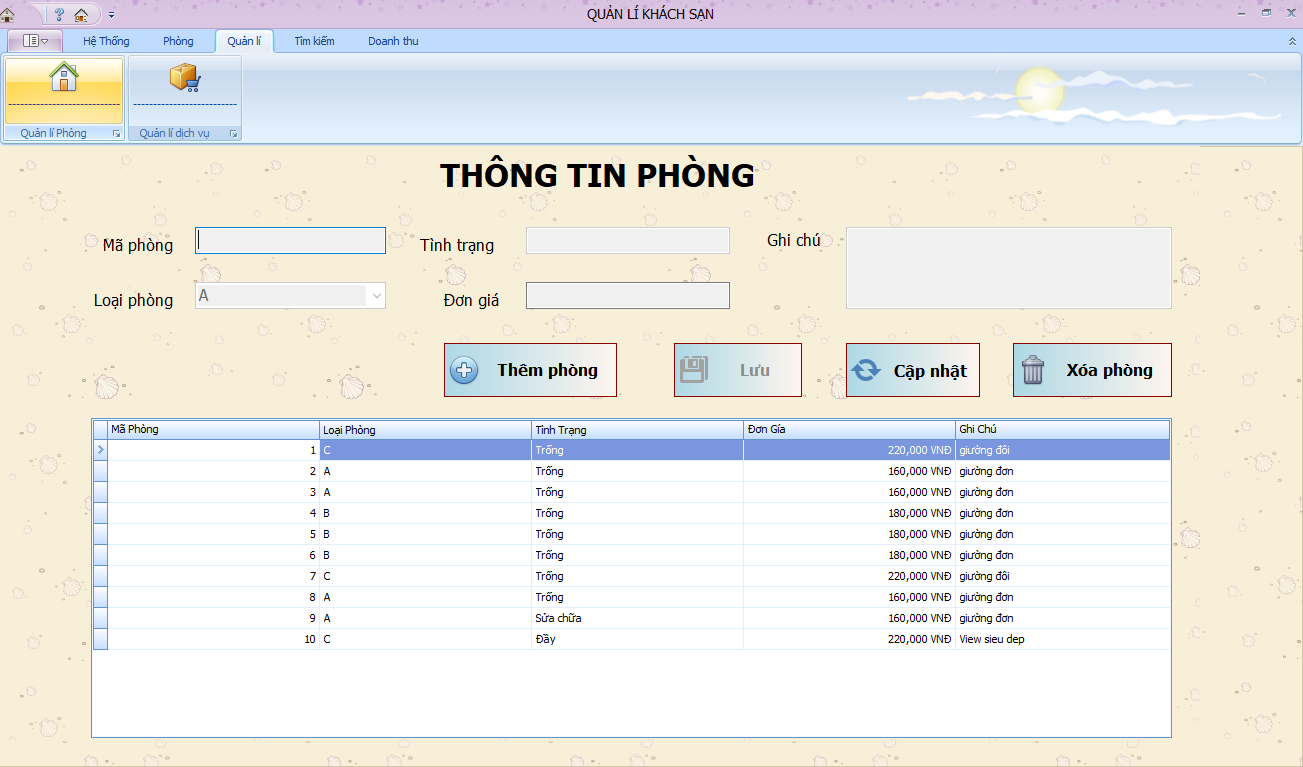
* **Find customer screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbTenKhachHang | label | Customer name |
| 2 | lbDiaChi | label | Customer address |
| 4 | lbCMND | label | Identity card number |
| 5 | lbLoaiKhach | Label | Customer address |
| 6 | lbMaPhong | Label | Room id |
| 7 | txtTenKhachHang | TextBox | Textbox Enter customer name |
| 8 | txtDiaChi | TextBox | Enter Customer address |
| 9 | txtCMND | TextBox | Enter Identity card number |
| 10 | cmbLoaiKhach | ComboBox | Combobox Customer address |
| 11 | cmbPhong | ComboBox | Combobox room list |
| 12 | btnLamMoi | Button | Button Refresh |
| 13 | dgvDanhSachKhachHang | DataGridView | Datagrid room list |

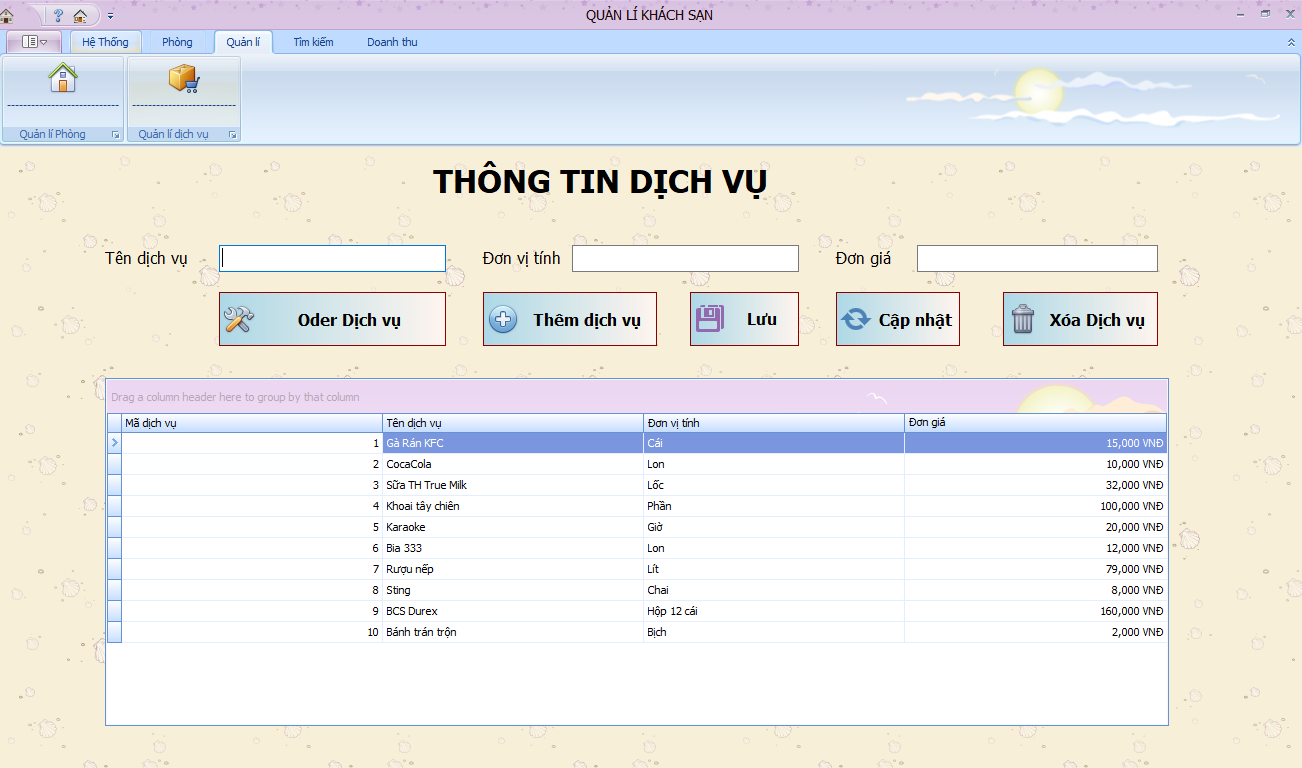
* **Room management service**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbMaPhong | Label | Room id |
| 2 | lbLoaiPhong | Label | Room type |
| 3 | lbDonGia | Label | Price |
| 4 | lbGhiChu | Label | Note |
| 5 | lbTinhTrang | Label | Room status |
| 6 | txtMaPhong | TextBox | Enter room id |
| 7 | txtTinhTrang | TextBox | Enter Room status |
| 8 | cmbLoaiPhong | ComboBox | Room type list |
| 9 | txtDonGia | TextBox | Enter price |
| 10 | txtGhiChu | TextBox | Enter note |
| 11 | btnThem | Button | Button Add room |
| 12 | btnCapNhat | Button | Button Update room |
| 13 | btnXoa | Button | Button Delete room |
| 14 | btnLuu | Button | Button Save room |
| 15 | dgvDanhSachPhong | DataGridView | Datagrid room list |

* **Service management screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbTenDichVu | label | Service name |
| 2 | lbDonViTinh | label | Unit |
| 3 | lbDonGia | label | Price |
| 4 | txtTenDichVu | TextBox | Enter Service name |
| 5 | txtDonViTinh | TextBox | Enter unit |
| 6 | txtDonGia | TextBox | Enter price |
| 7 | btnXoa | Button | Button Delete service |
| 8 | btnSua | Button | Button Edit service |
| 9 | btnThem | Button | Button Add service |
| 10 | btnOder | Button | To Order screen |
| 11 | btnLuu | Button | Save changes of service |
| 12 | dgvDanhSachDichVu | DataGridView | Datagrid service list |

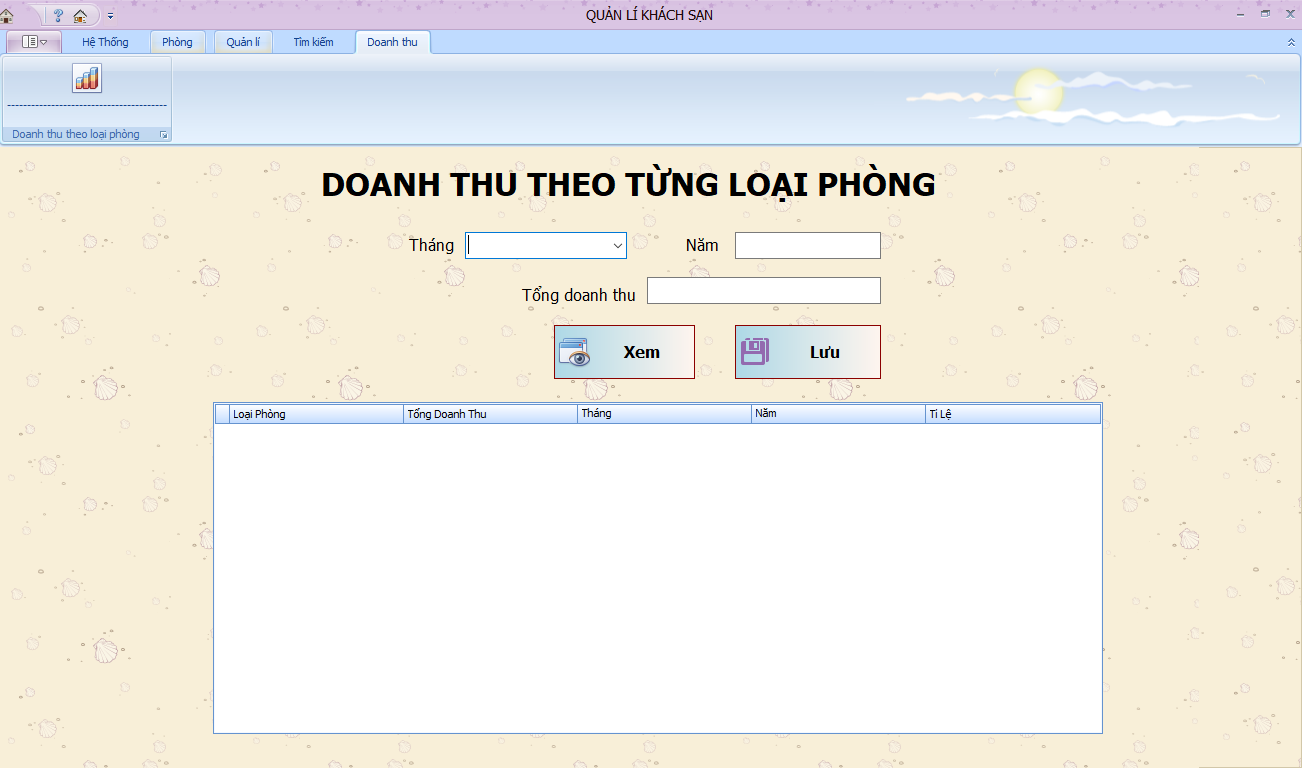
* **Order service screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbOder | Label | Order service |
| 2 | lbMaPhong | Label | Room id |
| 3 | lbTenDichVu | Label | Service name |
| 4 | lbSoLuong | Label | Number of |
| 5 | cmbMaPhong | ComboBox | Room id list |
| 6 | cmbDichVu | ComboBox | Service list |
| 7 | txtSoLuong | TextBox | Enter number of services needed |
| 8 | btnOder | Button | Order that service |
| 9 | btnQuayLai | Button | Back to service management screen |

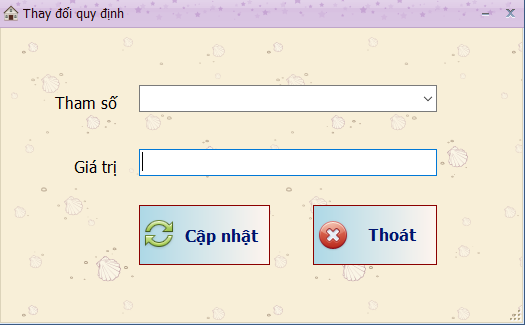
* **Turnover report screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbTongDoanhThu | Label | Total turnover |
| 2 | lbThang | Label | Month |
| 3 | lbNam | Label | Year |
| 4 | txtNam | TextBox | Enter year |
| 5 | txtThang | TextBox | Enter month |
| 6 | txtTongDoanhThu | TextBox | Display total turnover |
| 7 | btnXem | Button | Show turnover of room |
| 8 | btnLuu | Button | Save |
| 9 | dvgDoanhThu | Datagridview | Contains turnover info |

* **Change regulation screen**



* ***Design details:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Type** | **Meaning** |
| 1 | lbThamSo | Label | Parameter |
| 2 | lbGiaTri | Label | Value |
| 3 | cmbThamSo | ComBoBox | Parameter list |
| 4 | txtGiaTri | TextBox | Enter new value |
| 5 | btnCapNhat | Button | Update change |
| 6 | btnThoát | Button | Exit |

1. **Implementation**

**8.1 Technology**

* ***Language C#:***
* C# is a general-purpose, multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. C# is one of the programming languages designed for the Common Language Infrastructure, and saved in executive file (.exe or .dll).
* C#'s development team is led by Mads Torgersen. The most recent version is C# 7.3, which was released in 2018 alongside Visual Studio 2017 version 15.7.2
* ***EntityFramework***
* **Entity framwork** is a platform for developing application.
* Entity Framework is a O/RM in .NET framwork. One of the best in the aspect of data accessing in .NET
* Entity Framework is an accessing database technique that help programmer to get and manipulate data easily via data objects likes DataReader, DataSet
* This technology allow you to work with Relational database.
* ***DevExpress***
* DevExpress is Framework that was designed for .NET Framework. It provides controls and ultilities for developing software

**8.2** **Vấn đề khi cài đặt**

* ConnectionString errors are quite common error, that cause the program cant connect to the database
* Install some function sometimes get user into troubles
* UI are not really friendly

**9. Test documents**

**9.1 Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | UC ID | Condition to be tested | Test data | Expected result | Successful |
|  | UC\_login | Log in | True username and true password | Log in success |  |
|  | UC\_login | Log in | Wrong username or wrong password | Log in failed |  |
|  | UC\_reservation | Reservation | Room is booked or being repaired | Book room unsuccessfully |  |
|  | UC\_reservation | Reservation | Empty room | Book room successfully |  |
|  | UC\_checkout | Checkout | Exactly the room used | Check-out successfully |  |
|  | UC\_checkout | Checkout | Not the room used | Check-out unsuccessfully |  |
|  | UC\_order service | Order service | Service in inventory meets requirements | Order service successfully |  |
|  | UC\_order service | Order service | Service in inventory does not meet requirements | Order service unsuccessfully |  |
|  | UC\_addroom | Add room | Room not exist | Add room successfully |  |
|  | UC\_addroom | Add room | Room existed | Add room unsuccessfully |  |
|  | UC\_delete room | Delete room | Room exist | Delete room successfully |  |
|  | UC\_delete room | Delete  room | Room is using | Delete room fail |  |
|  | UC\_edit room | Edit room | Edited room code is not duplicated in database | Edit room successfully |  |
|  | UC\_edit room | Edit room | Edited room code is duplicated in database | Edit room unsuccessfully |  |
|  | UC\_add service | Add service | Service not exist | Add service successfully |  |
|  | UC\_add service | Add service | Service existed | Can’t add service |  |
|  | UC\_delete service | Delete service | Service exist | Service deleted |  |
|  | UC\_edit service | Edit service | Service id is not duplicated | Edit service successfully |  |
|  | UC\_edit service | Edit service | Service id is duplicated | Edit service unsuccessfully |  |
|  | UC\_find room | Find room | Has that room type | Room found |  |
|  | UC\_find room | Find room | Does not have that room type | Room not found |  |
|  | UC\_create new account | Create new account | Username or password not exist | Account created |  |
|  | UC\_create new account | Create new account | Username or password existed | Create account unsuccessfully |  |
|  | UC\_change password | Change password | Password follows the rules | Password changed |  |
|  | UC\_change password | Change password | Password not follows the rules | Change password unsuccessfully |  |
|  | UC\_change regulation | Change regulation | Password follows the rules | Regulation changed |  |
|  | UC\_turnover statistics by room type | Turnover statistics by room type | Input time matches the room lending time | Statistic |  |
|  | UC\_turnover statistics by room type | Turnover statistics by room type | Input time matches the room using time | Turnover statistics unsuccessfully |  |

**9.2 Test Scenario**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | Requirement Name | Test scenario | | Test Case | |
|  |  | TS Name | TS ID | TC Name | TC ID |
| 1 | Login | OK | TS\_login\_01 | a. Verify OK with valid data | TS\_login\_01\_01 |
| b. Verify OK with invalid data | TS\_login\_01\_02 |
| Cancel | TS\_login\_02 | a. Verify Cancel with valid data | TS\_login\_02\_01 |
| b. Verify Cancel with invalid data | TS\_login\_02\_02 |
| 2 | Reservation | OK | TS\_Reservation\_01 | a. Verify OK with valid data | TS\_Reservation 01\_01 |
| b. Verify OK with invalid data | TS\_Reservation\_01\_02 |
| Cancel | TS\_Reservation\_02 | a. Verify Cancel with valid data | TS\_Reservation\_02\_01 |
| b. Verify Cancel with invalid data | TS\_Reservation\_02\_02 |
| 3 | Checkout | OK | TS\_checkout\_01 | a. Verify OK with valid data | TS\_checkout\_01\_01 |
| b. Verify OK with invalid data | TS\_checkout\_01\_02 |
| Cancel | TS\_checkout\_02 | a. Verify Cancel with valid data | TS\_checkout\_02\_01 |
| b. Verify Cancel with invalid data | TS\_checkout\_02\_02 |
| 4 | Order service | OK | TS\_orderservice\_01 | a. Verify OK with valid data | TS\_orderservice\_01\_01 |
| b. Verify OK with invalid data | TS\_orderservice\_01\_02 |
| Cancel | TS\_orderservice\_02 | a. Verify Cancel with valid data | TS\_orderservice\_02\_01 |
| b. Verify Cancel with invalid data | TS\_orderservice\_02\_02 |
| 5 | Add room | OK | TS\_addroom\_01 | a. Verify OK with valid data | TS\_addroom\_01\_01 |
| b. Verify OK with invalid data | TS\_addroom\_01\_02 |
| Cancel | TS\_addroom\_02 | a. Verify Cancel with valid data | TS\_addroom\_02\_01 |
| b. Verify Cancel with invalid data | TS\_addroom\_02\_02 |
| 6 | Edit user | OK | TS\_editroom\_01 | a. Verify OK with valid data | TS\_editroom\_01\_01 |
| b. Verify OK with invalid data | TS\_editroom\_01\_02 |
| Cancel | TS\_editroom\_02 | a. Verify Cancel with valid data | TS\_editroom\_02\_01 |
| b. Verify Cancel with invalid data | TS\_editroom\_02\_02 |
| 7 | Delete room | OK | TS\_deleteroom\_01 | a. Verify OK with valid data | TS\_deleteroom\_01\_01 |
| b. Verify OK with invalid data | TS\_deleteroom\_01\_02 |
| Cancel | TS\_deleteroom\_02 | a. Verify Cancel with valid data | TS\_deleteroom\_02\_01 |
| b. Verify Cancel with invalid data | TS\_deleteroom\_02\_02 |
| 8 | Find room | OK | TS\_findroom\_01 | a. Verify OK with valid data | TS\_findroom\_01\_01 |
| b. Verify OK with invalid data | TS\_findroom\_01\_02 |
| Cancel | TS\_findroom\_02 | a. Verify Cancel with valid data | TS\_findroom\_02\_01 |
| b. Verify Cancel with invalid data | TS\_findroom\_02\_02 |
| 9 | Add service | OK | TS\_addservice\_01 | a. Verify OK with valid data | TS\_addservice\_01\_01 |
| b. Verify OK with invalid data | TS\_addservice\_01\_02 |
| Cancel | TS\_addservice\_02 | a. Verify Cancel with valid data | TS\_addservice\_02\_01 |
| b. Verify Cancel with invalid data | TS\_addservice\_02\_02 |
| 10 | Edit service | OK | TS\_editservice\_01 | a. Verify OK with valid data | TS\_editservice\_01\_01 |
| b. Verify OK with invalid data | TS\_editservice\_01\_02 |
| Cancel | TS\_editservice\_02 | a. Verify Cancel with valid data | TS\_editservice\_02\_01 |
| b. Verify Cancel with invalid data | TS\_editservice\_02\_02 |
| 11 | Delete service | OK | TS\_deleteservice\_01 | a. Verify OK with valid data | TS\_deleteservice\_01\_01 |
| b. Verify OK with invalid data | TS\_deleteservice\_01\_02 |
| Cancel | TS\_deleteservice\_02 | a. Verify Cancel with valid data | TS\_deleteservice\_02\_01 |
| b. Verify Cancel with invalid data | TS\_deleteservice\_02\_02 |
| 12 | Create new account | OK | TS\_createnewacaccount\_01 | a. Verify OK with valid data | TS\_createnewacaccount\_01\_01 |
| b. Verify OK with invalid data | TS\_createnewacaccount\_01\_02 |
| Cancel | TS\_createnewacaccount\_02 | a. Verify Cancel with valid data | TS\_createnewacaccount\_02\_01 |
| b. Verify Cancel with invalid data | TS\_createnewacaccount\_02\_02 |
| 13 | Change password | OK | TS\_changepassword\_01 | a. Verify OK with valid data | TS\_changepassword\_01\_01 |
| b. Verify OK with invalid data | TS\_changepassword\_01\_02 |
| Cancel | TS\_changepassword\_02 | a. Verify Cancel with valid data | TS\_changepassword\_02\_01 |
| b. Verify Cancel with invalid data | TS\_changepassword\_02\_02 |
| 14 | Change regulation | OK | TS\_changeregulation\_01 | a. Verify OK with valid data | TS\_changeregulation\_01\_01 |
| b. Verify OK with invalid data | TS\_changeregulation\_01\_02 |
| Cancel | TS\_changeregulation\_02 | a. Verify Cancel with valid data | TS\_changeregulation\_02\_01 |
| b. Verify Cancel with invalid data | TS\_changeregulation\_02\_02 |
| 15 | Turnover statistics by room type | OK | TS\_turnoverstatisticsbyroomtype\_01 | a. Verify OK with valid data | TS\_turnoverstatisticsbyroomtype\_01\_01 |
| b. Verify OK with invalid data | TS\_turnoverstatisticsbyroomtype\_01\_02 |
| Cancel | TS\_turnoverstatisticsbyroomtype\_02 | a. Verify Cancel with valid data | TS\_turnoverstatisticsbyroomtype\_02\_01 |
| b. Verify Cancel with invalid data | TS\_turnoverstatisticsbyroomtype\_02\_02 |

**9.3 Requirement Traceability Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | UC id | TS id | TC id | Status | Defect |
| 1 | UC\_login | TS\_login\_01 | TS\_login\_01\_01 |  |  |
| 2 | TS\_login\_01\_02 |  |  |
| 3 | TS\_login\_02 | TS\_login\_02\_01 |  |  |
| 4 | TS\_login\_02\_02 |  |  |
| 5 | UC\_reservation | TS\_Reservation\_01 | TS\_Reservation 01\_01 |  |  |
| 6 | TS\_Reservation\_01\_02 |  |  |
| 7 | TS\_Reservation\_02 | TS\_Reservation\_02\_01 |  |  |
| 8 | TS\_Reservation\_02\_02 |  |  |
| 9 | UC\_checkout | TS\_checkout\_01 | TS\_checkout\_01\_01 |  |  |
| 10 | TS\_checkout\_01\_02 |  |  |
| 11 | TS\_checkout\_02 | TS\_checkout\_02\_01 |  |  |
| 12 | TS\_checkout\_02\_02 |  |  |
| 13 | UC\_odersevice | TS\_orderservice\_01 | TS\_orderservice\_01\_01 |  |  |
| 14 | TS\_orderservice\_01\_02 |  |  |
| 15 | TS\_orderservice\_02 | TS\_orderservice\_02\_01 |  |  |
| 16 | TS\_orderservice\_02\_02 |  |  |
| 17 | UC\_addroom | TS\_addroom\_01 | TS\_addroom\_01\_01 |  |  |
| 18 | TS\_addroom\_01\_02 |  |  |
| 19 | TS\_addroom\_02 | TS\_addroom\_02\_01 |  |  |
| 20 | TS\_addroom\_02\_02 |  |  |
| 21 | UC\_editroom | TS\_editroom\_01 | TS\_editroom\_01\_01 |  |  |
| 22 | TS\_editroom\_01\_02 |  |  |
| 23 | TS\_editroom\_02 | TS\_editroom\_02\_01 |  |  |
| 24 | TS\_editroom\_02\_02 |  |  |
| 25 | UC\_deleteroom | TS\_deleteroom\_01 | TS\_deleteroom\_01\_01 |  |  |
| 26 | TS\_deleteroom\_01\_02 |  |  |
| 27 | TS\_deleteroom\_02 | TS\_deleteroom\_02\_01 |  |  |
| 1 | TS\_deleteroom\_02\_02 |  |  |
| 2 | UC\_finroom | TS\_findroom\_01 | TS\_findroom\_01\_01 |  |  |
| 3 | TS\_findroom\_01\_02 |  |  |
| 4 | TS\_findroom\_02 | TS\_findroom\_02\_01 |  |  |
| 5 | TS\_findroom\_02\_02 |  |  |
| 6 | UC\_addservice | TS\_addservice\_01 | TS\_addservice\_01\_01 |  |  |
| 7 | TS\_addservice\_01\_02 |  |  |
| 8 | TS\_addservice\_02 | TS\_addservice\_02\_01 |  |  |
| 9 | TS\_addservice\_02\_02 |  |  |
| 10 | UC\_editservice | TS\_editservice\_01 | TS\_editservice\_01\_01 |  |  |
| 11 | TS\_editservice\_01\_02 |  |  |
| 12 | TS\_editservice\_02 | TS\_editservice\_02\_01 |  |  |
| 13 | TS\_editservice\_02\_02 |  |  |
| 14 | UC\_deleteservice | TS\_deleteservice\_01 | TS\_deleteservice\_01\_01 |  |  |
| 15 | TS\_deleteservice\_01\_02 |  |  |
| 16 | TS\_deleteservice\_02 | TS\_deleteservice\_02\_01 |  |  |
| 17 | TS\_deleteservice\_02\_02 |  |  |
| 18 | UC\_createnewaccount | TS\_createnewacaccount\_01 | TS\_createnewacaccount\_01\_01 |  |  |
| 19 | TS\_createnewacaccount\_01\_02 |  |  |
| 20 | TS\_createnewacaccount\_02 | TS\_createnewacaccount\_02\_01 |  |  |
| 21 | TS\_createnewacaccount\_02\_02 |  |  |
| 22 | UC\_changepassword | TS\_changepassword\_01 | TS\_changepassword\_01\_01 |  |  |
| 23 | TS\_changepassword\_01\_02 |  |  |
| 24 | TS\_changepassword\_02 | TS\_changepassword\_02\_01 |  |  |
| 25 | TS\_changepassword\_02\_02 |  |  |
| 26 | UC\_changeregulation | TS\_changeregulation\_01 | TS\_changeregulation\_01\_01 |  |  |
| 27 | TS\_changeregulation\_01\_02 |  |  |
| 28 | TS\_changeregulation\_02 | TS\_changeregulation\_02\_01 |  |  |
| 29 | TS\_changeregulation\_02\_02 |  |  |
| 31 | UC\_turnoverstaticticsbyroom | TS\_turnoverstatisticsbyroomtype\_01 | TS\_turnoverstatisticsbyroomtype\_01\_01 |  |  |
| 32 | TS\_turnoverstatisticsbyroomtype\_01\_02 |  |  |
| 33 | TS\_turnoverstatisticsbyroomtype\_02 | TS\_turnoverstatisticsbyroomtype\_02\_01 |  |  |
| 34 | TS\_turnoverstatisticsbyroomtype\_02\_02 |  |  |

**10.Tổng kết**

Phần mềm bước đầu được hoàn thành hoàn chỉnh đúng theo kế hoạch nhóm đề ra. Tuy gặp nhiều khó khăn nhưng cũng đã giải quyết tương đối ổn thõa.

* ***Bảng phân công nhiệm vụ***

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Chức vụ** | **Nhiệm vụ** |
| Trần Nam Khánh | Nhóm trưởng | * Tìm hiểu về devexpress và thiết kế giao diện * Tìm hiểu và thuyết trình trong buổi seminar về mô hình thác nước * Phân công nhiệm vụ cho các thành viên còn lại * Tìm hiểu và giải quyết vấn đề khi nhóm gặp phải khó khăn khi coding * Viết báo cáo đồ án |
| Đậu Thị Kim Dung | Thành viên | * Thiết kế database * Lập trình các chức năng chính của phần mềm * Hỗ trợ Seminar mô hình thác nước |
| Lê Trung Hiếu | Thành viên | * Lập trình một số chức năng của phần mềm * Hỗ trợ thiết kế database * Kiểm tra và rà sát lỗi phần mềm |
| Đậu Thị Thanh Huyền | Thành viên | * Hỗ trợ viết báo cáo * Kiểm tra và rà sát lỗi phần mềm |