**CATALOG**

[1. Introduce 3](#_Toc533721958)

[1.1. Purpose of document 3](#_Toc533721959)

[1.2. Scope 3](#_Toc533721960)

[1.3. Definitions. 3](#_Toc533721961)

[1.4. References 3](#_Toc533721962)

[1.5. Document description 3](#_Toc533721963)

[2. Overview of software 4](#_Toc533721964)

[3. Architectural design 5](#_Toc533721965)

[*3.1. Software architecture* 5](#_Toc533721966)

[3.1.1. Architectural models 5](#_Toc533721967)

[3.1.2. Architectural description 5](#_Toc533721968)

[*3.2. Hardware architecture design* 6](#_Toc533721969)

[3.2.1 General model of architechture 6](#_Toc533721970)

[3.2.2 Component and package list 6](#_Toc533721971)

[3.2.3 Explaination for interactions between components 7](#_Toc533721972)

[4. Database Design 7](#_Toc533721973)

[4.1 Relational Database Design 7](#_Toc533721974)

[4.2 Database Design 8](#_Toc533721975)

[4.3 Database diagrams 13](#_Toc533721976)

[5. Component design 15](#_Toc533721977)

[5.1 Manager 15](#_Toc533721978)

[5.1.1 Manage Room 15](#_Toc533721979)

[5.1.2 Manage Service 15](#_Toc533721980)

[5.1.3 Search Customer Information 15](#_Toc533721981)

[5.1.4 Change Rules 16](#_Toc533721982)

[5.1.5 Turnover Statistics 16](#_Toc533721983)

[5.1.6 Manage Account 16](#_Toc533721984)

[5.2 Receptionist 17](#_Toc533721985)

[5.2.1 Book Room 17](#_Toc533721986)

[5.2.2. Order Service 17](#_Toc533721987)

[5.2.3 Search Room 18](#_Toc533721988)

[5.2.4 Search Customer 18](#_Toc533721989)

[5.2.5 Checkout 18](#_Toc533721990)

[6. User interface design 19](#_Toc533721991)

[6.1 Relational diagram of screens in interface 19](#_Toc533721992)

[6.2 Screen list and function description 19](#_Toc533721993)

[6.3. User interface and details 20](#_Toc533721994)

[6.3.1 Naming convention of form components 20](#_Toc533721995)

[6.3.2 Interfaces and details 22](#_Toc533721996)

[7. Implementation 34](#_Toc533721997)

[7.1. Technology 34](#_Toc533721998)

[7.2. Installing issues 34](#_Toc533721999)

[8. Test documents 35](#_Toc533722000)

[8.1. Test Cases 35](#_Toc533722001)

[8.2. Test Scenario 36](#_Toc533722002)

[8.3. Requirement Traceability Matrix 38](#_Toc533722003)

# 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.
* **Part 5** **– Component Design:** Design subsystems of the system
* **Part 6** **– User Interface Design:** An overview of the interfaces, interface images, and accompanying operations
* **Part 7 – Implementation:** Using technology and installing issues
* **Part 8 – Test Document:** Describe test case and test scenario

**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. Architectural design**

## *3.1. Software architecture*

### 3.1.1. Architectural models

The system is divided into 2 modules as follows:

- General management module: Perform function related to Manage rooms, Manage services, Manage accounts, Turnover statistics, Change rules, Search Customers

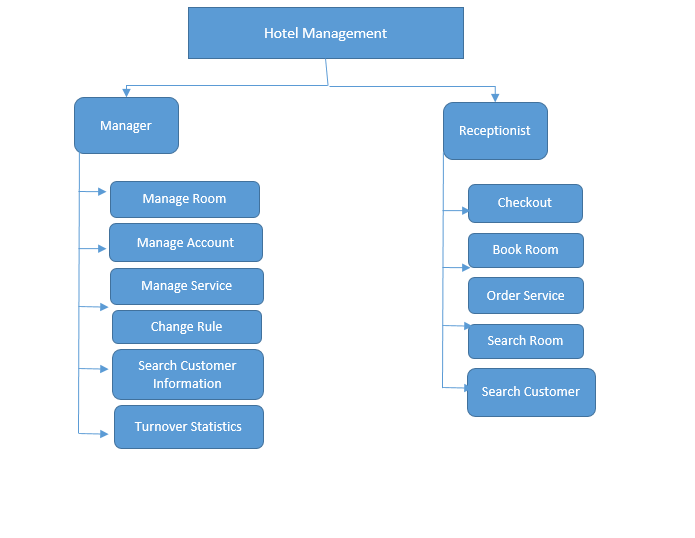
- Booking module: Book room, Checkout, Search customer, Search room information, Order Services

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.

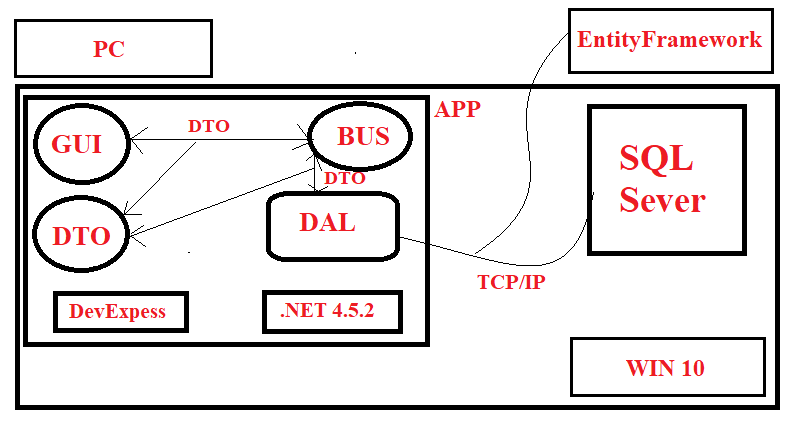
### 3.1.2. Architectural description



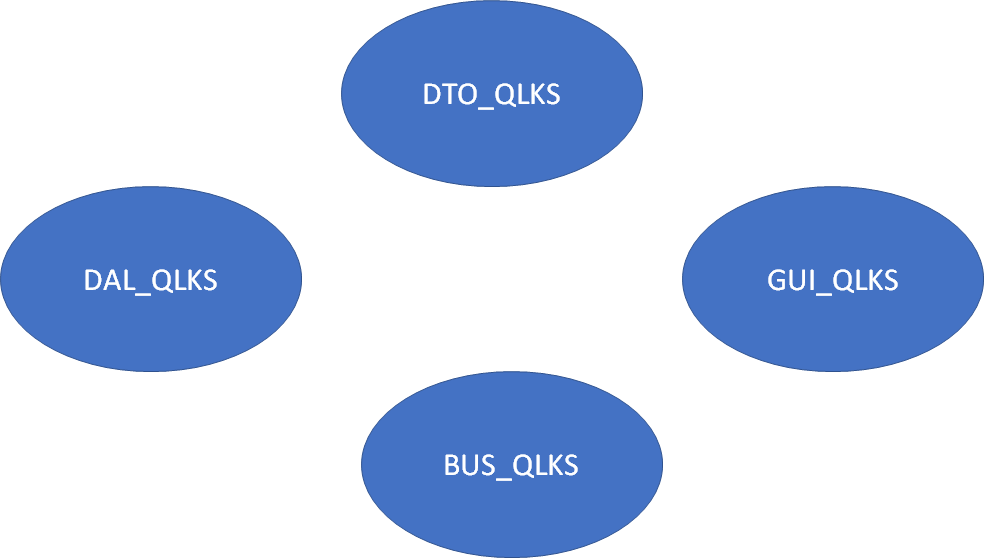
## *3.2. Hardware architecture design*

### 3.2.1 General model of architechture

This application is a 3-tier application , using EntityFramework

****

### 3.2.2 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

### 3.2.3 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 are responsible for calculation that follow the requirements sending from GUI\_QLKS, if there is no need of database BUS\_QLKS will send back data to GUI\_QLKS, otherwise pushing proceeded data to DAL\_ QLKS.

DAL\_ QLKS contain classes that are responsible for accessing database then return result to BUS\_QLKS.

DTO\_QLKS contain classes that modelize data objects in database, for transfering data between layers

**4. Database Design**

## 4.1 Relational Database Design

* **Entities**

Based on the specific activities of the bookstore 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

## 4.2 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 |

3.BAOCAO\_MATDOSUNGPHONG

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

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

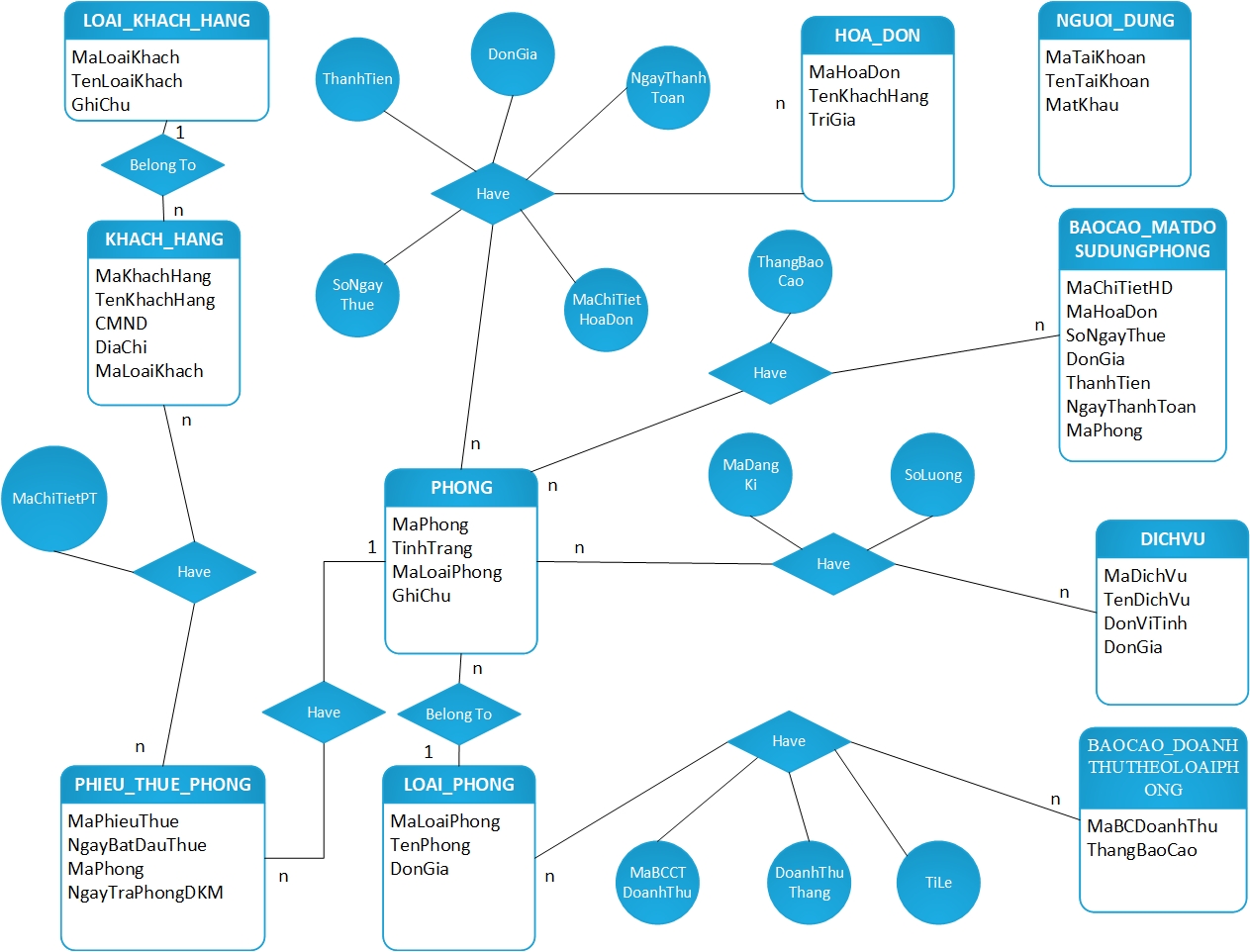
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaBCCTDoanhThu | int | Primary key | Turnover detail report code |
| 2 | MaLoaiPhong | Nvarchar(10) | Primary key | Room type code |
| 3 | MaBCDoanhThu | int | Primary key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

15.NGUOI\_DUNG

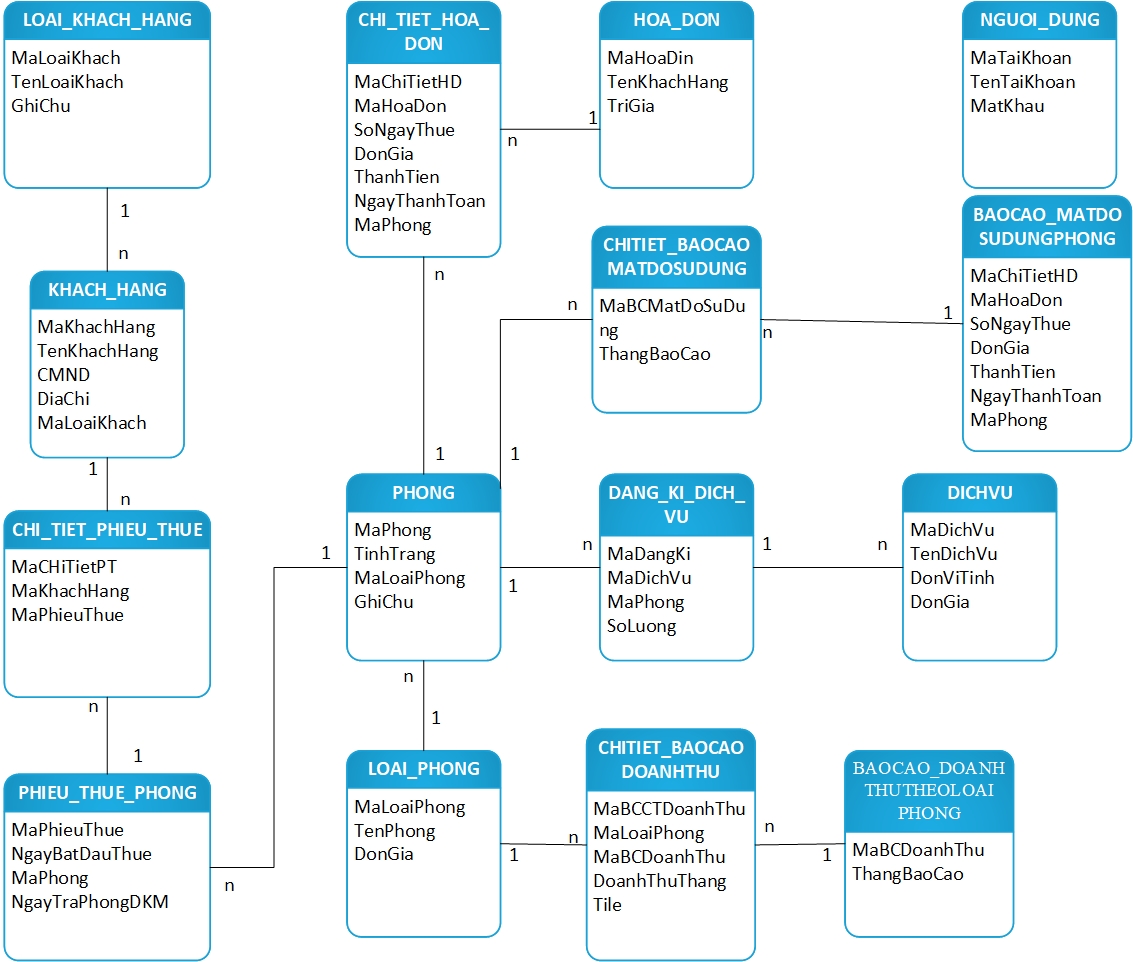
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| **1** | **MaTaiKhoan** | **int** | **Primary key** | **Account code** |
| 2 | TenTaiKhoan | char(10) |  | Username |
| 3 | Matkhau | Char(20) |  | Password |

## 4.3 Database diagrams

4.3.1 Entity – Relationship Model



4.3.2 Relationship Model



# 5. Component design

## 5.1 Manager

### 5.1.1 Manage Room

- Function information: Allow manager update(Add, Edit, Delete, Search) room information.

- Input: Room information

- Output: Room information was updated

- Describe steps:

+ Step 1: Show form display room information

+ Step 2: Manager enter information

+ Step 3: Confirm information and display result

### 5.1.2 Manage Service

- Function information: Allow manager update(Add, Edit, Delete) service information.

- Input: Service information

- Output: Room information was updated

- Describe steps:

+ Step 1: Show form display service information

+ Step 2: Manager enter information

+ Step 3: Confirm information and display result

### 5.1.3 Search Customer Information

- Function information: Allow manager find out customer information

- Input: Customer information

- Output: Customer records that meet the input

- Describe steps:

+ Step 1: Display all records of customer

+ Step 2: Manager enters information

+ Step 3: Result display automatically

+ Step 4: Manager clicks the record

+ Step 5: Display corresponding room

### 5.1.4 Change Rules

- Function information: Allow manager update(Add, Edit, Delete) rule information.

- Input: Rule information

- Output: Rule information was updated.

- Describe steps:

+ Step 1: Show form display rules information

+ Step 2: Manager enters information

+ Step 3: Confirm information and display result

### 5.1.5 Turnover Statistics

- Function information: : Allow manager see revenue statistic by month and year.

- Input: Month and year of revenue need to find.

- Output: Display corresponding revenue statistics

- Describe steps:

+ Step 1: Manager enters information

+ Step 2: Confirm information and display result

### 5.1.6 Manage Account

- Function information: Allow manager update(Add, Edit, Delete, Change security level) account information.

- Input: Account information

- Output: Account information was updated.

- Describe steps:

+ Step 1: Show form display accounts information

+ Step 2: Manager enters information

+ Step 3: Confirm information and display result

## 5.2 Receptionist

### 5.2.1 Book Room

- Function information: Allow receotionist create booking form for customer who want to book rooms.

- Input: Information of room and customer

- Output: Room is booked and add customer information

- Describe steps:

+ Step 1: Display empty rooms information.

+ Step 2: Choise room by customer requirement.

+ Step 3: Receptionist confirm room.

+ Step 4: Enter customer information

+ Step 5: Confirm information and display list customers in room.

+ Step 6: Save and create booking form.

### 5.2.2. Order Service

- Function information: Allow receotionist order service for customers by their requirement.

- Input: Information of room and service.

- Output: Service is ordered and add to service cost of that room.

- Describe steps:

+ Step 1: Choise room and service

+ Step 2: Enter quantity of service.

+ Step 3: Confirm information and display message success.

### 5.2.3 Search Room

- Function information: Allow receotionist find room by room information.

- Input: Room information

- Output: Display corresponding room

- Describe steps:

+ Step 1: Display all rooms information

+ Step 2: Enter room information

+ Step 3: Confirm and display results.

### 5.2.4 Search Customer

- Function information: Allow receotionist find room by customer information.

- Input: Customer information

- Output: Display corresponding customer record

- Describe steps:

+ Step 1: Display all customers information

+ Step 2: Enter customer information

+ Step 3: Confirm and display results.

### 5.2.5 Checkout

- Function information: Allow receotionist create bill for customer who check out.

- Input: Room and customer information

- Output: Total price, update room status to empty.

- Describe steps:

+ Step 1: Display customer and room information

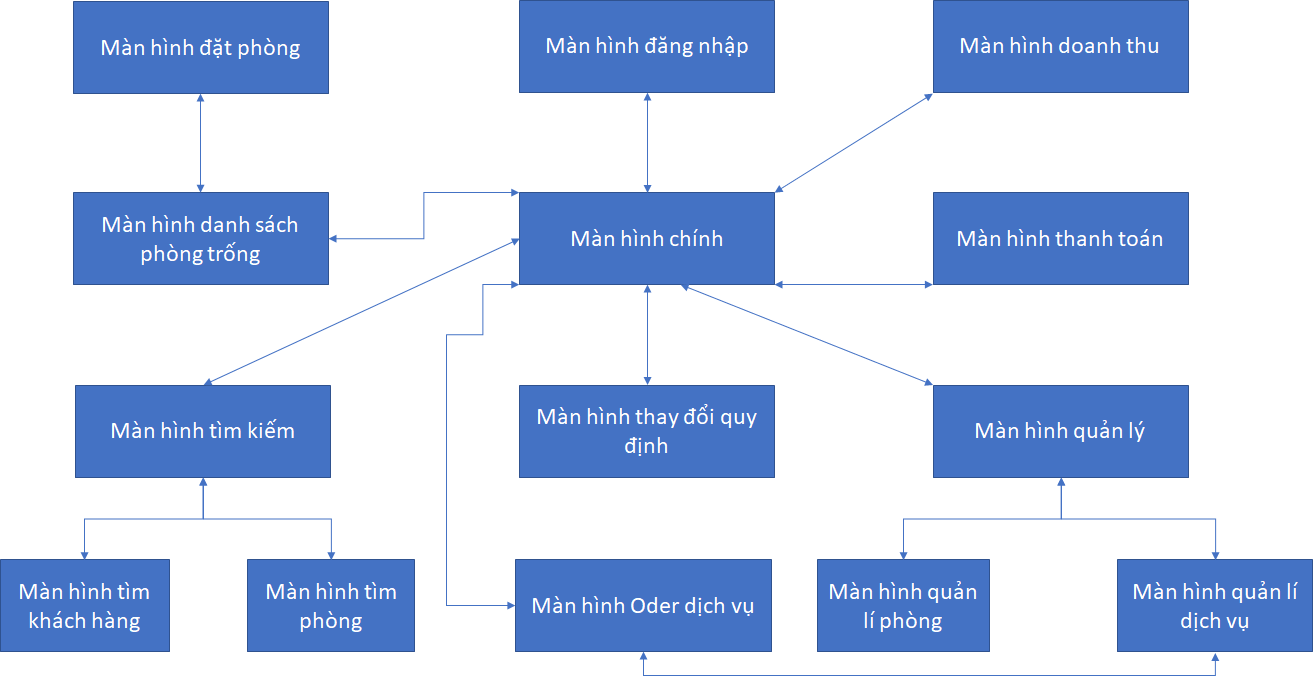
+ Step 2: Confirm and create bill

+ Step 3: Customer check the bill and confirm.

+ Step 4: Create bill and display message.

# 6. User interface design

## 6.1 Relational diagram of screens in interface

****

## 6.2 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 |

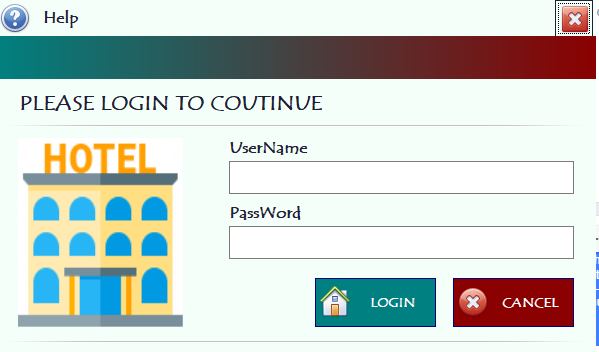
## 6.3. User interface and details

### 6.3.1 Naming convention of form components

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

### 6.3.2 Interfaces and details

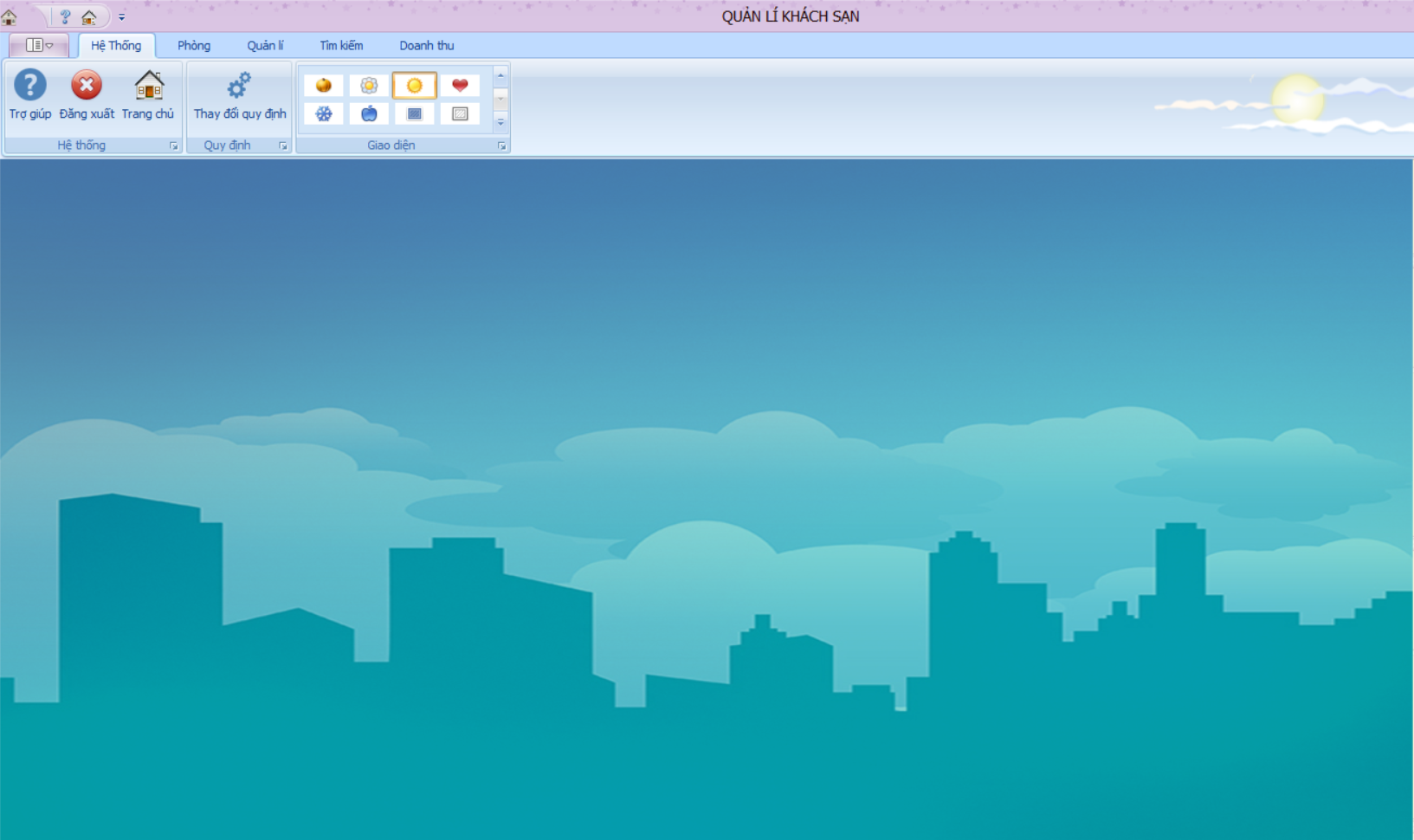
* **Login screen**



* ***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 |

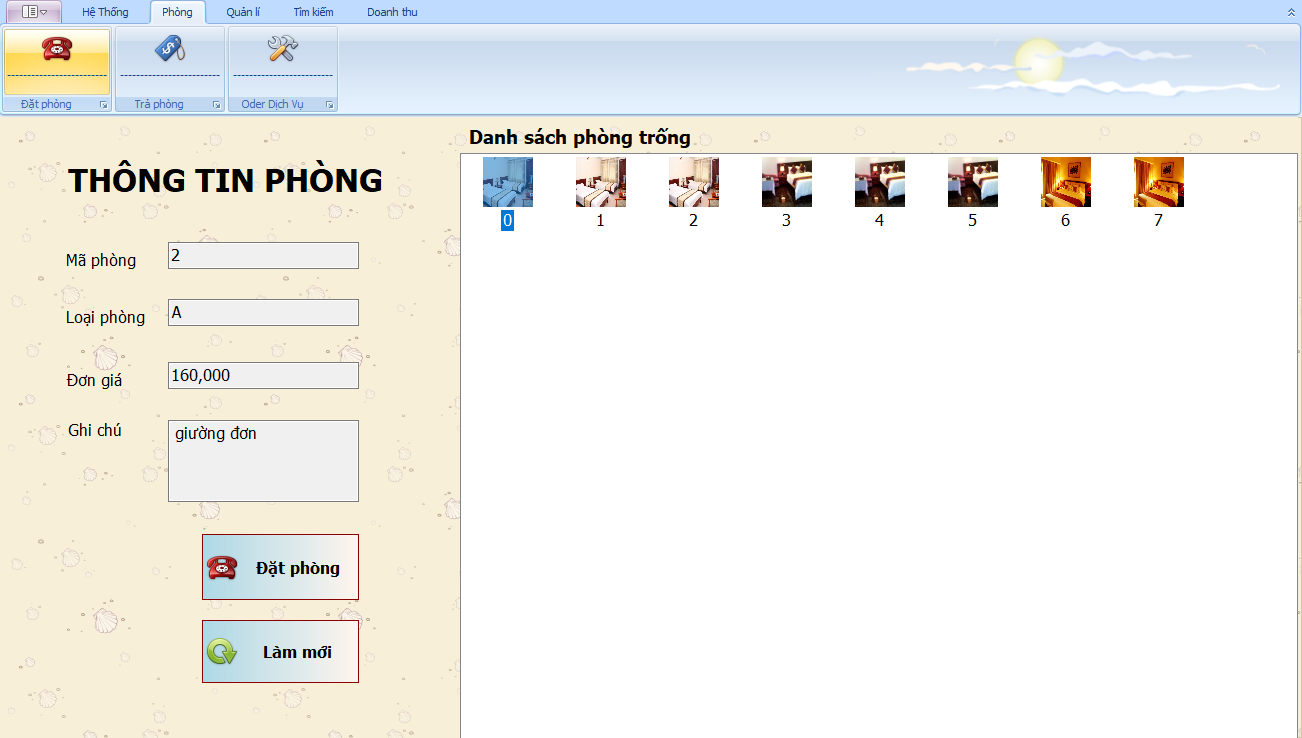
* **Main screen**



* ***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 |

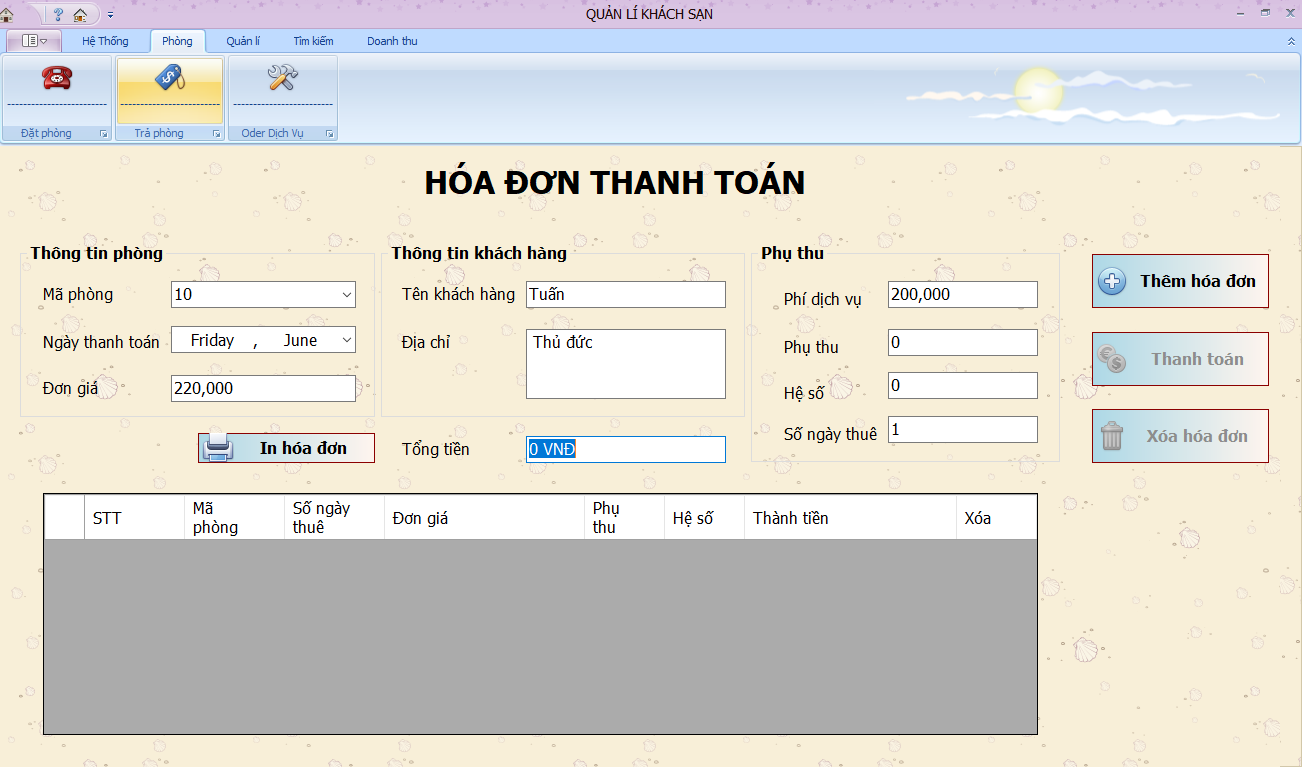
* **Book room screen**



* ***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) |

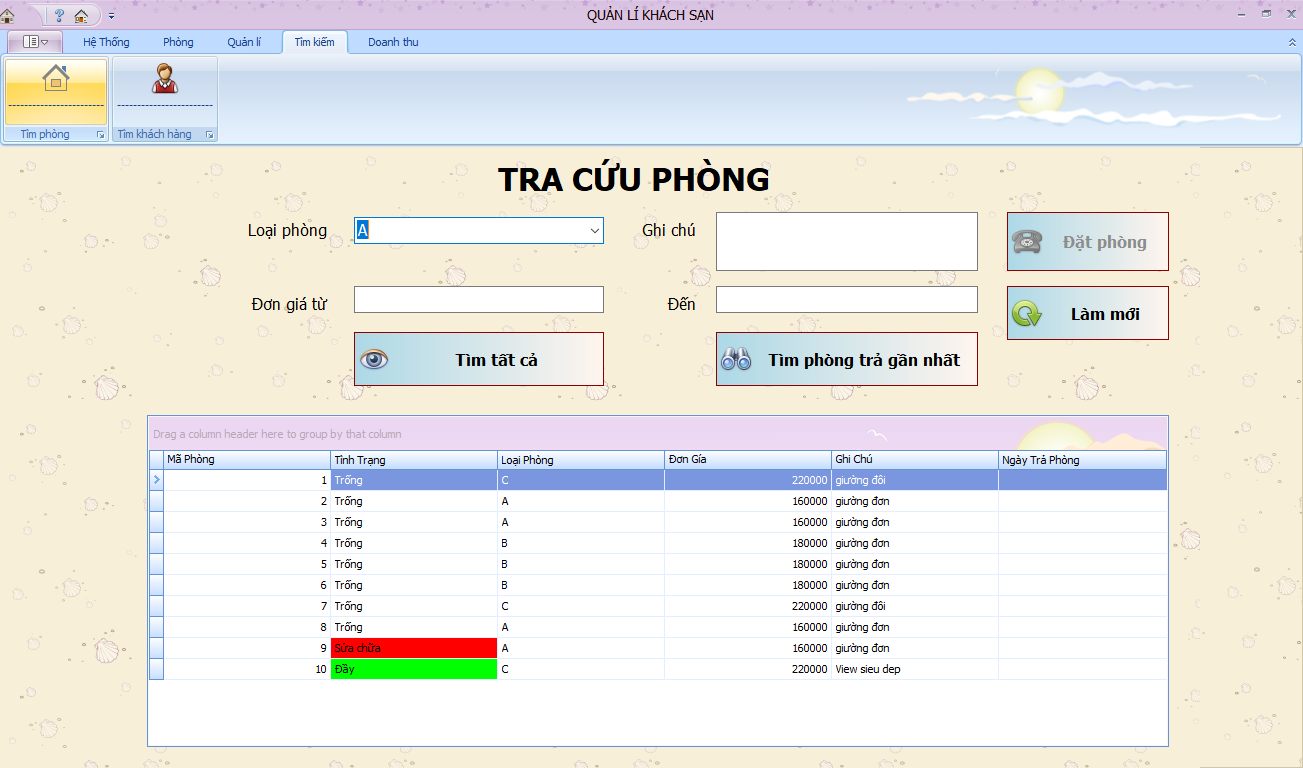
* **Checkout screen**



* ***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 |

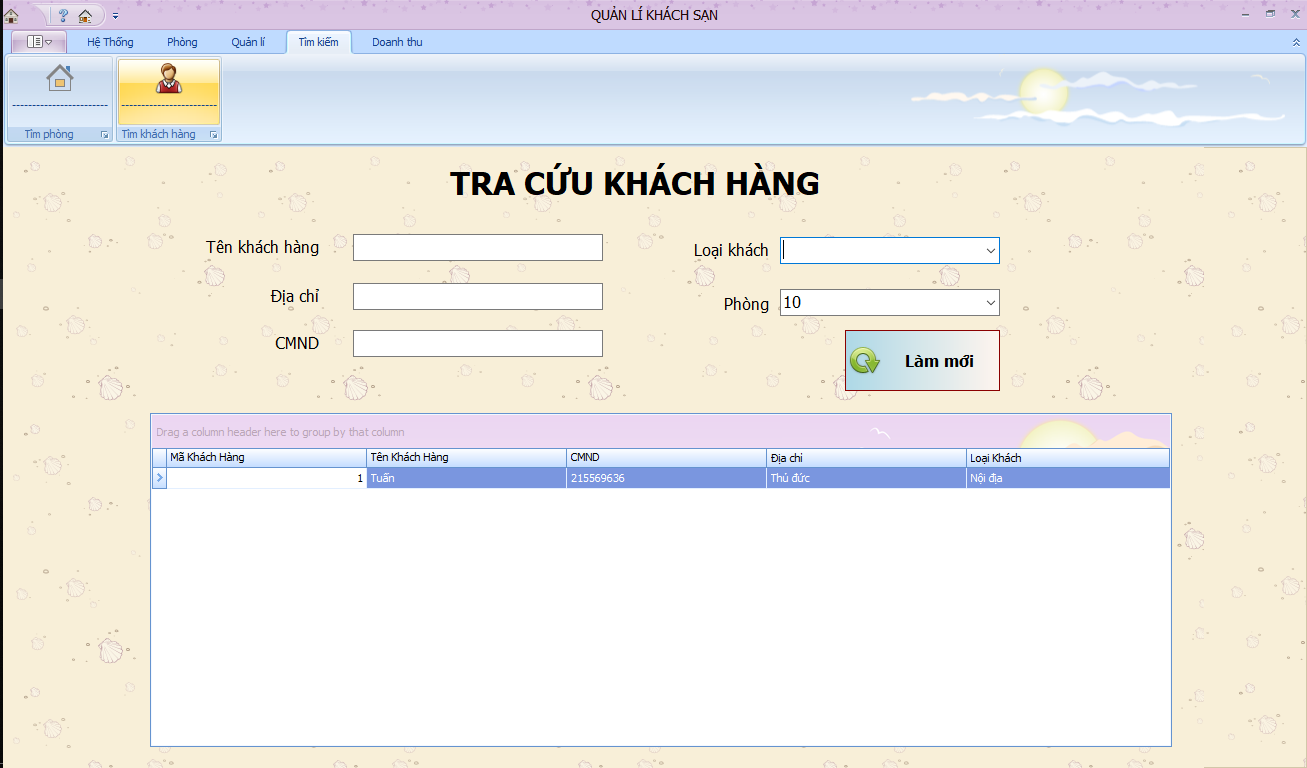
* **Find room screen**



* ***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 |

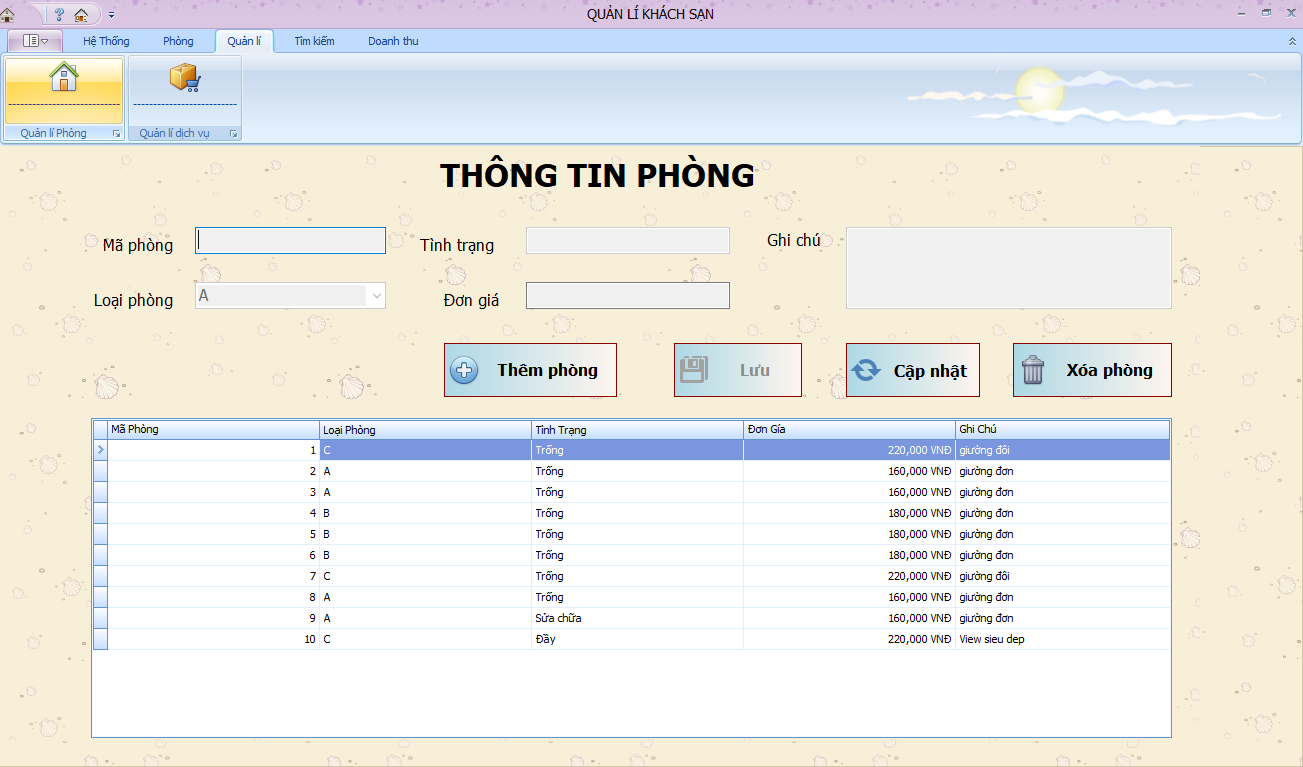
* **Find customer screen**



* ***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 |

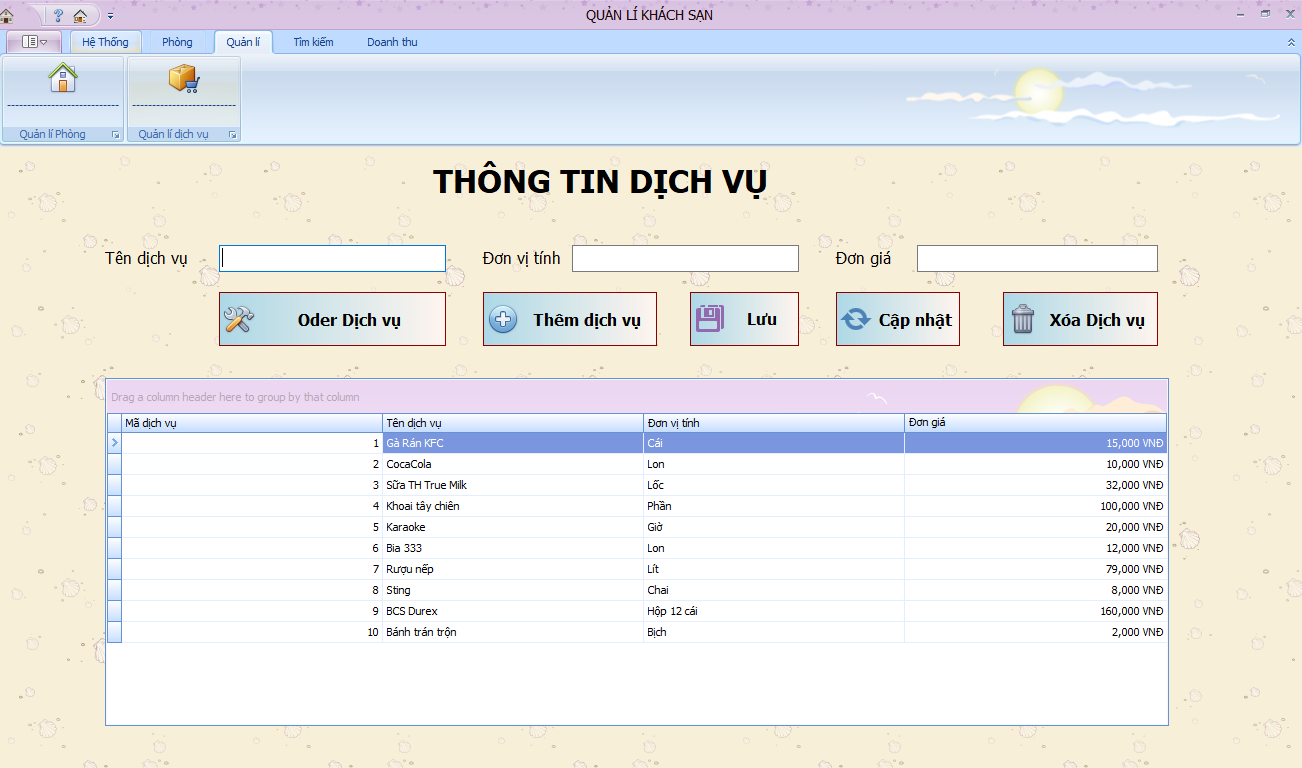
* **Room management service**



* ***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 |

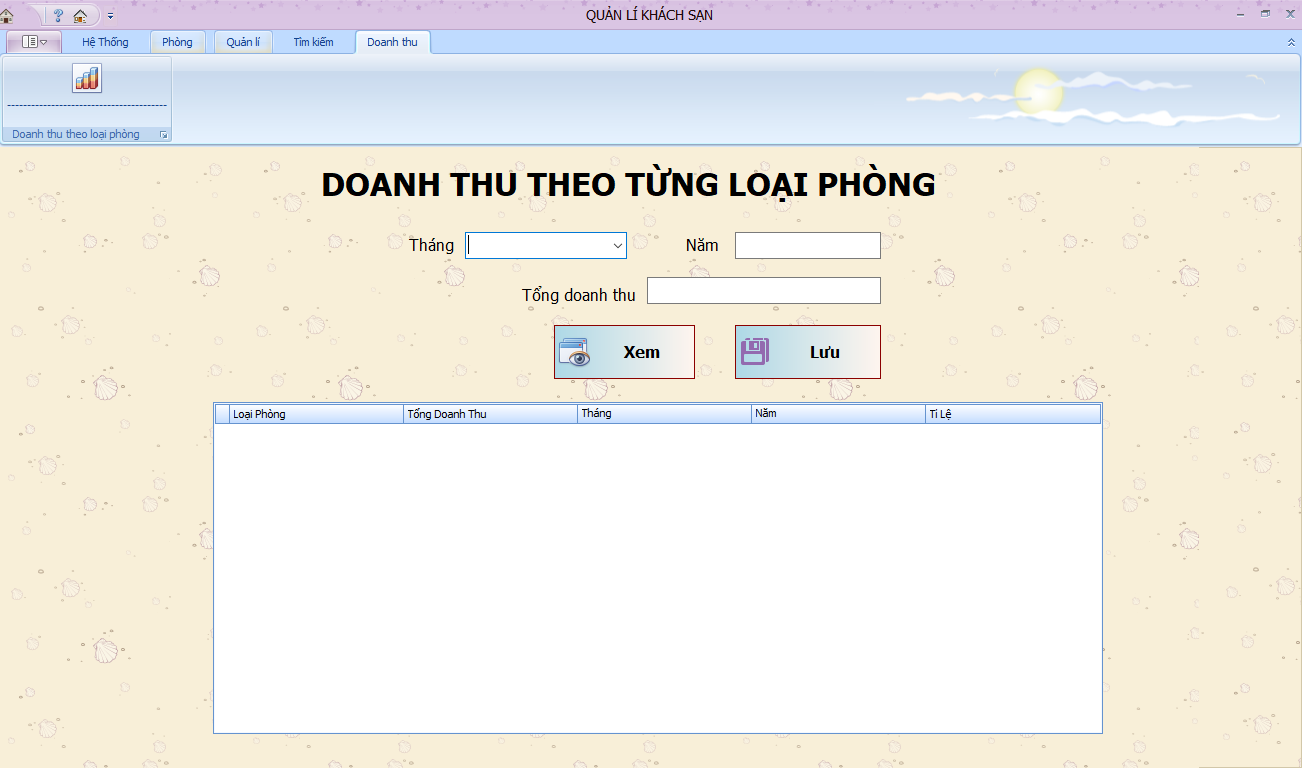
* **Order service screen**



* ***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 |

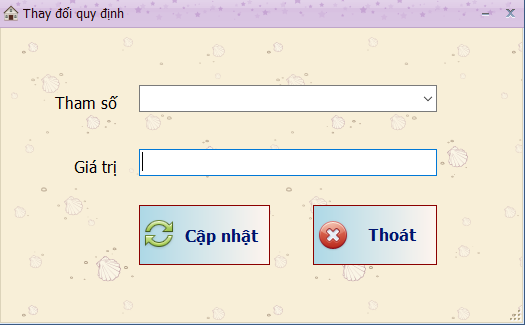
* **Turnover report screen**



* ***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 |

# 7. Implementation

## 7.1. 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

## 7.2. Installing issues

* 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

# 8. Test documents

## 8.1. Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | UC ID | Condition to be tested | Test data | Expected result | Successful |
|  | UC\_login | Log in | UserName and Password match | Log in success |  |
|  | UC\_login | Log in | UserName matches and Password does not | Log in failed |  |
|  | UC\_login | Log in | Invalid UserName and valid Password | Log in failed |  |
|  | UC\_login | Log in | Enter invalid UserName and invalid Password | Log in failed |  |
|  | UC\_booking | booking | There is at least 1 empty room | Book room successfully |  |
|  | UC\_ booking | booking | All rooms are not empty or under repairing | Book room unsuccessfully |  |
|  | UC\_ booking | booking | There is no empty room | Book room unsuccessfully |  |
|  | UC\_checkout | Checkout | All data are correctly loaded with correspoding | Check-out successfully |  |
|  | UC\_checkout | Checkout | Checkout date is earlier than starting date | 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 empty | Delete room successfully |  |
|  | UC\_delete room | Delet e  room | Room in use | 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 not in use | Service deleted |  |
|  | UC\_edit service | Edit service | Data input is valid | Edit service successfully |  |
|  | UC\_edit service | Edit service | Data input is invalid | 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 not exist | Account created |  |
|  | UC\_create new account | Create new account | Username 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 | Input data valid | 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 |  |

## 8.2. Test Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq.No | Requirement Name | Test scenario | | Test Case | |
|  |  | TS Name | TS ID | TC Name | TC ID |
| 1 | Login | Success | TS\_login\_01 | a. Username and password match | TS\_login\_01\_01 |
| Fail | TS\_login\_02 | UserName matches and Password does not | TS\_login\_02\_01 |
| Invalid UserName and valid Password | TS\_login\_02\_02 |
| 2 | Booking | Success | TS\_Reservation\_01 | There is at least 1 empty room | TS\_Reservation 01\_01  TS\_Reservation\_01\_02 |
| Fail | TS\_Reservation\_02 | All rooms are not empty or under repairing | TS\_Reservation\_02\_01 |
| There is no empty room | TS\_Reservation\_02\_02 |
| 3 | Checkout | Success | TS\_checkout\_01 | All data are correctly loaded with correspoding | TS\_checkout\_01\_01 |
|  |  |
| Fail | TS\_checkout\_02 | Checkout date is earlier than starting date | TS\_checkout\_02\_01 |
|  |  |
| 4 | Order service | Success | TS\_orderservice\_01 | Service in inventory meets requirements | TS\_orderservice\_01\_01 |
|  |  |
| Fail | TS\_orderservice\_02 | Service in inventory does not meet requirements | TS\_orderservice\_02\_01 |
|  |  |
| 5 | Add room | Success | TS\_addroom\_01 | Room does exist | TS\_addroom\_01\_01 |
| Fail | TS\_addroom\_02 | Room exists | TS\_addroom\_02\_01 |
| 6 | Edit user | Success | TS\_editroom\_01 | User not exist | TS\_editroom\_01\_01 |
| Fail | TS\_editroom\_02 | User exists | TS\_editroom\_02\_01 |
| 7 | Delete room | Success | TS\_deleteroom\_01 | Empty room | TS\_deleteroom\_01\_01 |
| Fail | TS\_deleteroom\_02 | Room in use | TS\_deleteroom\_02\_01 |
| 8 | Find room | Success | TS\_findroom\_01 | Has that room type | TS\_findroom\_01\_01 |
| Fail | TS\_findroom\_02 | Does not has that room type | TS\_findroom\_02\_01 |
| 9 | Add service | Success | TS\_addservice\_01 | Service not exist | TS\_addservice\_01\_01 |
| Fail | TS\_addservice\_02 | Service exists | TS\_addservice\_02\_01 |
| 10 | Edit service | Success | TS\_editservice\_01 | Data input is valid | TS\_editservice\_01\_01 |
| Fail | TS\_editservice\_02 | Data input is invalid | TS\_editservice\_02\_01 |
| 11 | Delete service | Success | TS\_deleteservice\_01 | Service not in use | TS\_deleteservice\_01\_01 |
| Fail | TS\_deleteservice\_02 | Service in use | TS\_deleteservice\_02\_01 |
| 12 | Create new account | Success | TS\_createnewacaccount\_01 | Username not exist | TS\_createnewacaccount\_01\_01 |
| Fail | TS\_createnewacaccount\_02 | Username existed | TS\_createnewacaccount\_02\_01 |
| 13 | Change password | Success | TS\_changepassword\_01 | Password follows the rules | TS\_changepassword\_01\_01 |
| Fail | TS\_changepassword\_02 | Password does not follow the rules | TS\_changepassword\_02\_01 |
| 14 | Change regulation | Success | TS\_changeregulation\_01 | Input data valid | TS\_changeregulation\_01\_01 |
| Fail | TS\_changeregulation\_02 | Input data not valid | TS\_changeregulation\_02\_01 |
| 15 | Turnover statistics by room type | Success | TS\_turnoverstatisticsbyroomtype\_01 | Input time matches the room lending time | TS\_turnoverstatisticsbyroomtype\_01\_01 |
| Fail | TS\_turnoverstatisticsbyroomtype\_02 | Input time matches the room using time | TS\_turnoverstatisticsbyroomtype\_02\_01 |

## 8.3. Requirement Traceability Matrix

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