**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.
* **Part 5** **– Design components (module):** Design subsystems of the system.
* **Part 6 – User Interface Design:** An overview of the interfaces, interface images, and accompanying operations.

# 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. Architectural 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

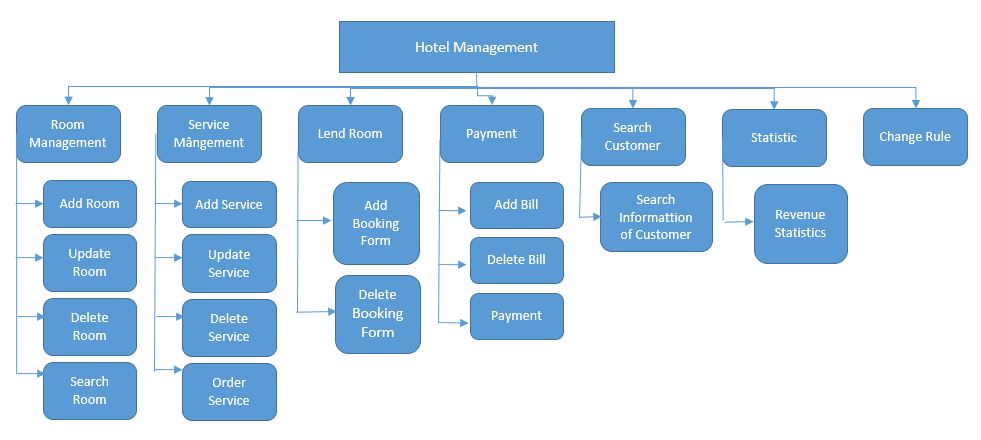
- Change Rule: Perform functions related to: Change rule in hotel

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.

### 3.2. Architectural description



# 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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

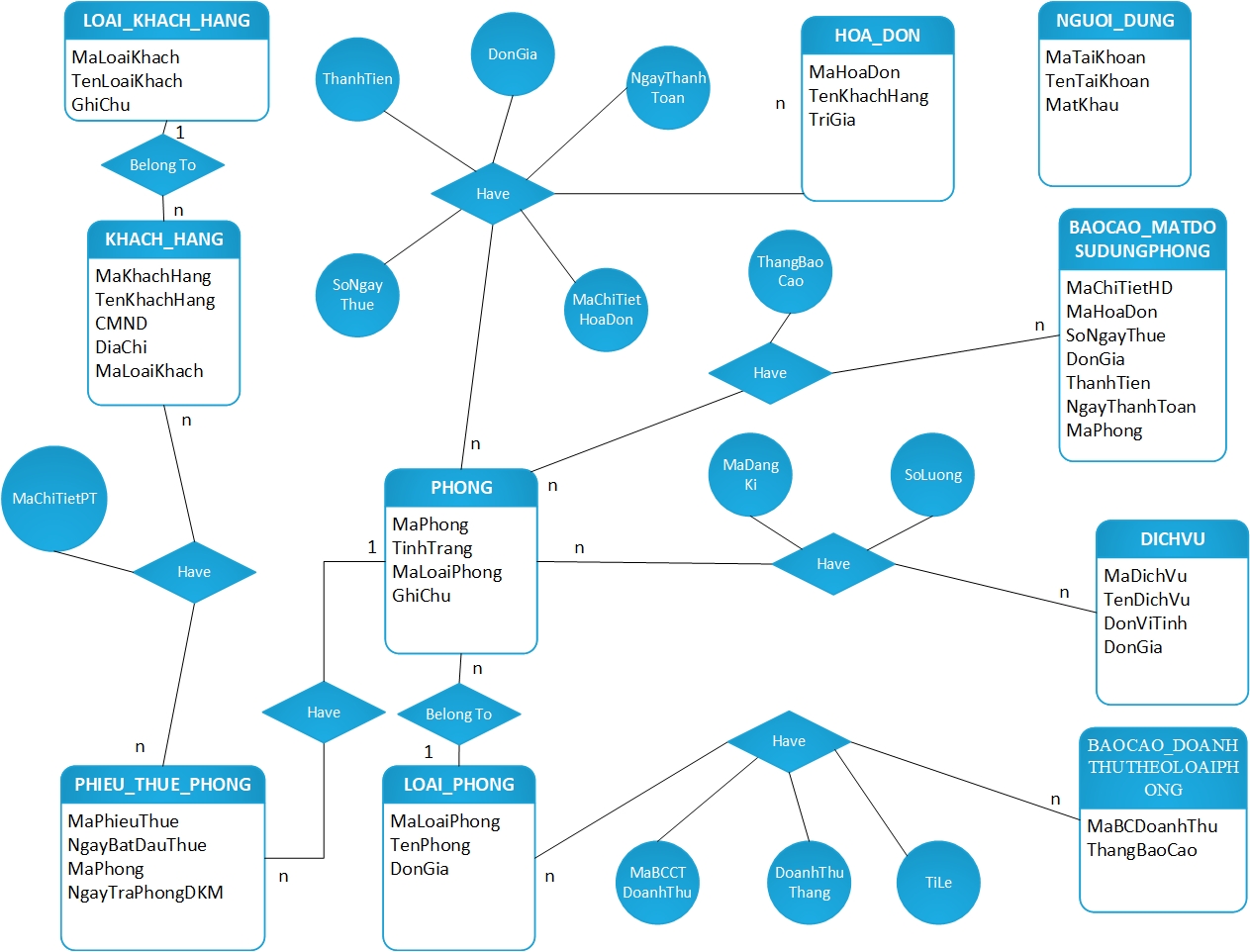
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| 1 | MaBCCTDoanhThu | int | Primary key | Turnover detail report code |
| 2 | MaLoaiPhong | Nvarchar(10) | Primary key | Room type code |
| 3 | MaBCDoanhThu | int | Primary key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

15.NGUOI\_DUNG

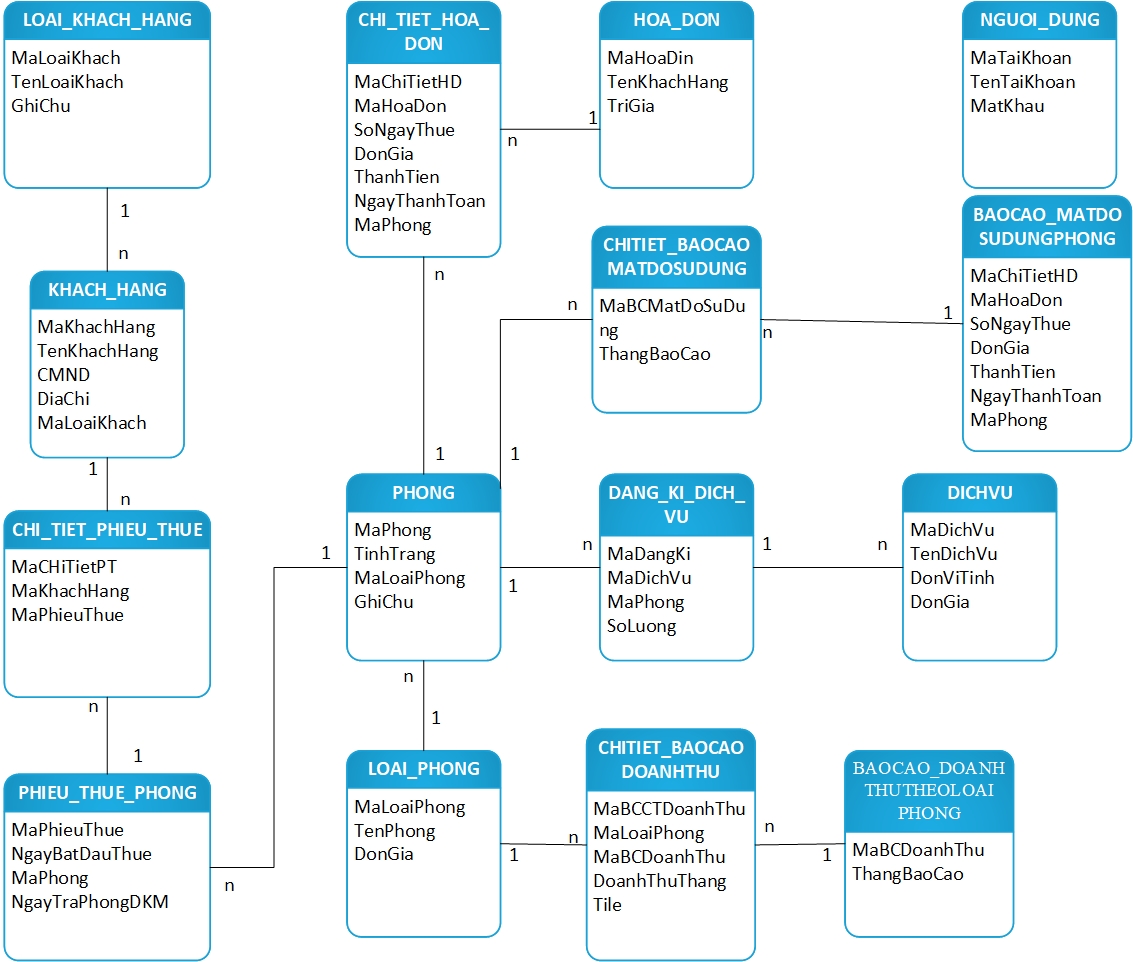
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Attribute** | **Data type** | **Constraint** | **Note** |
| **1** | **MaTaiKhoan** | **int** | **Primary key** | **Account code** |
| 2 | TenTaiKhoan | char(10) |  | Username |
| 3 | Matkhau | Char(20) |  | Password |

### 4.3 Database diagrams

4.3.1 Entity – Relationship Model



4.3.2 Relationship Model



6. User Interface Design

**\* Các quy ước đặt tên cho các thành phần của form**

|  |  |  |  |
| --- | --- | --- | --- |
| ***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 |

7. Test Case