**ĐIỂM SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

**NHẬN XÉT CỦA GIÁO VIÊN:**

…………………………………………………………………………………….

…………………………………………………………………………………….

…………………………………………………………………………………….

…………………………………………………………………………………….

…………………………………………………………………………………….

…………………………………………………………………………………….

…………………………………………………………………………………….

Tp. Hồ Chí Minh, ngày ….., tháng 12, năm 2020

Giáo viên chấm điểm

*(Ký và ghi họ tên)*

**LỜI CẢM ƠN**

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, tiến sĩ Nguyễn Thành Sơn, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm ơn thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau.

Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

**Tp. Hồ Chí Minh, ngày 21 tháng 12 năm 2020**

**Nhóm sinh viên thực hiện**

Trường ĐH Sư phạm Kỹ thuật TP. HCM

Khoa Công nghệ thông tin

**MÔ TẢ ĐỀ TÀI MÔN HỌC HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

Họ và Tên SV thực hiện 1 : **Nguyễn Thanh Minh Đức** Mã số SV : **19110017**

Họ và Tên SV thực hiện 2 : **Lê Vũ Minh Hoàng** Mã số SV : **19110181**

Họ và Tên SV thực hiện 2 :**Lê Trần Minh Nhựt** Mã số SV : **19110257**

Họ và Tên SV thực hiện 2 : **Nguyễn Thị Xuân Thanh** Mã số SV : **19110285**

Họ và Tên SV thực hiện 2 : **Đoàn Thị Thanh Vân** Mã số SV : **19110313**

Thời gian thực hiện : **Từ : 5/3/2021 Đến : 25/5/2021** Ngành:**Công nghệ thông tin**

Tên đề tài : **Phần mềm quản lý nhà xe**

GV hướng dẫn : **TS. Nguyễn Thành Sơn**

**Nhiệm vụ của đề tài :** Xây dựng phần mềm quản lý nhà xe đảm bảo đủ các chức năng sau:

1. Quản lý thông tin nhân viên.
2. Quản lý thời gian làm việc của nhân viên.
3. Quản lý các khu vực của nhà xe.

4. Quản lý thông tin khách hàng.

5. Quản lý tiền lương nhân viên.

6. Quản lí thông tin lượng xe đã gửi.

**MỤC LỤC**

PHẦN 1. MỞ ĐẦU 1

1. LÝ DO CHỌN ĐỀ TÀI 1
2. CÁC CHỨC NĂNG CỦA ĐỀ TÀI 1
3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 2
4. GIAO DIỆN DỰ KIẾN 3
5. KẾ HOẠCH THỰC HIỆN 4
6. Kế hoạch 4
7. Phân công công việc 5

PHẦN 2 : NỘI DUNG 7

[CHƯƠNG 1: THIẾT KẾ CƠ SỞ DỮ LIỆU](#_heading=h.30j0zll)

[1.1. Mô hình quan hệ](#_heading=h.1fob9te)

[1.2. Tạo các cơ sở dữ liệu](#_heading=h.3znysh7)

[1.3. Database Diagram](#_heading=h.2et92p0)

[CHƯƠNG 2: TẠO VIEW, STORE PROCEDURE, FUNCTION, TRIGGER TẦNG CƠ SỞ DỮ LIỆU](#_heading=h.tyjcwt)

[2.1. Tạo các View](#_heading=h.3dy6vkm)

[2.2. Tạo các Store Procedure](#_heading=h.1t3h5sf)

[2.3. Tạo các Function 9](#_heading=h.4d34og8)

[2.4. Tạo các Trigger](#_heading=h.2s8eyo1)

[2.5. Tạo các Role, phân quyền](#_heading=h.17dp8vu)

[CHƯƠNG 3: THIẾT KẾ GIAO DIỆN 8](#_heading=h.3rdcrjn)

[3.1. Form Login 8](#_heading=h.26in1rg)

[3.2. Form Main 8](#_heading=h.lnxbz9)

[3.3. Form Staff 9](#_heading=h.35nkun2)

[3.4. Form Working](#_heading=h.1ksv4uv)

[3.5. Form Salary](#_heading=h.44sinio)

[3.6.Form Customer 2](#_heading=h.44sinio)

[3.7.Form Sticket 3](#_heading=h.44sinio)

[3.8. Form Area 4](#_heading=h.44sinio)

CHƯƠNG 4: KẾT NỐI VỚI CƠ SỞ DỮ LIỆU……………………………………45

4.1 Chức năng của các lớp phương thức .…………………………………………….45

4.2 Đặc tả các phương pháp trong lớp ……………………………………………….46

PHẦN 3: KẾT LUẬN……………………………………………………………..….53

1. KẾT LUẬN……………………………………………………………………....53

2 . HƯỚNG PHÁT TRIỂN………………………………......………………….….53

3. TÀLIỆUTHAMKHẢO…………………………………………………………...54

**DANH MỤC CÁC HÌNH**

Hình 2.1.1: ERD 7

Hình 2.1.2: Class Diagram 10

Hình 2.3.1. Thiết kế Form Login 38

Hình 2.3.2: Thiết kế Form MainMenu 39

Hình 2.3.3: Thiết kế Form Staff 40

Hình 2.3.4: Thiết kế Form Work 40

Hình 2.3.5: Thiết kế Form Salary 43

Hình 2.3.6 Thiết kế Form Customer…………………………………………………..43

Hình 2.3.7: Thiết kế Form Sticket 43

Hình 2.3.8: Thiết kế Form Area 44

**DANH MỤC CÁC BẢNG**

Bảng 1.5.1. Kế hoạch thực hiện theo tuần 4

Bảng 1.5.2. Phân công công việc 4

Bảng 2.1.1. Mô tả các Table 8

Bảng 2.1.2. Lập trình tạo Table 8

Bảng 2.2.1. Tạo các View 11

Bảng 2.2.2. Store Procedure cho Employee 12

Bảng 2.2.3. Store Procedure cho Area 13

Bảng 2.2.4. Store Procedure cho Customer 14

Bảng 2.2.5. Store Procedure cho Salary 16

Bảng 2.2.6. Store Procedure cho Ticket 17

Bảng 2.2.7. Store Procedure cho Work 18

Bảng 2.2.8. Function cho Employee 19

Bảng 2.2.9. Function cho Ticket 20

Bảng 2.2.10. Function cho Work 21

Bảng 2.2.10. Function cho Salary 21

Bảng 2.2.11. Trigger cho Employee 22

Bảng 2.2.12. Trigger cho Salary 28

Bảng 2.2.13. Trigger cho Customer 30

Bảng 2.2.14. Trigger cho Ticket 30

Bảng 2.2.15. Trigger cho Work 35

Bảng 2.2.16. Tạo role Manager và phân quyền……………………………………….35

Bảng 2.2.17. Tạo role Employee và phân quyền……………………………………...37

Bảng 2.4.1. Đặt tả các lớp chức năng …………………………………………………45

Bảng 2.4.2. Đặc tả các phương thức trong lớp AreaDal ……………………………...46

Bảng 2.4.3. Đặc tả các phương thức trong lớp SalaryDal…………………………… 47

Bảng 2.4.4. Đặc tả các phương thức trong lớp CustomerDal…………………………48

Bảng 2.4.5. Đặc tả các phương thức trong lớp EmployeeDal ………………………...49

Bảng 2.4.6. Đặc tả các phương thức trong lớp TicketDal …………………………….50

Bảng 2.4.8. Đặc tả các phương thức trong lớp WorkDal ……………………………..51

**PHẦN 1. MỞ ĐẦU**

1. **LÝ DO CHỌN ĐỀ TÀI**

Một trong những yêu cầu cấp bách hiện nay là đưa công nghệ thông tin vào công tác quản lí nhằm làm giảm bớt sức lao động của con người, tiết kiệm thời gian, tăng độ chính xác, thu hẹp không gian lưu trữ, tránh thất thoát và đảm bảo tính an toàn của dữ liệu. Vì vậy, cần có một phần mềm giúp quản lý phòng Lab một cách đơn giản và hiệu quả.

Phần mềm giúp người sử dụng có cái nhìn trực quan nhất về thông tin dữ liệu. Với người quản lý sẽ có thể quản lý nhân sự một các dễ dàng mà không cần tốn quá nhiều công sức và thời gian. Với nhân viên, họ sẽ xem được thông tin chính xác nhất về công việc được giao. Người dùng chỉ cần thao tác đúng chính xác các mục được định dạng sẵn trên giao diện, chương trình đảm bảo dữ liệu được truy suất ra chính xác nhất.

1. **CÁC CHỨC NĂNG CỦA ĐỀ TÀI**

* Thêm, sửa, xóa, cập nhật nhân viên nếu có sự thay đổi.
* Phân công giờ làm việc cho mỗi nhân viên..
* Thêm, sửa, xóa thông tin vé gửi xe.
* Quản lí thời gian hoạt động của từng khu vực nhà xe.
* Phân quyền đăng nhập quản lí và nhân viên.
* Điểm danh thời gian làm việc của nhân viên.
* Tính toán tiền lương nhân viên.

**3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

Nhà xe cần chương trình quản lý nhân viên, doanh thu và khu vực bao gồm quản lý các thông tin sau:

1. **Nhân viên** bao gồm mã nhân viên (khóa chính), tên nhân viên, ngày sinh, địa chỉ, số điện thoại, giới tính, khu vực, mã người quản lí, mật khẩu đăng nhập, lương cơ bản. Mỗi nhân viên được xếp vào vị trị :Quản lí và nhân viên.
2. **Công** **việc** bao gồm mã nhân viên (khóa chính), ngày làm. Quản lí được quyền chỉnh sửa ngày làm cho nhân viên. Công việc ghi lại thời gian làm việc của từng nhân viên.

**c) Khu vực** bao gồm mã khu vực (khóa chính), địa điểm, sức chứa, giờ mở cửa, giờ đóng cửa. Mỗi khu vực có thời gian hoạt động riêng biệt.

**d) Lương nhân viên** có mã nhân viên (khóa chính), tháng, năm, tiền phạt, tổng giá trị lương. Nhân viên có tổng giá trị lượng được tính theo số lượng thời gian đã làm việc và số tiền bị phạt.

**e) Khách hàng** bao gồm mã khách hàng (khóa chính), ngày làm thẻ, ngày hết hạn**.**

**f) Vé xe** có mã khách hàng, mã nhân viên, mã khu vực, ngày gửi, giờ vào, giờ ra. Vé xe có mã khách hàng và mã nhân viên là khóa chính.

**4. GIAO DIỆN DỰ KIẾN**

**Trang đăng nhập** dùng để phân quyền truy cập vào phần mềm quản lí nhà xe bao gồm nhân viên và quản lý của nhà xe đó.

**Trang trang chủ** hiện thị hình ảnh trường đại học Sư phạm Kĩ thuật Thành phố Hồ Chí Minh.

**Trang Quản lí xe** hiện thị đầy đủ danh sách thông tin chi tiết của khách hàng có giao dịch gửi xe. Trang thực hiện các chức năng thêm, sửa, xóa thông tin khách hàng và vé xe.

**Trang Nhân sự** hiện thị thông tin nhân viên. Trang thực hiện các chức năng thêm, sửa, xóa thông tin khách hàng.

**Trang phân công việc** hiện thị danh sách công việc. Trang thực hiện các chức năng thêm, sửa, xóa các công việc được phân công theo thời gian của nhân viên.

**Trang lương nhân viên** hiện thị bảng lương của nhân viên. Trang thực hiện các chức năng thêm, sửa, xóa các bảng lương của từng nhân viên.

**Trang bãi đậu xe** hiện thị danh sách thông tin khu vực nhà xe. Trang thực hiện các chức năng thêm, sửa, xóa thông tin khu vực nhà xe.

1. **KẾ HOẠCH THỰC HIỆN**
2. **Kế hoạch**

*Bảng 1.5.1. Kế hoạch thực hiện theo tuần*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TUẦN** | **CÔNG VIỆC THỰC HIỆN** | **NGÀY BD** | **NGÀY KT** | **KẾT QUẢ** |
| 1 | Viết mô tả về đề tài (nêu rõ các thành phần dữ liệu, chức năng và giao diện cần có để project chạy tốt | 5/3/2021 | 31/3/2021 | Hoàn thành |
| 2 | Thiết kế CSDL và các ràng buộc (dùng ERD) | 15/3/2021 | 2/5/2021 | Hoàn thành |
| 3 | Cài đặt CSDL và các ràng buộc, trigger. Nhập dữ liệu | 16/3/2021 | 15/5/2021 | Hoàn thành |
| 4 | Thiết kế các view (trình bày bằng mã giả) | 17/3/2021 | 4/5/2021 | Hoàn thành |
| 5 | Cài đặt các view | 18/3/2021 | 15/5/2021 | Hoàn thành |
| 6 | Thiết kế giao diện, xác định các users và quyền sử dụng CSDL của họ | 20/4/2021 | 20/5/2021 | Hoàn thành |
| 7 | Cài đặt giao diện và kết nối chương trình với CSDL, tạo user, phần quyền | 25/4/2021 | 20/5/2021 | Hoàn thành |
| 8 | Thiết kế các hàm, thủ tục thực hiện các chức năng của đề tài | 20/4/2021 | 20/5/2021 | Hoàn thành |
| 9 | 20/4/2021 | 20/5/2021 | Hoàn thành |
| 10 | Cài đặt các hàm, thủ tục thực hiện các chức năng của đề tài | 20/4/2021 | 20/5/2021 | Hoàn thành |
| 11 | Viết báo cáo | 15/5/2020 | 25/5/2020 | Hoàn thành |

1. **Phân công công việc**

*Bảng 1.5.2. Phân công công việc*

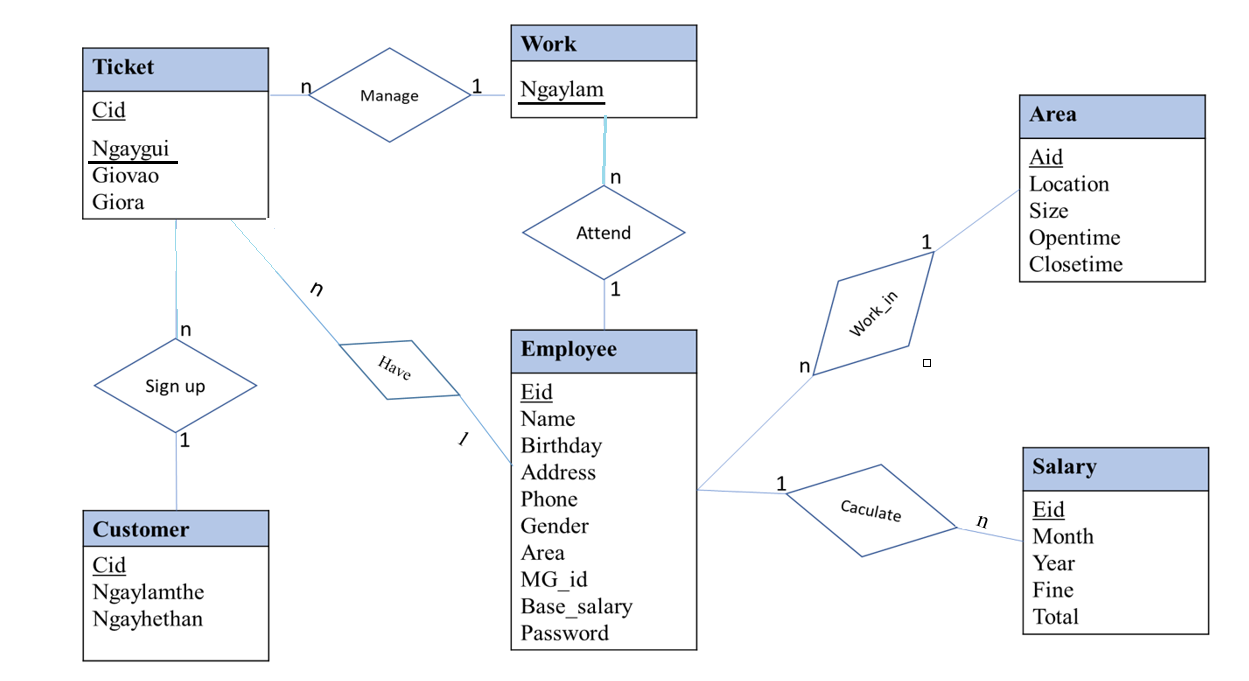
|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Tên sinh viên** | **Mô tả công việc** | **Đóng góp** |
| 1 | Nguyễn Thanh Minh Đức | * Lập kế hoạch thực hiện, phân chia công việc các thành viên trong nhóm. * Thiết kế cơ sở dữ liệu. * Phụ trách form Trang chủ * Phụ trách form Đăng nhập * Phụ trách form quản lí xe * Phụ trách form nhân sự và tài khoản | 20% |
| 2 | Lê Vũ Minh Hoàng | * Thiết kế cơ sở dữ liệu. * Thiết kế bảng Area * Thiết kế bảng Ticket * Thiết kế bảng Customer | 20% |
| 3 | Lê Trần Minh Nhựt | * Thiết kế cơ sở dữ liệu * Thiết kế trigger * Thiết kế function * Thiết kế bảng Employee * Thiết kế bảng Salary * Thiết kế bảng Work | 20% |
| 4 | Nguyễn Thị Xuân Thanh | * Thiết kế cơ sở dữ liệu * Phụ trách thực hiện báo cáo * Vẽ ERD * Thiết kế Powerpoint | 20% |
| 5 | Đoàn Thị Thanh Vân | * Thiết kế cơ sở dữ liệu * Thiết kế function * Phụ trách thực hiện báo cáo * Phụ trách form thông tin thành viên | 20% |

**PHẦN 2. NỘI DUNG**

**CHƯƠNG 1: THIẾT KẾ CƠ SỞ DỮ LIỆU**

**1.1. Mô hình quan hệ**

1.1.1. Mô hình ERD



Hình 2.1.1.ERD

* + 1. *Chuyển ERD sang mô hình quan hệ*

**Employee** *(Eid, Name, Birthday, Address, Phone, Gender, Area, MG\_id, Base\_salary, Password ,Area\_id)*

**Area** *(Aid, Location, Size,Opentime, Closetime)*

**Work** *(Eid, Ngaylam)*

**Ticket** *(Cid, Eid, Ngaygui, Giovao, Giora)*

**Customer** *(Cid, Ngaylamthe, Ngayhethan)*

**Salary** *(Eid, Month, Year, Fine, Total)*

**1.2. Tạo các cơ sở dữ liệu**

*1.1.1. Mô tả các Table*

*Bảng 2.1.1. Mô tả các Table*

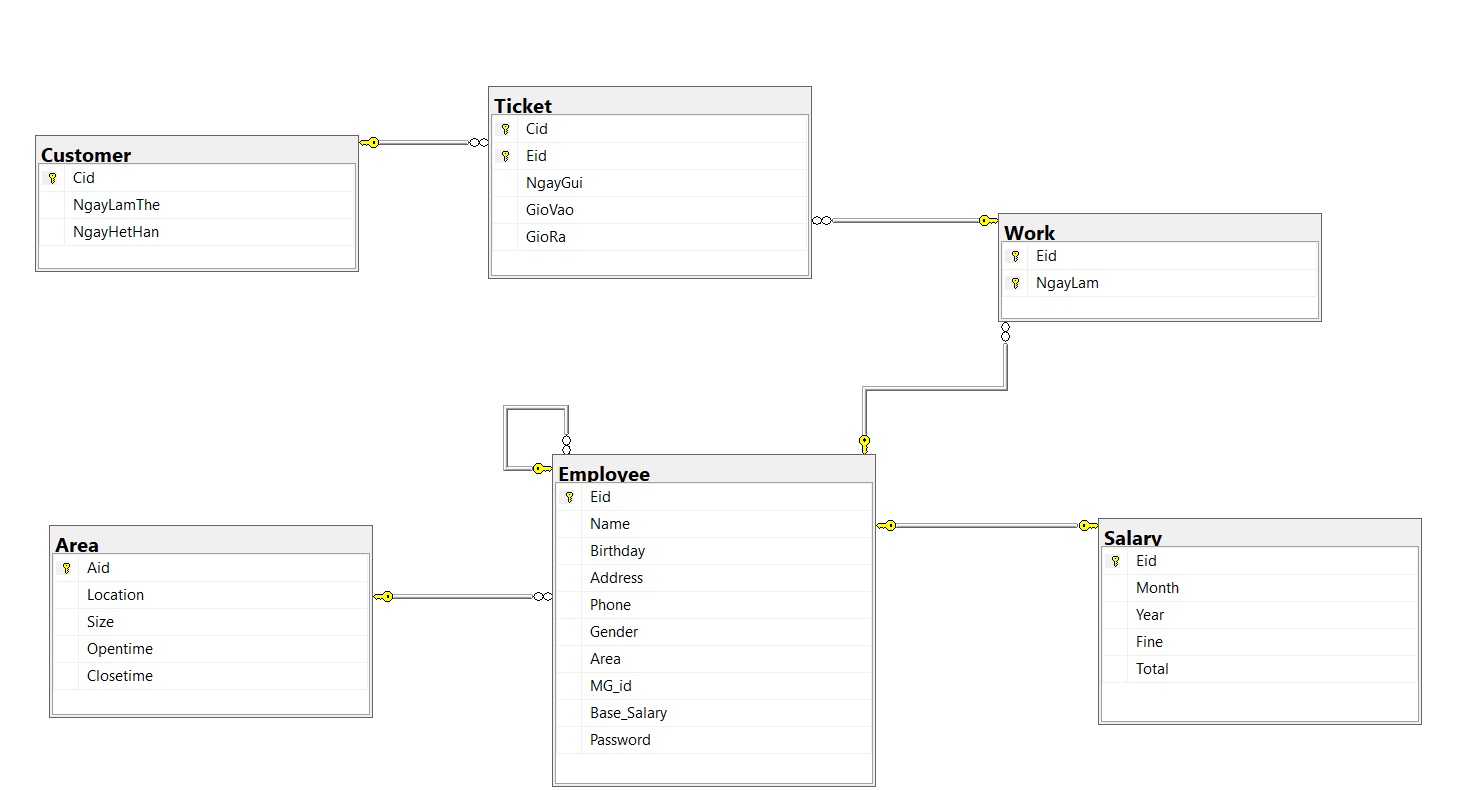
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên table** | **Mục đích** |
| 1 | Employee | - Chứa thông tin cá nhân và thông tin đăng nhập của nhân viên. |
| 2 | Work | - Chứa thông tin giờ làm việc của nhân viên. |
| 3 | Salary | - Chứa thông tin về lương nhân viên. |
| 4 | Area | - Chứa thông tin về khu vực nhà xe. |
| 5 | Customer | - Chứa thông tin về khách hàng. |
| 6 | Thông tin gửi xe | - Chứa thông tin về vé xe. |

*1.2.2. Lập trình tạo Table*

*Bảng 2.1.2. Lập trình tạo Table*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Table** | **Tên file, STT dòng khai báo** | **Code** |
| 1 | Area | QuanLiNhaXe.sql (7) | CREATE TABLE dbo.Area  (  Aid INT PRIMARY KEY,  Location CHAR(25) NOT NULL,  Size INT CHECK (Size > 300) NOT NULL,  Opentime TIME(0) NOT NULL,  Closetime TIME(0) NOT NULL,  ); |
| 2 | Employee | QuanLiNhaXe.sql (26) | CREATE TABLE dbo.Employee  (  Eid INT NOT NULL,  Name CHAR(50) NOT NULL,  Birthday DATE NOT NULL,  Address CHAR(255) NOT NULL,  Phone CHAR(12) NOT NULL,  Gender CHAR(12) NOT NULL,  Area INT NOT NULL,  MG\_id INT NOT NULL  REFERENCES dbo.Employee (Eid),  Base\_Salary INT NOT NULL,  PRIMARY KEY (Eid),  FOREIGN KEY (Area) REFERENCES dbo.Area (Aid)  ); |
| 3 | Work | QuanLiNhaXe.sql (26) | CREATE TABLE dbo.Work  (  Eid INT NOT NULL,  NgayLam DATE NOT NULL,  PRIMARY KEY (Eid,NgayLam ),  FOREIGN KEY (Eid) REFERENCES dbo.Employee (Eid)  ); |
| 4 | Salary | QuanLiNhaXe.sql (26) | CREATE TABLE Salary  (  Eid INT  REFERENCES dbo.Employee (Eid) NOT NULL,  Month INT CHECK (Month > 0  AND Month < 13  ) NOT NULL,  Year INT NOT NULL,  Fine INT NOT NULL,  Total INT NOT NULL,  PRIMARY KEY (Eid),  ); |
| 5 | Customer | QuanLiNhaXe.sql (26) | CREATE TABLE dbo.Customer  (  Cid INT NOT NULL,  NgayLamThe DATE NOT NULL,  NgayHetHan DATE NOT NULL,  PRIMARY KEY (Cid),  ); |
| 6 | Ticket | QuanLiNhaXe.sql (26) | CREATE TABLE dbo.Ticket  (  Cid INT NOT NULL,  Eid INT NOT NULL,  NgayGui DATE NOT NULL,  GioVao TIME(0) NOT NULL,  GioRa TIME(0) NOT NULL,  PRIMARY KEY (  Cid,  Eid  ),  FOREIGN KEY (Cid) REFERENCES Customer (Cid),  FOREIGN KEY ( Eid,NgayGui ) REFERENCES dbo.Work ( Eid,NgayLam )  ); |

**1.3. Database Diagram**

****

*Hình 2.1.2: Class Diagram*

**CHƯƠNG 2: TẠO VIEW, STORE PROCEDURE, FUNCTION, TRIGGER TẦNG CƠ SỞ DỮ LIỆU**

**2.1. Tạo các View**

*Bảng 2.2.1. Tạo các View*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
| 1 |  | CREATE VIEW ViewEmployee  AS  SELECT \*  FROM dbo.Employee; |
| 2 |  | CREATE VIEW ViewCustomer  AS  SELECT Customer.Cid,  Customer.NgayLamThe,  Customer.NgayHetHan,  Ticket.Eid,  Ticket.NgayGui,  Ticket.GioVao,  Ticket.GioRa  FROM dbo.Customer  INNER JOIN dbo.Ticket  ON Customer.Cid = Ticket.Cid; |
| 3 |  | CREATE VIEW ViewWork  AS  SELECT \*  FROM dbo.Work; |
| 4 |  | CREATE VIEW ViewSalary  AS  SELECT \*  FROM dbo.Salary; |
| 5 |  | CREATE VIEW ViewTicket  AS  SELECT \*  FROM dbo.Ticket; |
| 6 |  | CREATE VIEW ViewArea  AS  SELECT \*  FROM dbo.Area; |

**2.2. Tạo các Store Procedure**

*2.2.1. Store Procedure cho Employee*

*Bảng 2.2.2. Store Procedure cho Employee*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- thêm mới nhân viên  CREATE PROC InputEmployee  @Eid INT,@Name CHAR(50),@Birthday DATE,  @Address CHAR(255),@Phone NVARCHAR(12),@Gender CHAR(10),@Area INT,@MG\_id INT,@Password CHAR(100),@Base\_Salary INT  AS  BEGIN  INSERT INTO dbo.Employee  (Eid, Name, Birthday, Address, Phone, Gender, Area, MG\_id, Password, Base\_Salary)  VALUES (@Eid, @Name, @Birthday, @Address, @Phone, @Gender, @Area, @MG\_id, @Password, @Base\_Salary);  END; |
|  |  | --- chỉnh sửa nhân viên  CREATE PROC UpdateEmployee  @Eid INT, @Name CHAR(50), @Birthday DATE,@Address CHAR(255),@Phone NVARCHAR(12),@Gender CHAR(10),@Area INT,@MG\_id INT,@Password CHAR(100),@Base\_Salary INT  AS  BEGIN  UPDATE dbo.Employee  SET Name = @Name, Birthday = @Birthday, Address = @Address, Phone = @Phone, Gender = @Gender, Area = @Area, MG\_id = @MG\_id,Password = @Password, Base\_Salary = @Base\_Salary  WHERE Eid = @Eid;  END; |
|  |  | --- xóa nhân viên  CREATE PROC DeleteEmployee (@id INT)  AS  BEGIN  BEGIN TRANSACTION DeleteEmployee;  DELETE dbo.Work  WHERE Eid = @id;  DELETE dbo.Salary  WHERE Eid = @id;  DELETE dbo.Employee  WHERE Eid = @id;  COMMIT TRANSACTION DeleteEmployee;  END; |
|  |  | --Lấy danh sách tất cả nhân viên  CREATE PROC LoadEmployee  AS  BEGIN  SELECT \*  FROM ViewEmployee  END |

*2.2.2. Store Procedure cho Area*

*Bảng 2.2.3. Store Procedure cho Area*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Thêm sửa khu gửi xe  CREATE PROC InputArea  @Aid INT, @Location CHAR(25),@Size INT,  @Opentime TIME(0), @Closetime TIME(0)  AS  BEGIN  INSERT INTO dbo.Area  (  Aid,Location,Size,Opentime,Closetime  )  VALUES  (@Aid, @Location, @Size, @Opentime, @Closetime);  END; |
|  |  | --- Chỉnh sửa khu gửi xe  CREATE PROC UpdateArea  @Aid INT, @Location CHAR(25), @Size INT, @Opentime TIME(0), @Closetime TIME(0)  AS  BEGIN  UPDATE dbo.Area  SET  Location = @Location,  Size = @Size,  Opentime = @Opentime,  Closetime = @Closetime  WHERE Aid = @Aid;  END; |
|  |  | --- Xóa khu gửi xe  CREATE PROC DeleteArea  (@id INT)  AS  BEGIN  BEGIN TRANSACTION DeleteArea;  DELETE dbo.Customer  WHERE Cid = @id;  DELETE dbo.Employee  WHERE Area = @id;  DELETE dbo.Area  WHERE Aid = @id;  COMMIT TRANSACTION DeleteArea;  END; |

*2.2.3. Store Procedure cho Customer*

*Bảng 2.2.4. Store Procedure cho Customer*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- thêm khách hàng  CREATE PROC InputCustomer  @Cid INT, @NgayLamThe DATE, @NgayHetHan DATE  AS  BEGIN  INSERT INTO dbo.Customer  ( Cid, NgayLamThe, NgayHetHan  )  VALUES  (@Cid, @NgayLamThe, @NgayHetHan);  END; |
|  |  | --- Cập nhật lại khách hàng  CREATE PROC UpdateCustomer  @Cid INT, @NgayLamThe DATE, @NgayHetHan DATE  AS  BEGIN  UPDATE dbo.Customer  SET NgayLamThe = @NgayLamThe,  NgayHetHan = @NgayHetHan  WHERE Cid = @Cid;  END; |
|  |  | -- Xóa khách hàng  CREATE PROC DeleteCustomer  (@id INT)  AS  BEGIN  BEGIN TRANSACTION DeleteCustomer;  DELETE dbo.Ticket  WHERE Cid = @id;  DELETE dbo.Customer  WHERE Cid = @id;  COMMIT TRANSACTION DeleteCustomer;  END; |
|  |  | --- Lấy danh sách của tất cả khách hàng  CREATE PROC LoadCustomer  AS  BEGIN  SELECT \*  FROM ViewCustomer  END |

*2.2.4. Store Procedure cho Salary*

*Bảng 2.2.5. Store Procedure cho Salary*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Thêm thông tin về tiền lương  CREATE PROC InputSalary  @Eid INT, @Month INT, @Year INT, @Fine INT, @Total INT  AS  BEGIN  INSERT INTO dbo.Salary  (  Eid, Month,Year, Fine, Total  )  VALUES  (@Eid, @Month, @Year, @Fine, @Total);  END; |
|  |  | --- Cập nhật lại lương  CREATE PROC UpdateSalary  @Eid INT, @Month INT, @Year INT, @Fine INT, @Total INT  AS  BEGIN  UPDATE dbo.Salary  SET Month = @Month,  Year = @Year,  Fine = @Fine,  @Total = @Total  WHERE Eid = @Eid;  END; |
|  |  | --- Xóa lương  CREATE PROC DeleteSalary  (@id INT)  AS  BEGIN  BEGIN TRANSACTION DeleteSalary;  DELETE dbo.Salary  WHERE Eid = @id;  COMMIT TRANSACTION DeleteSalary;  END; |

*2.2.5. Store Procedure cho Ticket*

*Bảng 2.2.6. Store Procedure cho Ticket*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Lấy ra danh sách tất cả thông tin về vé đã gửi  CREATE PROC LoadTicket  AS  BEGIN  SELECT \*  FROM ViewTicket  END |
|  |  | --- thêm vé  CREATE PROC InputTicket  @Cid INT, @Eid INT, @NgayGui DATE, @GioVao TIME, @GioRa Time  AS  BEGIN  INSERT INTO dbo.Ticket  (  Cid, Eid,NgayGui, GioVao, GioRa  )  VALUES  (@Cid, @Eid, @NgayGui, @GioVao, @GioRa);  END; |
|  |  | -- Cập nhật lại vé  CREATE PROC UpdateTicket  @Cid INT, @Eid INT, @NgayGui DATE, @GioVao TIME,@GioRa Time  AS  BEGIN  UPDATE dbo.Ticket  SET Eid = @Eid,  NgayGui = @NgayGui,  GioVao = @GioVao,  GioRa = @GioRa  WHERE Cid = @Cid;  END; |
|  |  | --- xóa vé  CREATE PROC DeleteTicket  (@id INT)  AS  BEGIN  BEGIN TRANSACTION DeleteTicket;  DELETE dbo.Ticket  WHERE Cid = @id;  COMMIT TRANSACTION DeleteTicket;  END;  GO |

*2.2.6. Store Procedure cho Work*

*Bảng 2.2.7. Store Procedure cho Work*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- thêm công việc  CREATE PROC InputWork  @Eid INT, @NgayLam DATE  AS  BEGIN  INSERT INTO dbo.Work  (  Eid, NgayLam  )  VALUES  (@Eid, @NgayLam);  END; |
|  |  | --- Cập nhật công việc  CREATE PROC UpdateWork  @Eid INT, @NgayLam DATE  AS  BEGIN  UPDATE dbo.Work  SET NgayLam = @NgayLam  WHERE Eid = @Eid;  END; |
|  |  | --- Xóa công việc  CREATE PROC DeleteWork  (  @Eid INT,  @NgayLam DATE  )  AS  BEGIN  BEGIN TRANSACTION DeleteWork;  DELETE dbo.Work  WHERE Eid = @Eid  AND NgayLam = @NgayLam;  COMMIT TRANSACTION DeleteWork;  END; |

**2.3. Tạo các Function**

*2.3.1. Function cho Employee*

*Bảng 2.2.8. Function cho Employee*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | -- Trả về danh sách nhân viên theo từng khu vực làm việc  --@id\_area là cụm từ nhập tìm kiếm  CREATE FUNCTION Fn\_searchemployee(@id\_area int)  RETURNS @employee\_list TABLE (  MaNV INT NOT NULL,  TenNV char(50) NOT NULL,  birthday DATE NOT NULL,  address char(255) NOT NULL,  phone char(12) NOT NULL,  gender char(12) NOT NULL,  area INT NOT NULL,  id\_mg INT NOT NULL,  base\_salary INT NOT NULL,  password char(100) NOT NULL  )  AS  BEGIN  INSERT into @employee\_list  select \* FROM dbo.Employee  where Area=@id\_area  RETURN;  END |

*2.3.2. Function cho Ticket*

*Bảng 2.2.9. Function cho Ticket*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- trả về thông tin gửi xe theo khu vực trong một ngày cụ thể  --@ngay là cụm từ tìm kiếm theo ngày nhập, @id\_area là khu vực muốn tìm kiếm  CREATE FUNCTION Fn\_thontinguixe(@ngay date, @id\_area int)  RETURNS @date\_list table(  MaKH INT NOT NULL,  MaNV INT NOT NULL ,  MaNhaXe INT NOT NULL,  NgayGui DATE NOT NULL,  GioVao time(0) NOT NULL,  GioRa time(0) NOT NULL  )  AS  BEGIN  INSERT INTO @date\_list  SELECT q.Cid,q.Eid,k.Area,q.NgayGui,q.GioVao,q.GioRa FROM  (SELECT \* FROM dbo.Ticket WHERE NgayGui = @ngay) q INNER JOIN dbo.Employee AS k ON k.Eid = q.Eid  WHERE @id\_area = k.Area  RETURN  END |
|  |  | - Hàm trả về tổng số lượng xe giữ được trong một ngày cụ thể  ---Trong đó @ngay là cụm từ dùng dể tìm kiếm theo ngày  CREATE FUNCTION Fn\_SoLuongXeTrongNgay(@ngay date)  RETURNS int  as  begin  declare @dem int  select @dem = count(\*)  from dbo.Ticket  where NgayGui = @ngay  return @dem  END |

*2.3.3. Function cho Work*

*Bảng 2.2.10. Function cho Work*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --Trả về id lớn nhất trong bảng Work  CREATE FUNCTION Fn\_GetIDFromDB()  RETURNS INT  AS  BEGIN  DECLARE @id INT  SELECT @id = MAX(Eid) FROM dbo.Work  IF(@id IS NULL)  RETURN 0  RETURN @id  END |

*2.3.3. Function cho Salary*

*Bảng 2.2.10. Function cho Salary*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Trả về total lương lớn nhất hiện có  CREATE FUNCTION Max\_Salary()  RETURNS INT  AS  BEGIN  DECLARE @Luong INT  SELECT @Luong = MAX(Total) FROM dbo.Salary  RETURN @Luong  END |

**2.4. Tạo các Trigger**

*2.4.1. Trigger cho Employee*

*Bảng 2.2.11. Trigger cho Employee*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Trigger khi cập nhật employee  \ |
|  |  | --Trigger khi DELETE Employee  CREATE TRIGGER T\_DeleteEmployee  ON dbo.Employee  FOR DELETE  AS  BEGIN  DECLARE @EmployeeID INT  DECLARE @Area INT  DECLARE @ManagerID INT  DECLARE @COUNT INT  SELECT @EmployeeID = Eid,@Area = Area,@ManagerID = MG\_id FROM Deleted  --Neu nhan vien la quan ly  IF (@ManagerID = @EmployeeID)  BEGIN  --Kiem tra so nhan vien đang quan ly  SELECT @COUNT = COUNT(Eid) FROM dbo.Employee WHERE @EmployeeID = MG\_id  IF @COUNT <> 1  BEGIN  RAISERROR('Chi duoc xoa quan ly khong phu trach nhan vien nao !',16,1)  ROLLBACK  RETURN  END  --Kiem tra khu vuc con quan ly khac khong  SET @COUNT = 0  SELECT @COUNT = COUNT(MG\_id) FROM dbo.Employee WHERE Area = @Area  IF @COUNT = 1  BEGIN  RAISERROR('Khu vuc khong con quan ly, Khong the xoa quan ly hien tai !',16,1)  ROLLBACK  RETURN  END  END  --Nhan vien binh thuong  ELSE  BEGIN  SELECT @COUNT = COUNT(Eid) FROM dbo.Employee WHERE Area = @Area  IF @COUNT = 1  BEGIN  RAISERROR('Khu vuc khong con nhan vien, Khong the xoa nhan vien hien tai !',16,1)  ROLLBACK  RETURN  END  END  BEGIN TRANSACTION  --Delete user  DECLARE @DropUser NVARCHAR(MAX)  SET @DropUser = N'DROP USER' + @EmployeeID  EXECUTE @DropUser  --Delete Login  DECLARE @DropLogin NVARCHAR(MAX)  SET @DropLogin = N'DROP LOGIN' + @EmployeeID  EXECUTE @DropLogin  IF (@@ERROR<>0)  BEGIN  RAISERROR(N'FOUND ERROR DELETE',16,1)  ROLLBACK TRANSACTION  RETURN  END  COMMIT  END |
|  |  | --Trigger khi INSERT Employee  CREATE TRIGGER T\_InsertEmploy  ON dbo.Employee  FOR INSERT  AS  BEGIN  DECLARE @EmployeeID INT  DECLARE @Password CHAR(100)  DECLARE @ManagerID INT  DECLARE @Age INT  DECLARE @Salary INT  SELECT @EmployeeID = Eid,@Password = Password,@ManagerID = MG\_id, @Age = YEAR(GETDATE()-YEAR(Birthday)),@Salary = Base\_Salary FROM Inserted    -----Kiem tra nhan vien phu hop tuoi lao dong  IF (@Age < 18 OR @Age > 60)  BEGIN  RAISERROR('Nhan vien phai trong do tuoi lao dong',16,1)  ROLLBACK  RETURN  END  -----Kiem tra luong co ban cua nhan vien  IF (@Salary < 11000000 AND @ManagerID = @EmployeeID)  BEGIN  RAISERROR('Nhan vien quan ly phai co luong co ban tu 11.000.000 VND',16,1)  ROLLBACK  RETURN  END  ELSE IF (@Salary < 8000000)  BEGIN  RAISERROR('Nhan vien phai co luong co ban trên 8.000.000 VND',16,1)  ROLLBACK  RETURN  END  BEGIN TRANSACTION  --Login  DECLARE @CreateLogin NVARCHAR(MAX)  SET @CreateLogin = 'CREATE LOGIN [' + @EmployeeID + '] WITH PASSWORD = ''' + @Password + '''' + ', DEFAULT\_DATABASE=[QuanLiNhaXe]'  EXECUTE @CreateLogin  --User  DECLARE @CreateUser NVARCHAR(MAX)  SET @CreateUser = 'CREATE USER [' + @EmployeeID + '] FOR LOGIN ['+ @EmployeeID + ']'  EXECUTE @CreateUser  --Role  IF @EmployeeID = @ManagerID  BEGIN  EXECUTE sp\_addrolemember @rolename = 'manager',  @membername = @EmployeeID  END  ELSE  BEGIN  EXECUTE sp\_addrolemember @rolename = 'employee',  @membername = @EmployeeID  END  IF (@@ERROR<>0)  BEGIN  RAISERROR(N'FOUND ERROR CREATE',16,1)  ROLLBACK TRANSACTION  RETURN  END  COMMIT  END |

*2.4.2. Trigger cho Salary*

*Bảng 2.2.12. Trigger cho Salary*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Trigger khi insert tiền lương cho nhân viên  CREATE TRIGGER T\_InsertSalary  ON dbo.Salary  FOR INSERT  AS  BEGIN  DECLARE @Id INT  DECLARE @TotalSalary INT  DECLARE @BasicSalary INT  DECLARE @Fine INT  SELECT @BasicSalary = Base\_Salary  FROM dbo.Employee  SELECT @TotalSalary = Total, @Fine = Fine, @Id = Eid FROM Inserted  IF NOT EXISTS (SELECT \* FROM dbo.Employee WHERE Eid = @Id)  BEGIN  RAISERROR('Id nhan vien khong ton tai',16,1)  ROLLBACK  RETURN  END  IF (@Fine>@BasicSalary)  BEGIN  RAISERROR('Tien phat khong the vuot qua luong co ban',16,1)  ROLLBACK  RETURN  END  IF (@TotalSalary <> @BasicSalary - @Fine)  BEGIN  RAISERROR('Tinh luong sai, vui long tinh lai (Luong = Luong co ban + tien phat)',16,1)  ROLLBACK  RETURN  END  END |
|  |  | --- trigger khi cập nhật lại lương  CREATE TRIGGER T\_UpdateSalary  ON dbo.Salary  FOR UPDATE  AS  BEGIN  DECLARE @TotalSalary INT  DECLARE @BasicSalary INT  DECLARE @Fine INT  SELECT @BasicSalary = Base\_Salary FROM dbo.Employee  SELECT @TotalSalary = Total, @Fine = Fine FROM Inserted  IF (@Fine>@BasicSalary)  BEGIN  RAISERROR('Tien phat khong the vuot qua luong co ban',16,1)  ROLLBACK  RETURN  END  IF (@TotalSalary <> @BasicSalary - @Fine)  BEGIN  RAISERROR('Tinh luong sai, vui long tinh lai (Luong = Luong co ban + tien phat)',16,1)  ROLLBACK  RETURN  END  END |

` *2.4.3. Trigger cho Customer*

*Bảng 2.2.13. Trigger cho Customer*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- trigger khi thêm một khách hàng mới  CREATE TRIGGER T\_InsertCustomer  ON dbo.Customer  FOR INSERT  AS  BEGIN  DECLARE @Ngaylamthe DATE  DECLARE @Id INT  DECLARE @Ngayhethan DATE  SELECT @Id= Cid, @Ngaylamthe=NgayLamThe, @Ngayhethan = NgayHetHan FROM Inserted  IF EXISTS (SELECT \* FROM dbo.Customer WHERE @Id = Cid)  BEGIN  RAISERROR('Id khach hang da ton tai',16,1)  ROLLBACK  RETURN  END  IF (@Ngayhethan < @Ngaylamthe)  BEGIN  RAISERROR('Ngay het han sai, vui long nhap lai',16,1)  ROLLBACK  RETURN  END  END |

*2.4.4. Trigger cho Ticket*

*Bảng 2.2.14. Trigger cho Ticket*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Trigger khi insert ticket  CREATE TRIGGER T\_InsertTicket  ON dbo.Ticket  FOR INSERT  AS  BEGIN  DECLARE @Area INT  DECLARE @Slot INT  DECLARE @Eid INT  DECLARE @Cid INT  DECLARE @NgayGui DATE  DECLARE @GioVao TIME(0)  DECLARE @GioRa TIME(0)  DECLARE @CountCar INT  DECLARE @Opentime TIME(0)  DECLARE @Closetime TIME(0)  SELECT @Eid = Eid, @NgayGui = NgayGui, @GioVao = GioVao, @GioRa = GioRa, @Cid = Cid FROM Inserted  SELECT @Area = Area FROM dbo.Employee WHERE @Eid = Employee.Eid  SELECT @CountCar = COUNT(\*) FROM (SELECT \* FROM dbo.Ticket WHERE NgayGui = @NgayGui) k INNER JOIN (SELECT Eid FROM dbo.Employee WHERE @Area = Employee.Area) q ON q.Eid = k.Eid  SELECT @Slot = Size, @Opentime=Opentime,@Closetime =Closetime FROM dbo.Area WHERE Area.Aid = @Area  IF (@GioVao < @Opentime OR @GioVao > @Closetime)  BEGIN  RAISERROR('Khu vuc chua den gio hoat dong, quy khach vui long thong cam',16,1)  ROLLBACK  RETURN  END  IF (@CountCar = @Slot)  BEGIN  RAISERROR('Khu nay da het cho, quy khach vui long qua khu khac',16,1)  ROLLBACK  RETURN  END  IF (@GioVao <= @GioRa)  BEGIN  RAISERROR('Nhap sai thoi gian ra vao vui long thu lai',16,1)  ROLLBACK  RETURN  END  IF EXISTS (SELECT \* FROM dbo.Ticket WHERE Cid = @Cid AND NgayGui = @NgayGui)  BEGIN  IF (@GioVao <= (SELECT MAX(GioRa) FROM dbo.Ticket WHERE Cid = @Cid AND NgayGui = @NgayGui))  BEGIN  RAISERROR('Xe da o trong bai xe, vui long thu lai',16,1)  ROLLBACK  RETURN  END  END  END |
|  |  | --- Trigger khi cập nhật lại ticket  CREATE TRIGGER T\_UpdateTicket  ON dbo.Ticket  FOR UPDATE  AS  BEGIN  DECLARE @Area INT  DECLARE @Eid INT  DECLARE @Slot INT  DECLARE @NgayGui DATE  DECLARE @GioVao TIME(0)  DECLARE @GioRa TIME(0)  DECLARE @CountCar INT  DECLARE @Opentime TIME(0)  DECLARE @Closetime TIME(0)  DECLARE @Area\_old INT  DECLARE @Eid\_old INT  DECLARE @NgayGui\_old DATE  SELECT @Eid = Eid, @NgayGui = NgayGui, @GioVao = GioVao, @GioRa = GioRa FROM Inserted  SELECT @Area = Area FROM dbo.Employee WHERE @Eid = Employee.Eid  SELECT @CountCar = COUNT(\*) FROM (SELECT \* FROM dbo.Ticket WHERE NgayGui = @NgayGui) k INNER JOIN (SELECT Eid FROM dbo.Employee WHERE @Area = Employee.Area) q ON q.Eid = k.Eid  SELECT @Slot = Size, @Opentime=Opentime,@Closetime =Closetime FROM dbo.Area WHERE Area.Aid = @Area  SELECT @Eid\_old = Eid, @NgayGui\_old = NgayGui FROM Deleted  SELECT @Area\_old = Area FROM dbo.Employee WHERE @Eid\_old = Employee.Eid  --Thay doi ngay gui xe  IF @NgayGui <> @NgayGui\_old  BEGIN  RAISERROR('Khong the thay doi ngay gui',16,1)  ROLLBACK  RETURN  END  IF @Area <> @Area\_old  BEGIN  IF (@GioVao < @Opentime OR @GioVao > @Closetime)  BEGIN  RAISERROR('Khu vuc chua den gio hoat dong, quy khach vui long thong cam',16,1)  ROLLBACK  RETURN  END  IF (@CountCar = @Slot)  BEGIN  RAISERROR('Khu nay da het cho, quy khach vui long qua khu khac',16,1)  ROLLBACK  RETURN  END  END  IF (@GioVao <= @GioRa)  BEGIN  RAISERROR('Nhap sai thoi gian ra vao vui long thu lai',16,1)  ROLLBACK  RETURN  END  END |

*2.4.5. Trigger cho Work*

*Bảng 2.2.15. Trigger cho Work*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --- Trigger khi cập nhật hoặc xóa công việc  CREATE TRIGGER T\_UpdateWork  ON dbo.Work  FOR UPDATE,DELETE  AS  BEGIN  DECLARE @Area INT  DECLARE @Date\_old DATE  DECLARE @Id\_old INT  SELECT @Id\_old = Eid, @Date\_old = NgayLam FROM Deleted  SELECT @Area = Area FROM dbo.Employee WHERE @Id\_old = Eid  --Kiem tra con ai lam chung khu khong ?  IF NOT EXISTS (SELECT COUNT(\*)  FROM (SELECT Eid,Area  FROM dbo.Employee  WHERE Area = @Area AND Eid <> MG\_id) q INNER JOIN (SELECT \* FROM dbo.Work WHERE NgayLam = @Date\_old) k ON q.Eid = k.Eid)  BEGIN  RAISERROR('Thieu nguoi lam nen khong the thay doi hoac xoa',16,1)  ROLLBACK  RETURN  END  END  GO |
|  |  |  |

**2.5. Tạo các Role, phân quyền**

*2.5.1. Tạo role Manager và phân quyền*

*Bảng 2.2.16. Tạo role Manager và phân quyền*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --tạo quyền  CREATE ROLE manager |
|  |  | -- Cấp quyền các trên Store procedure cho manager  GRANT EXECUTE, ALTER ON dbo.LoadArea TO manager  GRANT EXECUTE, ALTER ON dbo.LoadCustomer TO manager  GRANT EXECUTE, ALTER ON dbo.LoadEmployee TO manager  GRANT EXECUTE, ALTER ON dbo.LoadSalary TO manager  GRANT EXECUTE, ALTER ON dbo.LoadTicket TO manager  GRANT EXECUTE, ALTER ON dbo.LoadWork TO manager  GRANT EXECUTE, ALTER ON dbo.CountEmployee TO manager  GRANT EXECUTE, ALTER ON dbo.CountCustomer TO manager  GRANT EXECUTE, ALTER ON dbo.DeleteCustomer TO manager  GRANT EXECUTE, ALTER ON dbo.DeleteEmployee TO manager  GRANT EXECUTE, ALTER ON dbo.DeleteTicket TO manager  GRANT EXECUTE, ALTER ON dbo.DeleteWork TO manager  GRANT EXECUTE, ALTER ON dbo.InputCustomer TO manager  GRANT EXECUTE, ALTER ON dbo.InputEmployee TO manager  GRANT EXECUTE, ALTER ON dbo.InputWork TO manager  GRANT EXECUTE, ALTER ON dbo.InputTicket TO manager  GRANT EXECUTE, ALTER ON dbo.UpdateSalary TO manager  GRANT EXECUTE, ALTER ON dbo.UpdateCustomer TO manager  GRANT EXECUTE, ALTER ON dbo.UpdateTicket TO manager  GRANT EXECUTE, ALTER ON dbo.UpdateWork TO manager  GRANT EXECUTE, ALTER ON dbo.UpdateEmployee TO manager  --Cấp quyền trên Function cho manager  GRANT SELECT ON dbo.Fn\_searchemployee TO manager  GRANT SELECT ON dbo.Fn\_thontinguixe TO manager  GRANT EXECUTE ON dbo.Fn\_SoLuongXeTrongNgay TO manager  GRANT EXECUTE ON dbo.Max\_Salary TO manager  GRANT EXECUTE ON dbo.Fn\_GetIDFromDB TO manager |

*2.5.2. Tạo role Employee và phân quyền*

*Bảng 2.2.17. Tạo role Employee và phân quyền*

|  |  |  |
| --- | --- | --- |
| **STT** | **STT Tên file, STT dòng khai báo Code** | **Code** |
|  |  | --tạo quyền  CREATE ROLE employee |
|  |  | -- Cấp quyền các trên Store procedure cho employee  GRANT EXECUTE ON dbo.LoadArea TO employee  GRANT EXECUTE ON dbo.LoadCustomer TO employee  GRANT EXECUTE ON dbo.LoadEmployee TO employee  GRANT EXECUTE ON dbo.LoadSalary TO employee  GRANT EXECUTE ON dbo.LoadTicket TO employee  GRANT EXECUTE ON dbo.LoadWork TO employee  GRANT EXECUTE ON dbo.CountCustomer TO employee  GRANT EXECUTE ON dbo.DeleteCustomer TO employee  GRANT EXECUTE ON dbo.InputCustomer TO employee  GRANT EXECUTE ON dbo.UpdateTicket TO employee  GRANT EXECUTE ON dbo.InputTicket TO employee  GRANT EXECUTE ON dbo.UpdateEmployee TO employee  -- Cấp quyền các trên Function cho employee  GRANT SELECT ON dbo.Fn\_thontinguixe TO employee  GRANT EXECUTE ON dbo.Fn\_SoLuongXeTrongNgay TO employee  GRANT EXECUTE ON dbo.Fn\_GetIDFromDB TO employee |

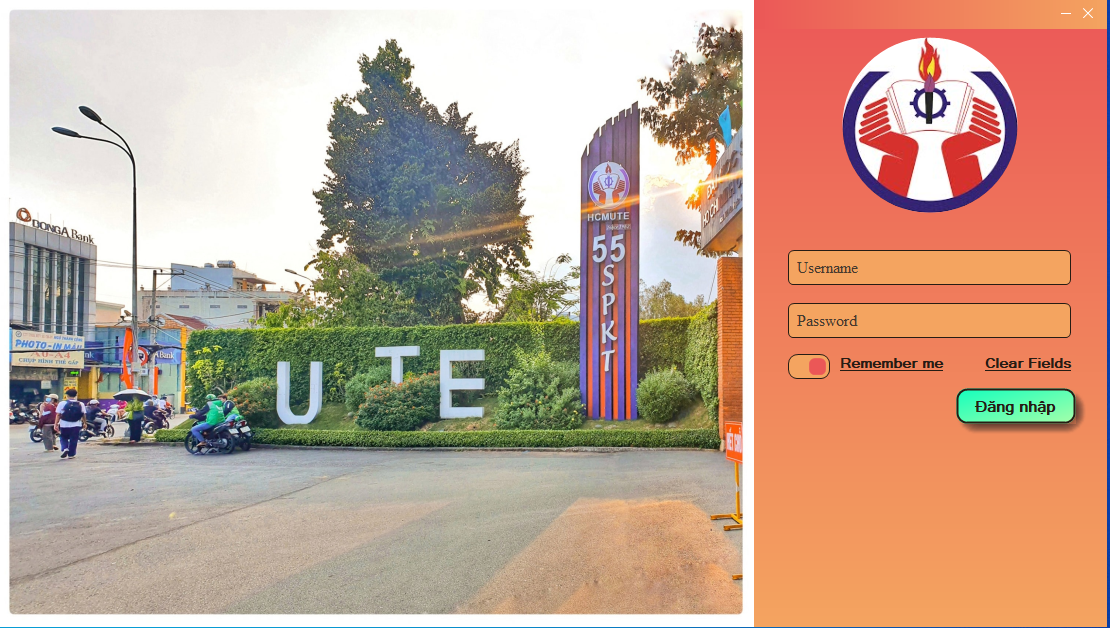
**CHƯƠNG 3: THIẾT KẾ GIAO DIỆN**

Giao diện của phần mềm quản lý nhà xe được thiết kế đơn giản, thân thiện để người dùng dễ dàng sử dụng. Phần giao diện này được tổ chức thành 8 form bao gồm : Form Login, Form MainMenu, Form Sticket, Form Customer, Form Salary, Form Staff, Form Work, Form Area.

**3.1. Form Login**

Giao diện đăng nhập dùng để nhân viên và quản lí đăng nhập vào phần mềm. Giao diện phân biệt người đăng nhập là quản lí hay nhân viên.

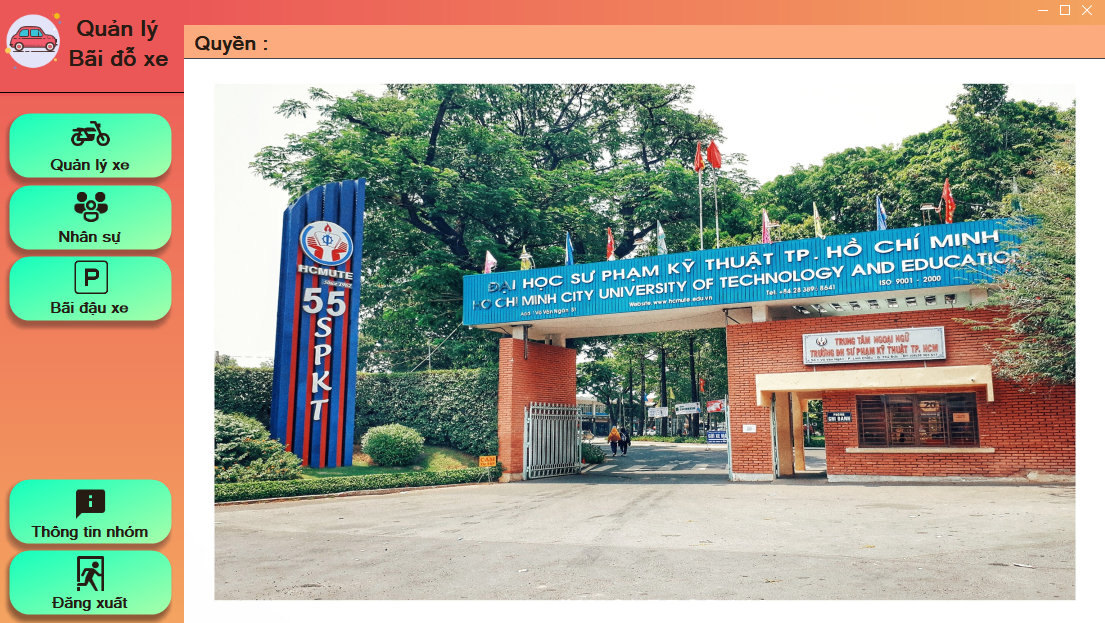
*Hình 2.3.1. Thiết kế Form Login*



**3.2. Form MainMenu**

Giao diện main menu là trang hiển thị tổng quan nhất về các chức năng của phần mềm. Trang giúp chúng ta truy cập đến các chức năng của phần mềm như quản lí nhân viên, quản lí vé xe, quản lí khu giữ xe.

*Hình 2.3.2. Thiết kế Form MainMenu*

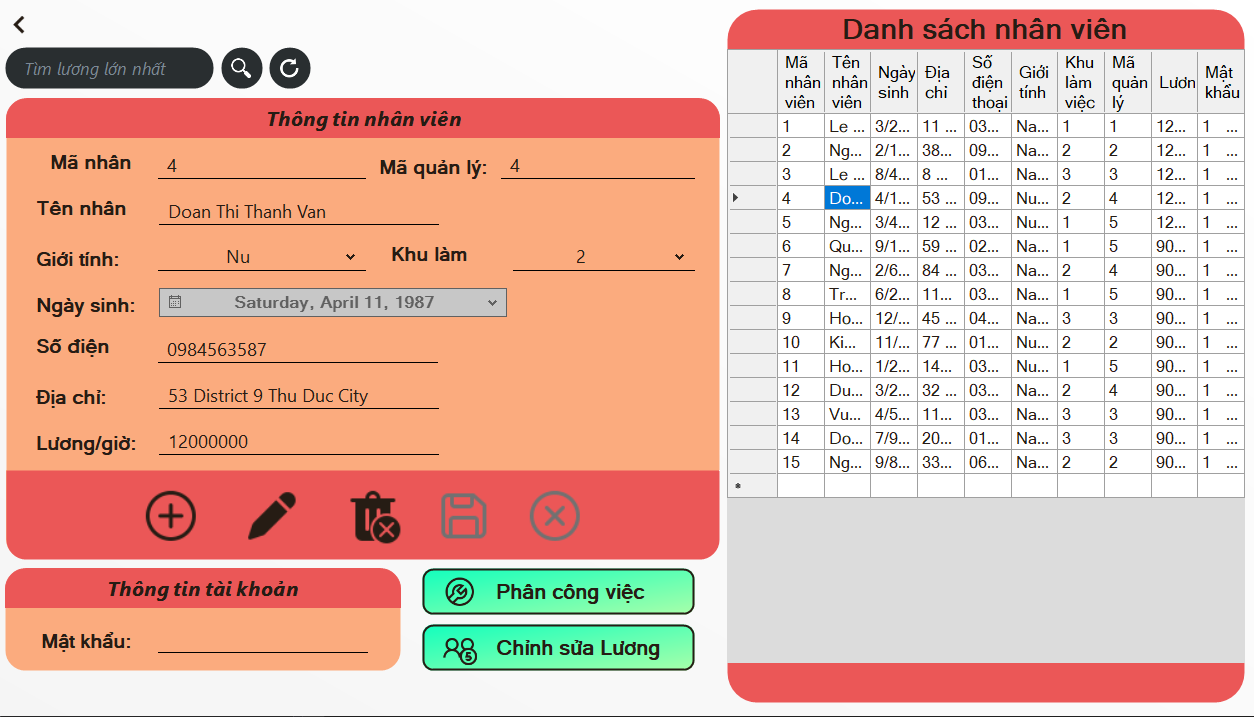


**3.3. Form Staff**

Giao diện hiển thị danh sách nhân viên và thông tin chi tiết về nhân viên bao gồm: mã nhân viên, họ tên nhân viên, mã người quản lý, ngày sinh, số điện thoại, lương cơ bản của nhân viên đó. Giao diện được cấp quyền :

* Nhân viên có thể thực hiện các chức năng chỉnh sửa thông tin cá nhân.
* Quản lý được thực hiện tất cả chức năng như thêm nhân viên, xóa nhân viên và chỉnh sửa nhân viên.

*Hình 2.3.3. Thiết kế Form Staff*

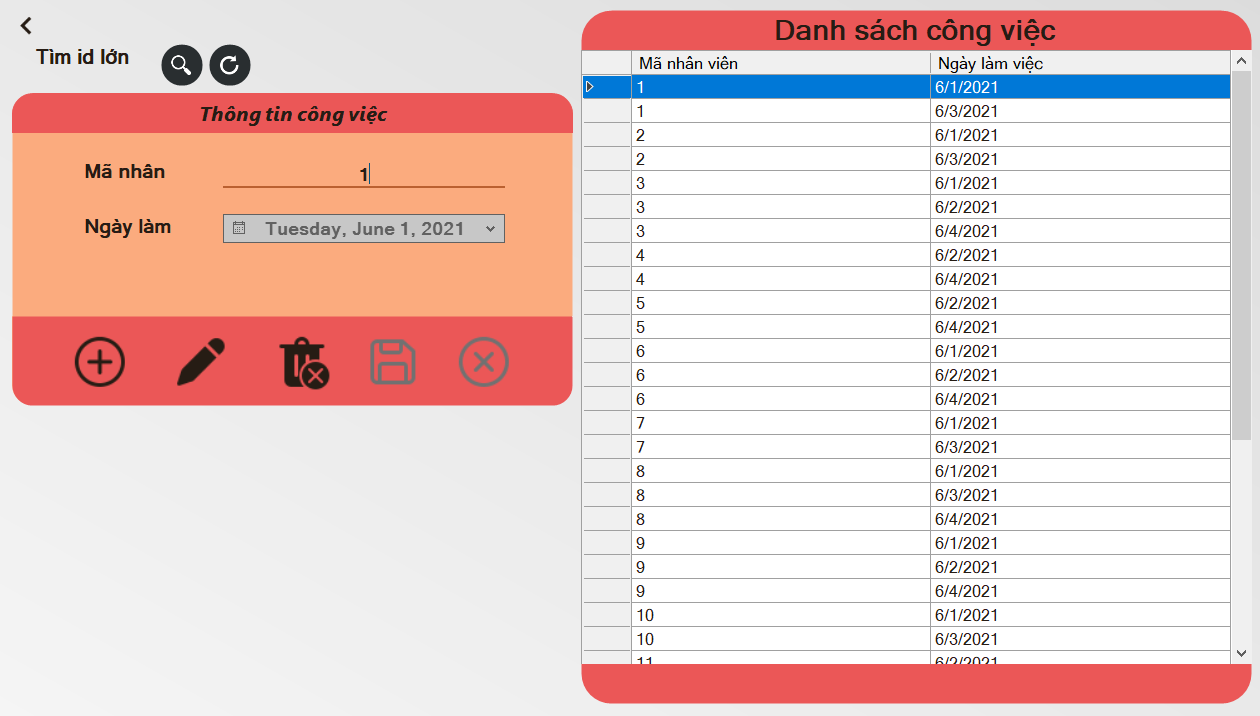
****

**3.4. Form Work**

Giao diện dùng để hiện thị danh sách công việc và thông tin chi tiết công việc bao gồm : mã nhân viên và ngày được phân công làm việc. Giao diện được cấp quyền :

* Nhân viên dựa vào danh sách công việc để biết được thời gian làm việc.
* Quản lý có quyền thêm, xóa, sửa công việc của nhân viên.

*Hình 2.3.4. Thiết kế Form Work*

****

**3.5. Form Salary**

Giao diện dùng để hiển thị bảng lương của nhân viên làm việc trong nhà xe. Mỗi dòng trên bảng lương biểu thị lương của một nhân viên trong một tháng. Lương của nhân viên được tính dựa vào thời gian họ làm việc. Giao diện cấp quyền :

* Quản lí được quyền thực hiện chức năng thêm, xóa, chỉnh sửa lương nhân viên.
* Nhân viên chỉ được quyền xem bảng lương.

*Hình 2.3.5. Thiết kế Form Salary*

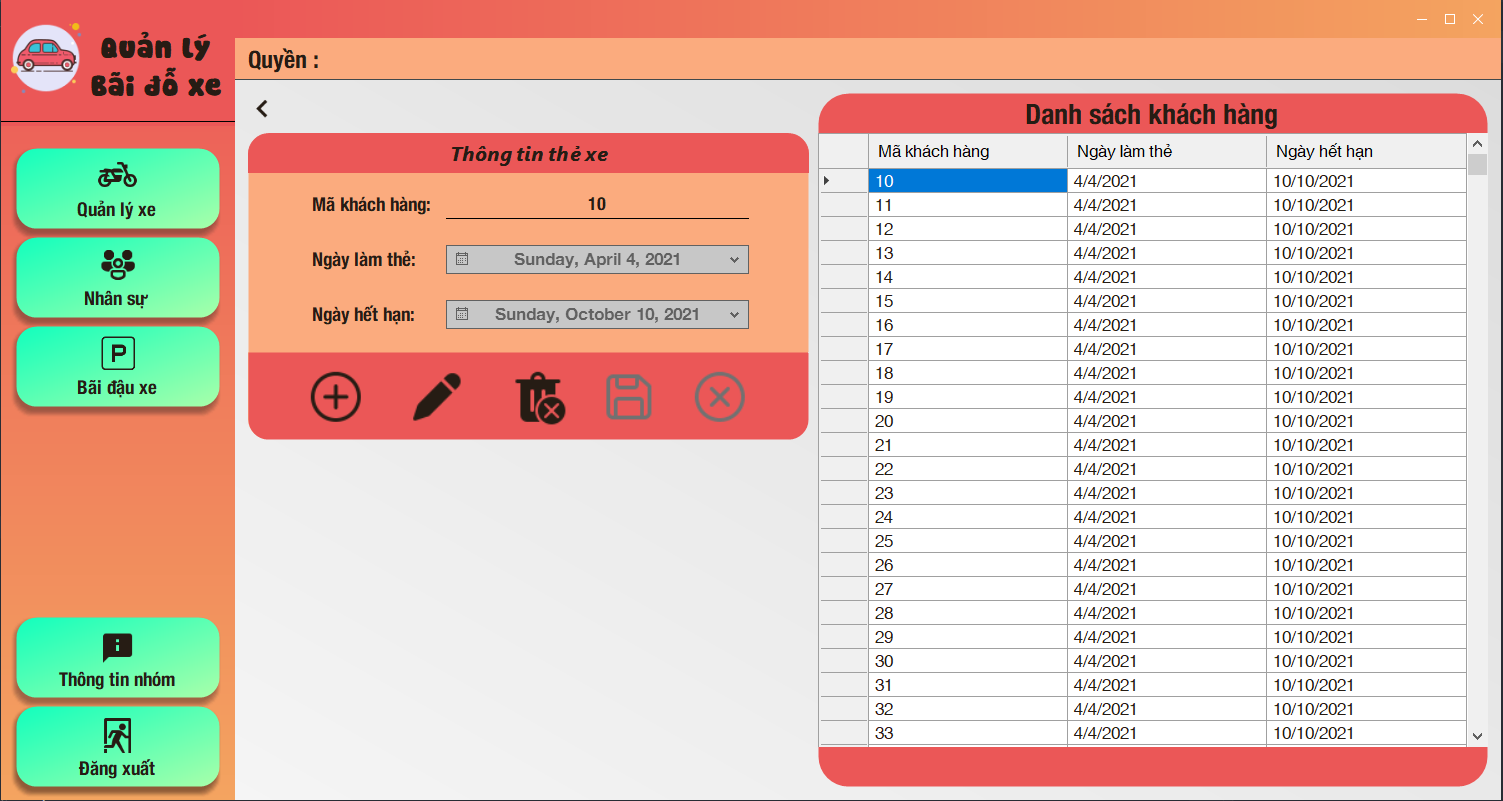


**3.6. Form Customer**

Form dùng để hiển thị thông tin khách hàng và thông tin về vé xe bao gồm : mã khách hàng, ngày làm thẻ và ngày hết hạn. . Giao diện cấp quyền :

* Quản lí được quyền thực hiện chức năng thêm, xóa, chỉnh sửa thông tin khách hàng.
* Nhân viên được quyền thực hiện chức năng thêm và xóa thông tin khách hàng.

*Hình 2.3.6. Thiết kế Form Customer*



**3.7. Form Sticket**

Form dùng để hiển thị thông tin của việc gửi xe bao gồm : mã khách hàng, mã nhân viên, ngày gửi, giờ ra và giờ vào. . Giao diện cấp quyền :

* Quản lí được quyền thực hiện chức năng thêm, xóa, chỉnh sửa thông tin vé xe.
* Nhân viên được quyền thực hiện chức năng thêm và xóa thông tin vé xe.

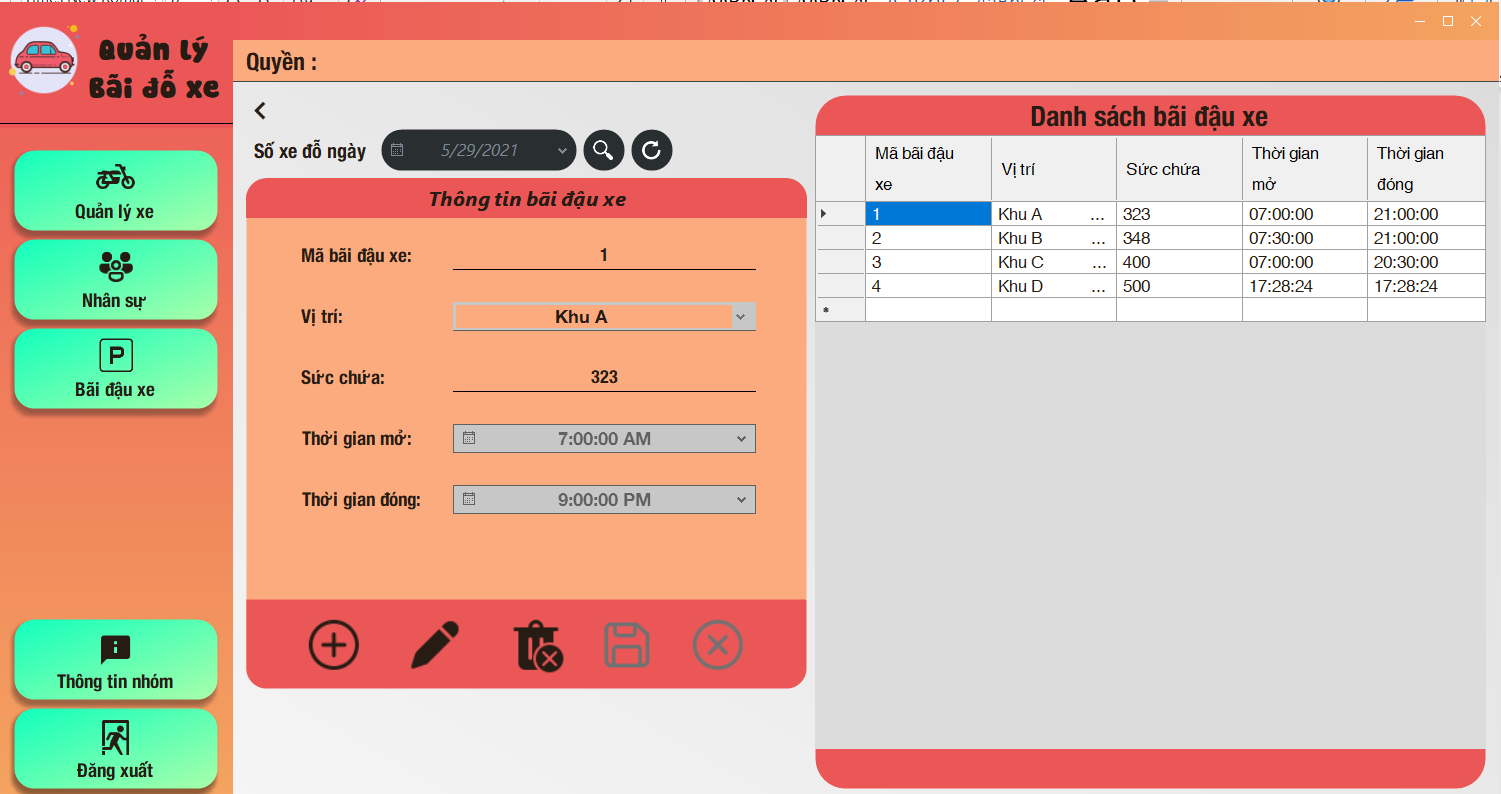
*Hình 2.3.7. Thiết kế Form Sticket*



**3.8 Form Area**

Form dùng để hiển thị thông tin của các nhà xe bao gồm: mã khu nhà xe, vị trí, sức chưa, thời gian mở cửa và thời gian đóng cửa. Giao diện chỉ cho phép quản lí và nhân viên được xem thông tin nhà xe.

*Hình 2.3.8. Thiết kế Form Area*



**CHƯƠNG 4: KẾT NỐI VỚI CƠ SỞ DỮ LIỆ**

**4.1. Chức năng của các lớp phương thức**

*Bảng 2.4.1. Đặt tả các lớp chức năng*

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Sinh viên phụ trách** |
| 1 | DataProvider | - Cung cấp các phương thức kết nối đến cơ sở dữ liệu, thực thi query và nhận về kết quả. | Minh Đức  Minh Nhựt |
| 2 | AreaDal | - Lớp Dal thực hiện chức năng xử lý trên cơ sở dữ liệu Area như: lấy khu giữ xe bằng mã nhà xe, lấy danh sách khu gửi xe, thêm khu gửi xe, cập nhật khu gửi xe, xóa khu gửi xe. | Minh Đức  Minh Nhựt |
| 3 | CustomerDAl | - Lớp Dal thực hiện chức năng xử lý trên cơ sở dữ liệu Customer như: lấy khách hàng bằng mã khách hàng, lấy danh sách khách hàng, thêm khách hàng, cập nhật khách hàng, xóa nhân viên. | Minh Đức  Minh Nhựt |
| 4 | EmployeeDal | - Lớp Dal thực hiện chức năng xử lý trên cơ sở dữ liệu Emlpoyee như: lấy nhân viên bằng mã nhân viên, lấy danh sách nhân viên, thêm nhân viên, cập nhật nhân viên, xóa nhân viên. | Minh Đức  Minh Nhựt |
| 5 | SalaryDal | - Lớp Dal thực hiện chức năng xử lý trên cơ sở dữ liệu Salary như: lấy ID salary kế tiếp, lấy bản lương theo mã nhân viên, tháng, năm, thực hiện tính toán bảng lương cho nhân viên, lấy danh sách các lương nhân viên theo tháng, năm. | Thanh Vân  Minh Hoàng |
| 6 | TicketDal | - Lớp Dal thực hiện chức năng xử lý trên cơ sở dữ liệu Ticket như: lấy danh sách vé gửi xe đã gửi, lấy danh sách vé bằng mã khu giữ xe, lọc vé, thêm, xóa, cập nhật thêm cho vé. | Thanh Vân  Minh Hoàng |
| 7 | WorkDal | - Lớp Dal thực hiện chức năng xử lý trên cơ sở dữ liệu Work như: lấy danh sách làm việc theo ngày, theo mã số nhân viên, thêm, cập nhật điểm danh. | Xuân Thanh |

**4.2. Đặt tả các phương thức trong lớp**

*Bảng 2.4.2. Đặc tả các phương thức trong lớp AreaDal*

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Tên file, STT dòng khai báo** |
| 1 | load\_area()  Input: None  Output: List  List<Area> | - Lấy danh sách khu gửi xe từ cơ sở dữ liệu. | AreaDal.cs |
| *2* | insert\_area(AreaPublic areaPublic)  Input: area  Output: bool | - Thêm khu gửi xe vào cơ sở dữ liệu. | AreaDal.cs |
| *3* | update\_area(AreaPublic areaPublic)  Input: area  Output: bool | - Cập nhật khu gửi xe “area” vào cơ sở dữ liệu | AreaDal.cs |
| *4* | delete\_area(AreaPublic areaPublic)  Input: area  Output: bool | - Xóa khu vực trong cơ sở dữ liệu.  - Xóa các work mà làm việc tại khu vực này.  - Xóa nhân viên làm việc tại các khu vực này. | AreaDal.cs |

*Bảng 2.4.3. Đặc tả các phương thức trong lớp SalaryDal*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *1* | load\_salary()  Input: None  Output: List  List<Salary> | - Lấy danh sách tiền lương từ cơ sở dữ liệu. | *SalaryDal.cs* |
| *2* | insert\_salary(SalaryPublic salaryPublic)  Input: salary  Output: bool | - Thêm tiền lương vào cơ sở dữ liệu. | SalaryDal.cs |
| *3* | update\_salary(SalaryPublic salaryPublic)  Input: salary  Output: bool | - Cập nhật tiền lương “salary” vào cơ sở dữ liệu | SalaryDal.cs |
| *4* | delete\_salary(SalaryPublic salaryPublic)  Input: salary  Output: bool | - Xóa tiền lương trong cơ sở dữ liệu. | SalaryDal.cs |
| *5* | max\_salary()  Input: none  Output: int hoặc null | * *Lấy ra tiền lương cao nhất trong cơ sở dữ liệu* | SalaryDal*.cs* |

*Bảng 2.4.4. Đặc tả các phương thức trong lớp CustomerDal*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *1* | load\_customer()  Input: None  Output: List  List<Customer> | - Lấy danh sách khách hàng từ cơ sở dữ liệu. | CustomerDal.cs |
| *2* | insert\_customer(CustomerPublic customerPublic)  Input: customer  Output: bool | - Thêm khách hàng vào cơ sở dữ liệu. | CustomerDal.cs |
| *3* | update\_customer(CustomerPublic customerPublic)  Input: customer  Output: bool | - Cập nhật khách hàng “salary” vào cơ sở dữ liệu | CustomerDal.cs |
| *4* | delete\_customer(CustomerPublic customerPublic)  Input: customer  Output: bool | - Xóa khách hàng trong cơ sở dữ liệu. | CustomerDal.cs |
| *5* | count\_customer()  Input: none  Output: int hoặc null | * *Đếm số lượng khách hàng trong cơ sở dữ liệu* | CustomerDal.cs |

*Bảng 2.4.6. Đặc tả các phương thức trong lớp EmployeeDal*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *1* | load\_employee()  Input: None  Output: List  List<Employee> | - Lấy danh sách nhan viên từ cơ sở dữ liệu. | EmployeeDal.cs |
| *2* | insert\_employee(EmployeePublic employeePublic)  Input: Employee  Output: bool | - Thêm khách hàng vào cơ sở dữ liệu. | EmployeeDal.cs |
| *3* | update\_employee(EmployeePublic employeePublic)  Input: Employee  Output: bool | - Cập nhật nhân viên “employee” vào cơ sở dữ liệu | EmployeeDal.cs |
| *4* | delete\_employee(EmployeePublic employeePublic)  Input: Employee  Output: bool | - Xóa nhân viên trong cơ sở dữ liệu.  - Xóa các work mà nhân viên này làm.  - Xóa lương của nhân viên trong cơ sở dữ liệu. | EmployeeDal.cs |
| *5* | count\_employee()  Input: none  Output: int hoặc null | * *Đếm tổng số nhân viên trong cơ sở dữ liệu* | EmployeeDal.cs |
| *6* | find\_employee(int areaId)  Input: area ID  Output: List<Employee> | * Lấy danh sách nhân viên có mã khu làm việc bằng “ID area ” từ cơ sở dữ liệu. | EmployeeDal.cs |

*Bảng 2.4.7. Đặc tả các phương thức trong lớp TicketDal*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *1* | load\_ticket()  Input: None  Output: List  List<Ticket> | - Lấy danh các thông tin vé xe từ cơ sở dữ liệu. | TicketDal.cs |
| *2* | insert\_ticket(TicketPublic ticketPublic)  Input: Employee  Output: bool | - Thêm thông tin vé xe vào cơ sở dữ liệu. | TicketDal.cs |
| *3* | update\_ticket(TicketPublic ticketPublic)  Input: Employee  Output: bool | - Cập nhật thông tin vé xe “ticket” vào cơ sở dữ liệu | TicketDal.cs |
| *4* | delete\_ticket(TicketPublic ticketPublic)  Input: Employee  Output: bool | - Xóa thông tin vé xe trong cơ sở dữ liệu. | TicketDal.cs |
| *5* | find\_ticket\_areaId(DateTime date, int areaId)  Input: date, Area\_ID  Output: List<Ticket> | - Lấy danh sách các thông tin vé xe đang có ngày = date và khu vực = Area\_ID | TicketDal.cs |
| *6* | count\_car\_date(DateTime date)  Input: datr  Output: int | * Đếm tổng lượng xe gửi theo ngày =date | TicketDal.cs |

*Bảng 2.4.8. Đặc tả các phương thức trong lớp WorkDal*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *1* | load\_work()  Output: List  List<work> | - Lấy danh các thông tin cong việc từ cơ sở dữ liệu. | WorkDal.cs |
| *2* | insert\_work(WorkPublic workPublic)  Input: Work  Output: bool | - Thêm thông tin công việc vào cơ sở dữ liệu. | WorkDal.cs |
| *3* | update\_work(WorkPublic workPublic)  Input: Work  Output: bool | - Cập nhật thông tin công việc “work” vào cơ sở dữ liệu | WorkDal.cs |
| *4* | delete\_work(WorkPublic workPublic)  Input: Work  Output: bool | - Xóa thông tin công việc trong cơ sở dữ liệu. | WorkDal.cs |
| *5* | max\_id()  Input: none  Output: int | - Lấy ra ID lớn nhất từ cơ sở dữ liệu | WorkDal.cs |

**PHẦN 3. KẾT LUẬN  
1. KẾT LUẬN**

Về cơ bản nhóm tự nhận xét phần mềm của nhóm đã giải quyết được hầu như

các yêu cầu đã mà đồ án đã đề ra và còn thêm được các tính năng thêm hữu ích cho chương trình.

**Ưu điểm:**

● Giao diện gọn gàng, trực quan dễ tiếp cận, dễ làm quen.  
● Dung lượng khá nhẹ.  
● Chương trình tốn rất ít tài nguyên của hệ thống.  
● Chương trình chạy ổn định, được kiểm thử nhiều lần và cho ra kết quả chính xác, không bị crash, debug, except hay lag trong quá trình thực thi đảm bảo đúng yêu cầu người dùng

**Nhược điểm:**

●Thuật toán chưa tối ưu khi xử lí nhiều luồng thông tin dữ liệu lớn (Đôi lúc xử lí còn chậm).

● Chưa thể tìm laị dữ liệu khi chẳng may bị mất.

● Phân quyền người đăng nhập vẫn còn hạn chế.

**2. HƯỚNG PHÁT TRIỂN**

-Nâng cấp giao diện.

-Viết thêm tính năng tìm kiếm tài khoản khi bị mất mật khẩu

-Thêm tính năng đăng kí tài khoản.

-Public cơ sở dữ liệu trên mạng internet để có thể truy cập từ nhiều máy khác

nhau.

-Thêm tính năng xuất thông tin thành file excel và docx.

# **3. TÀI LIỆU KHAM KHẢO**

[1] Nguyễn Thành Sơn, *Hệ quản trị cơ sở dữ liệu*, Đại học sư phạm kỹ thuật thành phố Hồ Chí Minh.  
[2] Huỳnh Xuân Phụng, *Lập trình trên Windown*, Đại học sư phạm kỹ thuật thành phố Hồ Chí Minh, https://youtu.be/1AjwxaBapEA, [10/11/2020]

[3] Đồ án báo cáo mẫu, *BaoCao\_DoAn\_DBMS\_iLAB,* Đại học sư phạm kỹ thuật thành phố Hồ Chí Minh.

[4] Hoàng Văn Dũng, *Lập trình trên Windown*, Đại học sư phạm kỹ thuật thành phố Hồ Chí Minh.